

Perceptual Quality of Point Clouds with application to Compression

Présentée le 16 avril 2021

Faculté des sciences et techniques de l'ingénieur
Groupe Ebrahimi
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Evangelos ALEXIOU

Acceptée sur proposition du jury

Prof. P. Frossard, président du jury
Prof. T. Ebrahimi, directeur de thèse
Prof. F. M. B. Pereira, rapporteur
Prof. G. Lavoué, rapporteur
Prof. J.-Ph. Thiran, rapporteur

Make the best use of what is in your power,
and take the rest as it occurs.
— Epictetus

Acknowledgements

This experience wouldn't have been possible without the opportunity that was given to me by Prof. Touradj Ebrahimi to join his lab. Not only his supervision and support, but also his guidance and advices have been invaluable. The endless opportunities that were generously provided to me, triggered fascinating research, fruitful collaborations, personal relationships, great trips and participation in exhibitions that I couldn't possibly imagine, equipping me with knowledge and memories that will always be part of me.

I would also like to thank every member of my committee jury, namely, Prof. Fernando Pereira, Prof. Guillaume Lavoué, Prof. Jean-Philippe Thiran and Prof. Pascal Frossard, for taking the time to review this dissertation, expressing insightful aspects and constructive comments.

Moreover, I would like to express my appreciation to all the members of the JPEG community for the knowledge that I have gained through the interactions that were held as a result of my involvement in related activities. I have learnt a lot on how to advance, structure and communicate my research, as well as how to act as a collaborator.

There are critical moments in life that only when seen backwards, their importance is realized. My acquaintance with Prof. Alexandros Eleftheriadis was one such moment. For two years he had been providing me with knowledge, confidence, and vision, before opening my horizons for the journey in Lausanne. I want to express my deep gratitude and appreciation.

My friends in my home-town, Kwstas and Thanasis, as well as the friends that I made as early as my Bachelor years, Thanasis, Giannis, Panos, Michalis, Kwstas, among others, are the ones to thank for their constant companion, great laughs, and long-time experiences that formed us in these years.

I continue with new friends, and as such I consider all the MMSPG members I had the privilege to collaborate and share my daily life. From Martin that guided me in the early times to the "secrets" of the corridor, to the short-term but always fun memories with Lin and Ashu, the great times with excursions in French and Greek places with Anne-Flore, the drama and joy of life with Pinar, the entire experience that was shared with Eugene and Irene up until the end, it was just great! The MMSPG trips to Turin, Erfurt and Crans Montana, the intense moments in Montreux, and the beautiful Christmas dinners, will always be memorable. Many acknowledgements to Christine, for successfully arranging such events, dealing with any possible administration details (that I very much dislike) and for giving invaluable advices for survival in Switzerland, Lausanne, EPFL, etc. I would also like to thank David, for being the person who took me by hand and assisted in many ways in my first period in Lausanne. Nice moments were also shared with the newest members of the lab, namely, Saleh, Michela and

Acknowledgements

Davi. I am sure they will keep rocking the corridor, as their ancestors!

I would like to also to take a moment to thank people that, despite their short passage, were always considered as members of the lab, which was proven by our out-of-campus interactions, such as Ashu, Pablo and Sherry. Moreover, I would like to thank the students that collaborated with me. With each and everyone's question I was learning how to become a better mentor, and I am happy that for some of them, their experience was sufficient to decide their career path! Special thanks go to Joachim, Patrick, Delphine, Peisen, Xinyu, Amine, Pol, Manon, Peng Yi, Nanyang, Chun-Hung, Kuan, and Luis.

I am considering myself lucky for the wonderful people that I met and had the pleasure to share a kitchen, a bridge with the Alps view, funny moments, interesting discussions, delighting barbecues, and subjective evaluation experiences. Hermina, Roberto, Mireille, Guillermo, Clément, Beril, Helena, Vlad, Kwstas, Andreas, Nikos, the Turkish gang, and all the people in the corridor and around it, thank you for making our experience so much more colorful. A big thanks to Eugene for being a constant partner and for the great experiences we shared! I would also like to thank the Greeks that I met in Lausanne, for the nice times and for connecting me with my homeland. Marios, George and Lefteris, thank you guys! A special thanks goes to the other Greeks, the ones that I had the fortune to continue the journey with in Lausanne. Kwstas, Tolis and Meli, it has been a great pleasure sharing dinners, beers, barbecues, flats, quarantine-times, and lots of fun throughout all these years! You have been always available and supportive in every circumstance that was faced this entire period, for which I want to express my deepest appreciation. Cheers!

I have been shaped by the unconditional love of my family and their ethics in life. For that, there are no words to express the level of my gratitude to my father Giorgos, my mother Eucharistia, my brothers Giannis and Alexandros. They have always been providing the most comforting shelter to me, constantly proving that distance is just a number.

Concluding, I want to express my greatest acknowledgement to Irene, for her support to me. Starting from finding my first flat in Lausanne, up to the revision of this manuscript, Irene has always been contributing, assisting and making my life better. A big thanks for all the moments you made funnier, happier, easier, more positive, more constructive, is just too little!

Lausanne, January 28, 2021

Evangelos Alexiou

Abstract

Modern information technologies and human-centric communication systems employ advanced content representations for richer portrayals of the real world. The newly adopted imaging modalities offer additional information cues and permit the depiction of realistic sceneries, enabling immersive experiences and promoting the engagement of the user with the content. In this context, point clouds have emerged as an attractive option to represent immersive media. This type of visual data has seen a revived interest in the recent years, following the release of low-cost depth sensors and the wide integration of modern graphics processing units in mobile phones and personal computers. Point clouds can be naturally employed in extended reality applications that involve 6 degree-of-freedom interactions, allowing adjustments of the 3-D visual information in a per-point basis. At the same time, complexity reductions are promoted when compared to mesh modelling counterpart, due to the absence of connectivity information and the elimination of corresponding constraints from acquisition to rendering.

Yet, the vast amount of information that is required for faithful content representations implies the necessity for efficient data structures and compression algorithms. In particular, new coding schemes must be designed in order to reduce the amount of data and by extension the costs in processing, storage, and transmission of point clouds, while lossy compression solutions should restrain degradations for more appealing results. Furthermore, adequate subjective quality evaluation methodologies tailored to the nature of this 3D imaging modality are essential in order to obtain ground-truth data, and to better understand the impact of compression and rendering artifacts in visual quality. The development of high-performing objective metrics is also fundamental to accurately predict the perceptual quality of degraded models.

In this thesis, we address the aforementioned challenges by proposing new subjective quality assessment methodologies that better simulate realistic use-cases of 3D model consumption. We examine several aspects related to point cloud visualization and display means, by exploring different rendering approaches and by introducing experimental set-ups that offer different degrees of interactivity to the end-user. The behavior of human observers in 6 degrees-of-freedom virtual reality scenes is analysed, and visual attention maps are constructed using head and gaze trajectories recorded from eye-tracking experiments. Moreover, navigation data obtained from interactive subjective evaluations in desktop arrangements are exploited to improve image-based quality metrics, whose performance is examined in predicting visual impairments on point cloud contents. In the same line of research, we

Abstract

design novel point-based quality predictors for point cloud topology and texture degradations, and we rigorously analyse their performance using several subjectively annotated data sets. Furthermore, adopting well-established subjective evaluation methodologies, state-of-the-art compression solutions are benchmarked and best practices for rate allocation between geometry and texture encoding are derived. Lastly, a learning-based point cloud compression solution for encoding of both geometric and color information is proposed, and the impact of a series of parameters is examined on the obtained performance to pave the path for future efforts on the field.

Keywords: Point cloud, perceptual quality, subjective evaluation, objective metrics, compression, visual attention, rendering, augmented reality, virtual reality, deep-neural networks.

Sommario

Le moderne tecnologie informatiche e i sistemi di comunicazione centrati sugli utenti utilizzano modalità di rappresentazione dei contenuti avanzate per ottenere ritratti più ricchi del mondo reale. Queste nuove modalità visive offrono informazioni aggiuntive e permettono di mostrare scenari realistici, creando esperienze immersive e promuovendo l'interazione dell'utente col contenuto. In questo contesto, emergono i point cloud come una valida opzione per rappresentare contenuti multimediali immersivi. Questo tipo di dato visuale ha ottenuto un rinnovato interesse in anni recenti, in seguito alla distribuzione di sensori di profondità a basso costo e la diffusa integrazione di unità di elaborazione grafica moderne in personal computer e cellulari. I point cloud possono essere utilizzati naturalmente in applicazioni di extended reality che coinvolgono interazioni a 6 gradi di libertà, permettendo di cambiare l'aspetto del modello 3D visualizzato punto per punto. Al tempo stesso, si osservano riduzioni nella complessità rispetto alla controparte dei modelli mesh, dati dall'omissione di informazioni sulla connettività e l'eliminazione dei corrispettivi limiti, dall'acquisizione al rendering.

Tuttavia, la larga quantità di informazione che è richiesta per una rappresentazione fedele dei contenuti indica la necessità di avere strutture dati e algoritmi di compressione efficienti. Nello specifico, nuovi schemi di compressione devono essere progettati per ridurre la quantità di dati e per estensione i costi di elaborazione, salvataggio e trasmissione di point cloud, mentre soluzioni di compressione lossy devono limitare degradazioni visive per ottenere risultati più piacevoli. Inoltre, metodologie di valutazione soggettiva adeguate e pensate su misura per la natura di questa modalità di imaging 3D sono essenziali per ottenere informazioni ground-truth e per meglio capire l'impatto degli algoritmi di compressione e di artefatti nel rendering sulla qualità visiva. Lo sviluppo di metriche oggettive di qualità che siano altamente performanti è anch'esso fondamentale per predire accuratamente il livello di degradazione dei modelli.

In questa tesi, affrontiamo le suddette sfide proponendo nuove metodologie di valutazione soggettive che simulano più fedelmente usi realistici di consumo di modelli 3D. Esaminiamo diversi aspetti legati alla visualizzazione e al display di point cloud, esplorando diversi meccanismi di rendering e introducendo set-up sperimentali che offrono diversi gradi di interattività per l'utente. Il comportamento di osservatori umani in realtà virtuale con 6 gradi di libertà è analizzato, e mappe di attenzione visiva sono create usando traiettorie oculari e craniali registrate in esperimenti di eye-tracking. Inoltre, dati di navigazione ottenuti in esperimenti soggettivi interattivi in assetti desktop sono sfruttati per migliorare metriche oggettive di qua-

lità basate su immagini, la cui performance nel predire distorsioni visive su contenuti point cloud è esaminata. Nella stessa linea di ricerca, progettiamo nuovi predittori punto-punto di qualità per la valutazione oggettiva di qualità, per distorsioni sulla topologia e sulla texture dei point cloud, e analizziamo rigorosamente la loro performance usando diversi dataset con annotazioni soggettive. Inoltre, con l'adozione di metodologie di valutazione soggettiva e metriche oggettive collaudate, proponiamo un benchmark dello stato dell'arte in soluzioni di compressione, e consigli sull'allocazione di rate per la codifica di dati di geometria e texture sono ideati. Infine, si propone una soluzione di compressione learning-based per gestire la codifica di attributi geometrici e di colore di contenuti point cloud, e l'impatto di una serie di parametri sulla performance è esaminato, per aprire la strada a futuri lavori nel campo.

Parole chiave: Point cloud, qualità percepita, valutazione soggettiva, metriche oggettive, compressione, attenzione visiva, rendering, realtà aumentata, realtà virtuale, reti neurali.

Contents

Acknowledgements	i
Abstract	iii
Sommario	v
Table of Contents	vii
List of Figures	xiii
List of Tables	xxiii
1 Introduction	1
1.1 Contributions	3
1.1.1 Measuring perceptual quality	3
1.1.2 Modelling perceptual quality	4
1.1.3 Towards efficient compression	5
2 Related work	7
2.1 Acquisition	7
2.1.1 Passive techniques	7
2.1.2 Active techniques	8
2.1.3 Discussion	9
2.2 Compression	10
2.2.1 Model-based encoding	10
2.2.2 Projection-based encoding	12
2.2.3 Deep learning-based encoding	13
2.2.4 Discussion	14
2.3 Rendering	15
2.3.1 Point-based rendering	15
2.3.2 Mesh-based rendering	17
2.3.3 Discussion	18
2.4 Quality assessment	19
2.4.1 Subjective quality assessment	20
2.4.2 Objective quality metrics	25
	vii

2.4.3	Discussion	31
I	Measuring perceptual quality	33
3	Quality evaluation of point clouds geometry	35
3.1	Data set preparation	36
3.1.1	Content selection	36
3.1.2	Content preparation	36
3.1.3	Degradation types	37
3.2	Evaluation methodologies	40
3.2.1	Data set	40
3.2.2	Methodology	40
3.2.3	Results	43
3.3	Display devices	47
3.3.1	Data set	48
3.3.2	Methodology	48
3.3.3	Results	51
3.4	Rendering schemes	55
3.4.1	Data set	56
3.4.2	Methodology	56
3.4.3	Results	59
3.5	Conclusions	64
4	Quality evaluation of colored point clouds	65
4.1	Point rendering primitives	67
4.1.1	Data set	68
4.1.2	Methodology	69
4.1.3	Results	71
4.2	Point-based rendering schemes	75
4.2.1	Data set	76
4.2.2	Methodology	80
4.2.3	Results	83
4.3	Conclusions	90
5	Exploring immersive technologies	93
5.1	Subjective quality evaluation in virtual reality	94
5.1.1	Data set	95
5.1.2	Methodology	96
5.1.3	Results	99
5.2	Visual attention in virtual reality	104
5.2.1	Data set	107
5.2.2	Methodology	107

5.2.3	Results	115
5.3	Towards visual attention in virtual museums	119
5.3.1	Scene design	120
5.3.2	Supplementary tools	121
5.4	Conclusions	124
II	Modelling perceptual quality	127
6	Point-based objective quality metrics	129
6.1	Point cloud angular similarity	130
6.1.1	Definition	131
6.1.2	Validation methodology	134
6.1.3	Results	138
6.1.4	Discussion	149
6.2	Point cloud structural similarity	152
6.2.1	Definition	153
6.2.2	Validation methodology	157
6.2.3	Results	158
6.2.4	Discussion	171
6.3	Conclusions	173
7	Image-based objective quality metrics	175
7.1	Exploiting model views	176
7.1.1	Model views generation	177
7.1.2	Validation methodology	178
7.1.3	Results	181
7.2	Exploiting user views	186
7.2.1	User interactivity	187
7.2.2	Validation methodology	188
7.2.3	Results	189
7.3	Conclusions	194
8	Benchmarking of objective quality metrics	197
8.1	Validation methodology	197
8.1.1	Data set	197
8.1.2	Computation of quality metrics	198
8.1.3	Benchmarking of quality metrics	199
8.2	Results	200
8.3	Conclusions	209

III	Towards efficient compression	211
9	Benchmarking of MPEG codecs	213
9.1	Data set preparation	215
9.1.1	Content selection	215
9.1.2	Content preparation	215
9.1.3	Degradation types	216
9.2	MPEG Common Test Conditions	219
9.2.1	Data set	219
9.2.2	Methodology	219
9.2.3	Results	224
9.3	Rate allocation for geometry encoding	229
9.3.1	Data set	230
9.3.2	Methodology	231
9.3.3	Results	232
9.4	Rate allocation for geometry and color encoding	235
9.4.1	Data set	235
9.4.2	Methodology	236
9.4.3	Results	237
9.5	Conclusions	239
10	Learning-based encoding	241
10.1	Network architecture	242
10.1.1	Input	242
10.1.2	Auto-encoder	243
10.1.3	Output	245
10.2	Experimental setup	246
10.2.1	Data set	246
10.2.2	Evaluation methodology	247
10.2.3	Network configurations	248
10.3	Experimental results	248
10.3.1	Geometry against color impairments using the unified network	249
10.3.2	Unified network against separately trained networks	253
10.3.3	Benchmarking of unified network	254
10.4	Meta-analysis	257
10.4.1	Selection of training data for geometry compression	257
10.4.2	Resolution of testing data	258
10.4.3	Color space	260
10.4.4	Loss function	260
10.5	Conclusions	261
11	Conclusions	263

Annexes	271
A Statistical analysis tools	273
A.1 Processing of subjective scores	273
A.1.1 Category rating	274
A.1.2 Pair comparison	275
A.2 Comparison of subjective scores from different experiments	276
A.2.1 Data mapping	276
A.2.2 Statistical evaluation metrics	278
A.2.3 Estimation errors	279
A.2.4 Classification errors	280
A.2.5 Standard deviation of Opinion Score	281
A.2.6 Inferential statistical methods	281
A.3 Comparison of objective against subjective scores	283
A.3.1 Data mapping	283
A.3.2 Performance indexes	284
A.4 Comparison of rate-distortion curves	284
B Point cloud data structures	285
B.1 Octree structure	285
B.2 Voxel grids	286
B.2.1 Implementations	288
C Accuracy of normal estimation algorithms	291
C.1 Data set	292
C.2 Computation of normal vectors	292
C.2.1 Ground-truth normal vectors	292
C.2.2 Estimated normal vectors	293
C.3 Performance evaluation	293
C.4 Results	294
D Renderers	297
D.1 Voxel-based renderer	298
D.2 Splat-based renderer in VTK	299
D.3 Splat-based renderer in JS	300
D.4 Splat-based renderer in Unity: PointXR toolbox	302
E Open access material	305
Bibliography	309
Curriculum Vitae	329

List of Figures

3.1	Reference point cloud contents.	37
3.2	Illustrative examples of the visual artifacts that are introduced by the selected types of degradation.	38
3.3	testbed	41
3.4	Participant inspecting the point clouds under assessment in the desktop set-up.	42
3.5	Subjective scores against degradation levels using the DSIS test method. . . .	44
3.6	Subjective scores against degradation levels using the ACR test method. . . .	44
3.7	Comparison of subjective scores obtained from both test methods (DSIS is set as the ground truth).	46
3.8	Comparison of subjective scores obtained from both test methods (ACR is set as the ground truth).	47
3.9	Rendering application screen shot showing the reference <i>bunny</i> on the left and its impaired version with Gaussian noise of $\sigma = 0.008$ on the right.	49
3.10	Participant inspecting the point clouds under assessment in the HMD AR set-up.	50
3.11	Subjective scores against degradation levels in the HMD AR experiment. . . .	52
3.12	Comparison of subjective scores obtained under both display devices (Desktop scores are set as the ground truth).	53
3.13	Comparison of subjective scores obtained under both display devices (HMD AR scores are set as the ground truth).	54
3.14	Reference point cloud contents after conversion to meshes.	58
3.15	Reconstructed <i>bunny</i> under all degradation levels from Octree-pruning. . . .	58
3.16	Subjective scores against degradation levels per laboratory.	60
3.17	Comparison of subjective scores obtained from different laboratories (Bold text represents the ground truth).	61
3.18	Comparison of subjective scores obtained under both rendering schemes (Bold text represents the ground truth).	63
4.1	Point cloud content using different rendering methods.	66
4.2	Original point cloud contents.	68
4.3	Pair comparison of point rendering primitives across all contents.	71
4.4	Pair comparison of point rendering primitives per content.	72
4.5	Normalized quality scores from subjective preferences for the adopted point rendering primitives.	73

List of Figures

4.6	Content representations using disks, cubes and spheres as point rendering primitives from left to right.	74
4.7	Subjective scores after aggregating preferences over point clouds of the same sparsity level in the first row, and the same content type in the second row. . .	75
4.8	Operational logic of the rendering schemes under evaluation.	76
4.9	Pre-visualization processing workflow.	77
4.10	Reference test contents.	78
4.11	Splat-based subjective evaluation testbed.	81
4.12	Voxel-based subjective evaluation testbed.	82
4.13	Subjective scores against degradation levels using both rendering solutions. .	84
4.14	Screen-shots of the frontal view of <i>amphoriskos</i> , encoded at the lowest color quality and every geometric degradation level. Top row: models displayed in the splat-based renderer. Bottom row: models displayed in the voxel-based renderer.	85
4.15	Screen-shots of the frontal view of <i>longdress</i> , encoded at the highest color quality and every geometric degradation level. Top row: models displayed in the splat-based renderer. Bottom row: models displayed in the voxel-based renderer.	86
4.16	Subjective scores against total bit-rates using both rendering solutions.	87
4.17	Significance difference matrices at a 5% level per experiment, indicating the preference of subjects for a particular degradation against all others. Note that 0 and 6 denote the minimum and maximum numbers of preference, respectively, given a total of 6 contents.	88
4.18	Comparison of subjective scores obtained under both rendering solutions (Bold text represents the ground truth).	89
5.1	Frontal view of the models.	95
5.2	Virtual environment.	98
5.3	Experimental set-up.	99
5.4	Subjective scores against color bit-rates from both codecs, using both test methods.	100
5.5	Comparison of subjective scores between RAHT and Lifting (ground truth) color codec.	101
5.6	Comparison of subjective scores obtained under both DSIS variants (Bold text represents the ground truth).	102
5.7	Average time of interaction.	103
5.8	Interactivity patterns per evaluation protocol.	104
5.9	Selected contents for the experiment.	106
5.10	Apparatus of the experiment.	108
5.11	Schematic diagram with the hardware and software modules together with their inter-dependencies.	109
5.12	Eye camera window from Pupil Capture software.	109

5.13	Virtual reality scene. The environment and the illumination are not distracting, while the shadows underneath the models enhance the sense of realism. . . .	110
5.14	Gaze points calibration and evaluation of measurements.	111
5.15	Estimation of angular error for gaze points. On the <i>left</i> figure, we present the gaze point g and its four adjacent markers m_i , $i = \{1, 2, 3, 4\}$. We assume that the corresponding angular errors are <i>valid</i> . In the <i>middle</i> figure, the two candidate surrounding triangles are depicted, indicated as T_1 and T_2 . Given that g is closer to the vertices of T_1 , the barycentric coordinates of g in T_1 are computed and, then, interpolated in order to estimate the angular error of the gaze point g	113
5.16	Visual interpretations for key-components employed in the proposed heuristic methodology to identify fixations.	114
5.17	Histograms indicating the probability of gaze and fixation positions as a function of the viewing angle with respect to the head direction.	116
5.18	Importance weights from fixation density maps using gaze information.	117
5.19	Importance weights from using head-trajectories.	118
5.20	Time of inspection across models and users.	118
5.21	Overview of the virtual museum.	119
5.22	Exemplary views of the virtual environment from the user's perspective.	120
5.23	Demonstration of user interactions that is played-back using the recorded data. The head position is indicated by the white avatar and the gaze direction by the purple line.	121
5.24	Menu for configuration of markers.	122
5.25	Built-in markers integrated in the module.	122
6.1	Point cloud angular similarity metric.	132
6.2	G-PCD: Subjective against objective scores from the best-performing proposed (left) and anchor (right) quality metrics under Gaussian noise, using the ACR test method.	138
6.3	G-PCD: Subjective against objective scores from the best-performing proposed (left) and anchor (right) quality metrics under Octree-pruning, using the ACR test method.	139
6.4	J-PCED2: Performance indexes PLCC and SROCC of plane-to-plane metric, per normal estimation algorithm and configuration.	140
6.5	J-PCED2: Subjective against objective scores from plane-to-plane metric, per normal estimation algorithm and configuration.	141
6.6	J-PCED2: Subjective against objective scores from the best-performing configuration (i.e., plane fitting with $k = 256$) of the proposed (left) and anchor (right) quality metrics. The right plot is zoomed-in to provide a more informative view, capturing the majority of stimuli.	142
6.7	M-PCCD: Performance indexes PLCC and SROCC of plane-to-plane metric, per normal estimation algorithm and configuration.	143

List of Figures

6.8	M-PCCD: Subjective against objective scores from plane-to-plane metric, per normal estimation algorithm and configuration.	144
6.9	M-PCCD: Subjective against objective scores from the best-performing configuration (i.e., quadric fitting with $R = 30$) of the proposed (left) and anchor (right) quality metrics. The right plot is zoomed-in to provide a more informative view, capturing the majority of stimuli.	145
6.10	IRPC: Performance indexes PLCC and SROCC of plane-to-plane metric, per normal estimation algorithm and configuration.	146
6.11	IRPC: Subjective against objective scores from plane-to-plane metric, per normal estimation algorithm and configuration.	147
6.12	IRPC: Subjective against objective scores from the best-performing configuration (i.e., linear fitting with $R = 10/40$ for 10-bit and 12-bit contents, respectively) of the proposed (left) and anchor (right) quality metrics.	148
6.13	Visualization of the reference <i>longdress</i> with point shading. The normal vectors are estimated with plane fitting and $k = 8, 32, 128$ and 512 from left to right. . .	150
6.14	Illustration of normal surface approximations. The geometry of the reference model <i>longdress</i> , a version after encoding with Octree at R02, and another version after encoding using TriSoup at R01 following the MPEG Common Test Conditions are displayed, from left to right. The normal vectors are estimated using plane fitting with range search and radius 10. The obtained plane-to-plane scores are 0.883 and 0.796, for the second and third stimulus, respectively.	151
6.15	Block diagram of the feature extraction steps. \dot{P} indicates an input point cloud, and P the point cloud at the output of the <i>voxelization</i> step. The features map $F^{d,k,t,e}(P)$ consists of structural features extracted from every point p that belongs to P , for a selected voxel bit-depth d , neighborhood size k , attribute t and dispersion estimator e	153
6.16	Illustrative example of point association. The model A is set as the reference and the model B as under evaluation. The point a belongs to A and denotes the nearest neighbor of point b that belongs to B . The local neighborhoods $N(a)$ and $N(b)$ are defined around the former and the latter, respectively, in order to compute corresponding features.	155
6.17	J-PCED2: Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the neighborhood size is 6, 12, 24 and 48, from left to right.	159
6.18	J-PCED2: Curvature-based features. Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the corner left bar corresponds to no voxelization, whereas the rest of the bars correspond to voxel bit-depths equal to 9, 8, 7 and 6, from left to right.	160

6.19	J-PCED2: Subjective against objective scores from the best-performing configuration (i.e., curvature-based features, voxel depth of 9 bits, dispersion estimator σ^2 , neighborhood size of 6) of the proposed (left) and anchor (right) quality metrics. The right plot is zoomed-in to provide a more informative view, capturing the majority of stimuli.	161
6.20	M-PCCD: Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the neighborhood size is 6, 12, 24 and 48, from left to right.	162
6.21	M-PCCD: Luminance-based features. Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the corner left bar corresponds to no voxelization, whereas the rest of the bars correspond to voxel bit-depths equal to 9, 8, 7 and 6, from left to right.	163
6.22	M-PCCD: Subjective against objective scores from the best-performing configuration (i.e., luminance-based features, voxel depth of 9 bits, dispersion estimator σ^2 , neighborhood size of 12) of the proposed (left) and anchor (right) quality metrics. The right plot is zoomed-in to provide a more informative view, capturing the majority of stimuli.	164
6.23	IRPC <i>rpoint</i> : Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the neighborhood size is 6, 12, 24 and 48, from left to right.	165
6.24	IRPC <i>rpoint</i> : Normal-based features. Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the corner left bar corresponds to no voxelization, whereas the rest of the bars correspond to voxel bit-depths equal to 10, 9, 8 and 7, from left to right.	166
6.25	IRPC <i>rpoint</i> : Subjective against objective scores from the best-performing configuration (i.e., normal-based features, voxel depth of 9 bits, dispersion estimator QCD, neighborhood size of 48) of the proposed (left) and anchor (right) quality metrics.	167
6.26	IRPC <i>rcolor</i> : Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the neighborhood size is 6, 12, 24 and 48, from left to right.	168
6.27	IRPC <i>rcolor</i> : Luminance-based features. Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the corner left bar corresponds to no voxelization, whereas the rest of the bars correspond to voxel bit-depths equal to 10, 9, 8 and 7, from left to right.	169
6.28	IRPC <i>rcolor</i> : Subjective against objective scores from the best-performing configuration (i.e., luminance-based features, voxel depth of 8 bits, dispersion estimator <i>mAD</i> , neighborhood size of 24) of the proposed (left) and anchor (right) quality metrics.	170

List of Figures

6.29	Illustration of a point structural similarity map (black indicates similarity of 1). In the first row, the luminance component of the reference model <i>longdress</i> , and two versions after encoding with V-PCC at R03 and R01 following the MPEG Common Test Conditions are displayed, from left to right. In the bottom row, the corresponding structural similarity scores using luminance-based features are provided. The obtained PointSSIM scores are 0.519 and 0.359, for the second and the third stimulus, respectively.	172
7.1	Camera layouts to capture views of the models.	177
7.2	Model views captured from a camera layout with $K = 6$	178
7.3	Initial frontal view of reference model <i>loot</i> as displayed to participants during subjective evaluations in corresponding testbeds.	179
7.4	Subjective against objective scores from the best-performing image-based quality metrics per type of content, under voxel-based rendering.	181
7.5	Subjective against objective scores from the best-performing image-based quality metrics per type of content, under splat-based rendering.	182
7.6	Model view of <i>longdress</i> as consumed by subjects (left) and after removing the background information (right).	183
7.7	Subjective against objective scores from the best-performing image-based quality metrics per type of content, computed after background removal, under splat-based rendering.	184
7.8	Subjective against objective scores from MS-SSIM per type of content, computed on $K = 1$ model view after background removal, under splat-based rendering.	186
7.9	Subjective against objective scores from the best-performing image-based quality metrics per type of content, computed on the navigation tracks of the users.	190
7.10	MS-SSIM scores from the navigation tracks of every user inspecting a stimulus. In the first and second row, encoded versions of the model <i>amphoriskos</i> and <i>longdress</i> are depicted, respectively. Corresponding frontal views of the stimuli under consideration are provided in Figures 4.14 and 4.15. Remark that the evaluation of a given stimulus starts with the same frontal view for all the users; thus, the corresponding MS-SSIM value will be present for all subjects. Also note that at low quality levels, users are interacting less.	191
7.11	Dot markers on the view sphere correspond to camera positions for a 2-level subdivision of an icosahedron ($K = 162$). The color code represents the ranking of weights, ranging from dark blue (minimum) to dark red (maximum).	193
7.12	Views of model <i>biplane</i> on top and <i>loot</i> on the bottom, with corresponding importance weights.	194
8.1	Subjective against objective scores for a selection of metrics, considering the entire data set. A zoomed view is provided for PCQM, MMD, PCM_RR and point-to-plane with MSE.	202

8.1	Subjective against objective scores for a selection of metrics, considering the entire data set.	203
8.2	Subjective against objective scores for a selection of metrics, considering stimuli clustered per codec.	205
8.3	Subjective against objective scores for a selection of metrics, considering stimuli clustered per type of content.	207
9.1	Reference point cloud models. The set of objects is presented in the first row, whilst the set of human figures is illustrated in the second row.	214
9.2	V-PCC compression process. The original point cloud is initially decomposed into geometry video, texture video and metadata. Both video contents are smoothed by <i>Padding</i> to allow for the best HEVC (Bross et al., 2012) performance. The compressed bit-streams (metadata, geometry video and texture video) are packed into a single bit-stream: the compressed point cloud.	217
9.3	Overview of G-PCC geometry encoder. After voxelization, the geometry is encoded either by Octree or by TriSoup modules, which depends on Octree.	217
9.4	Overview of G-PCC color attribute encoder. In the scope of this work, either RAHT or Lifting are used to encode contents under test.	218
9.5	Illustration of artifacts occurred after encoding the content <i>amphoriskos</i> with the codecs under evaluation. To obtain comparable visual quality, different degradation levels are selected for V-PCC and G-PCC variants.	220
9.6	Illustration of the evaluation platform. Both reference and distorted models are presented side-by-side while being clearly remarked. Users' judgements can be submitted through the rating panel. The green bar at the bottom indicates the progress in the current batch.	222
9.7	Comparison of subjective scores obtained from the participated laboratories (Bold text represents the ground truth).	223
9.8	MOS vs SOS fitting for scores obtained in EPFL and UNB, with relative SOS coefficient α . The shaded plot indicates the 95% confidence bounds for both fittings.	224
9.9	Subjective scores against bit-rates from the degradation levels defined for every codec, grouped per content. Curves for point clouds that represent inanimate objects are illustrated.	225
9.10	Subjective scores against bit-rates from the degradation levels defined for every codec, grouped per content. Curves for point clouds that represent human figures are illustrated.	226
9.11	<i>Soldier</i> encoded with V-PCC. Although the R4 degraded version is blurrier with respect to R5, missing points in the latter model were rated as more annoying. (examples are highlighted in the figures).	227
9.12	<i>Biplane</i> encoded with V-PCC. The color smoothing resulting from the low-pass filtering in texture leads to less annoying artifacts for R2 with respect to R3.	227

List of Figures

9.13	Preference and tie probabilities for each pair of configurations under test in experiment 2, for the high bit-rate case. The color blue (yellow) of the bar indicates the probability of the configuration on the left (right) side being preferred over the one on the right (left) side. The orange bar indicates the tie probability.	232
9.14	Preference and tie probabilities for each pair of configurations under test in experiment 2, for the low bit-rate case. The color blue (yellow) of the bar indicates the probability of the configuration on the left (right) side being preferred over the one on the right (left) side. The orange bar indicates the tie probability.	233
9.15	Normalized MOS and relative CIs obtained from the winning frequencies gathered in experiment 2, for each configuration, averaged across the contents, separately for high and low bit-rates.	234
9.16	Preference and tie probabilities for each pair of configurations under test in experiment 3, for the high bit-rate case. The color blue (yellow) of the bar indicates the probability of the configuration on the left (right) side being preferred over the one on the right (left) side. The orange bar indicates the tie probability.	237
9.17	Preference and tie probabilities for each pair of configurations under test in experiment 3, for the low bit-rate case. The color blue (yellow) of the bar indicates the probability of the configuration on the left (right) side being preferred over the one on the right (left) side. The orange bar indicates the tie probability.	238
9.18	Normalized MOS and relative CIs obtained from the winning frequencies gathered in experiment 3, for each configuration, averaged across the contents, separately for high and low bit-rates.	239
10.1	Auto-encoding architecture.	243
10.2	Sample models used for training.	246
10.3	Models used for testing.	246
10.4	Rate-distortion performance of the unified network architecture, according to geometry metric D2-PSNR, with different λ allocations to geometry and color ($\lambda_g : \lambda_c$). Solid black represents pure geometry compression ($\lambda_c = 0$), solid red represents 1:1 allocation. Dashed lines represent allocations for which $\lambda_g > \lambda_c$, whereas for dotted lines, $\lambda_g < \lambda_c$	249
10.5	Rate-distortion performance of the unified network architecture, according to geometry metric YUV-PSNR, with different λ allocations to geometry and color ($\lambda_g : \lambda_c$). Solid black represents pure color compression ($\lambda_g = 0$), solid red represents 1:1 allocation. Dashed lines represent allocations for which $\lambda_g > \lambda_c$, whereas for dotted lines, $\lambda_g < \lambda_c$	250

10.6	Bjontegaard dB gains for each allocation $\lambda_g : \lambda_c$ with respect to allocation 1:1, for color metric YUV-PSNR (blue) and geometry metric D2-PSNR (red). Dashed lines represent dB gains when using pure color compression (blue) or pure geometry compression (red), with respect to 1:1 baseline.	251
10.7	Visual comparison for <i>longdress</i> , for different distortion allocation ratios. . . .	252
10.8	Visual comparison for <i>guanyin</i> , for different distortion allocation ratios. . . .	252
10.9	Rate-distortion performance of the unified model and the separately trained networks, according to geometry metric D2-PSNR.	254
10.10	Rate-distortion performance of the unified model and the separately trained networks, according to color metric YUV-PSNR.	254
10.11	Rate-distortion performance of the of the unified model, trained with block resolution of 32 and 64, against the MPEG anchor, according to geometry metric D2-PSNR.	255
10.12	Rate-distortion performance of the unified model, trained with block resolution of 32 and 64, against the MPEG anchor, according to color metric YUV-PSNR.	255
10.13	Visual comparison for <i>longdress</i> , compressed using the proposed network and the MPEG anchor.	256
10.14	Visual comparison for <i>guanyin</i> , compressed using the proposed network and the MPEG anchor.	256
10.15	Rate-distortion performance of the geometry-only network, using different data sets and training data resolutions, according to geometry metric D2-PSNR.	258
10.16	Rate-distortion performance of the geometry-only network, for different testing grid resolutions, according to geometry metric D2-PSNR. First row represents results obtained with a training block resolution of 32, whereas the second row depicts results with training block resolution of 64.	259
10.17	Rate-distortion performance of the color-only network, for different testing grid resolutions, according to color metric YUV-PSNR. In parenthesis, the training data resolution that was used for the learned model.	259
10.18	Rate-distortion performance of the color-only network, for different input color spaces, according to color metric YUV-PSNR.	260
10.19	Rate-distortion performance of the color-only network, for different loss functions, according to color metric YUV-PSNR.	261
B.1	Octree data structure decomposition.	286
B.2	Octree-based compression. Illustration from (Kammerl et al., 2012).	286
B.3	Quantization steps.	287
B.4	Voxelization.	288
B.5	Illustration of visual effects for different color mapping techniques in voxelization.	290
C.1	Reference mesh contents (in parenthesis the number of faces).	292

List of Figures

C.2	Average normal estimation error in degrees. A different normal estimation algorithm is displayed in each row, while the same model is presented across a column.	294
C.3	Average number of points per neighborhood, for <i>guanyin</i> , <i>roy</i> and <i>vase</i>	295
D.1	Illustration of shader interpolation. Figure from (Schütz and Wimmer, 2015). .	302
D.2	Color coded block diagram with scene dependencies to enable corresponding evaluation protocols.	303
D.3	Example of the PointXR toolbox scene for adjusting the rendering configurations of a model.	304

List of Tables

2.1	A categorization of subjective studies in terms of model [G: geometry, C: colored], motion [S: static, D: dynamic, B: both], distortion [GN: geometry noise, CN: color noise, DN: de-noise, D: down-sampling, O: octree pruning, C: compression, S: streaming], rendering [Min-P/T: minimum size points/triangles, Fix-P/Q/E/C: fixed size points/quads/ellipsoids/cubes, Adp-P/Q/C: adaptive size points/quads/cubes, Prj-V: projected voxels, Mesh: reconstructed mesh], inspection [I: interactive, P: passive], and protocol [DS-Seq/Sim: double stimulus sequential/simultaneous, SS: single stimulus, PC: paired comparison]. <i>Unk</i> stands for unspecified.	26
2.2	A categorization of objective quality metrics in terms of class [FR: full-reference, RR: reduced-reference], domain [P: point-based, I: image-based], inputs [L: location, N: normals, C: curvatures, RGB.: red-green-blue color, TI: texture image, DI: depth image], and features . The character "&" indicates AND, while " " denotes CONDITION with optional inputs referred on the left side.	31
3.1	Geometric characterization of the reference point cloud contents.	37
3.2	Configuration parameters for Octree-pruning. With *, we annotate contents that are employed only during the training.	39
3.3	Performance indexes to compare the test methods (DSIS is set as the ground truth).	46
3.4	Performance indexes to compare the test methods (ACR is set as the ground truth).	47
3.5	Performance indexes to compare display devices (Desktop scores are set as the ground truth).	53
3.6	Performance indexes to compare display devices (HMD AR scores are set as the ground truth).	54
3.7	Equipment details per laboratory.	57
3.8	Subjects information per laboratory.	59
3.9	Performance indexes without using any fitting function for inter-laboratory correlation (Bold text represents the ground truth)	61
3.10	Performance indexes to compare the rendering schemes (Bold text represents the ground truth).	63
4.1	Point cloud contents characterization.	69

List of Tables

4.2	Geometric description of every reference content.	78
4.3	Percentage of discarded points, and geometry and color bpp per encoded stimulus.	79
4.4	Performance indexes to compare the rendering solutions (Bold text represents the ground truth).	89
5.1	Encoding configurations per model.	96
5.2	Rendering configurations per model.	97
5.3	Performance indexes to compare RAHT against Lifting (ground truth) color codec.	101
5.4	Performance indexes to compare the DSIS variants (Bold text represents the ground truth).	102
5.5	Point cloud contents characterization.	107
6.1	G-PCD: Performance indexes of objective quality metrics under Gaussian noise, for both test methods.	138
6.2	G-PCD: Performance indexes of objective quality metrics under Octree-pruning, for both test methods.	139
6.3	J-PCED2: Performance indexes of objective quality metrics. For plane-to-plane, the best-performing configuration per normal estimation algorithm is reported, using the following notation: [fitting surface, neighborhood configuration]. . .	142
6.4	M-PCCD: Performance indexes of objective quality metrics. For plane-to-plane, the best-performing configuration per normal estimation algorithm is reported, using the following notation: [fitting surface, neighborhood configuration]. . .	145
6.5	IRPC: Performance indexes of objective quality metrics. For plane-to-plane, the best-performing configuration per normal estimation algorithm is reported, using the following notation: [fitting surface, neighborhood configuration]. . .	148
6.6	J-PCED2: Performance indexes of objective quality metrics. For PointSSIM, the best-performing configuration is reported, using the following notation: [attribute, voxel depth, dispersion estimator, neighborhood size].	161
6.7	M-PCCD: Performance indexes of objective quality metrics. For PointSSIM, the best-performing configuration is reported, using the following notation: [attribute, voxel depth, dispersion estimator, neighborhood size].	164
6.8	IRPC <i>rpoint</i> : Performance indexes of objective quality metrics. For PointSSIM, the best-performing configuration is reported, using the following notation: [attribute, voxel depth, dispersion estimator, neighborhood size].	167
6.9	IRPC <i>rcolor</i> : Performance indexes of objective quality metrics. For PointSSIM, the best-performing configuration is reported, using the following notation: [attribute, voxel depth, dispersion estimator, neighborhood size].	170
7.1	Performance indexes of image-based quality metrics per type of content, under voxel-based rendering.	181

7.2	Performance indexes of image-based quality metrics per type of content, under splat-based rendering.	182
7.3	Performance indexes of image-based quality metrics per type of content, computed on the entire model views and after background removal, under splat-based rendering.	184
7.4	Performance indexes of MS-SSIM per type of content, computed on model views from different camera layouts after background removal, under splat-based rendering.	186
7.5	Performance indexes of image-based quality metrics per type of content, computed on the navigation tracks of the users and by using the camera layout with $K = 6$ model views.	190
7.6	Performance indexes of MS-SSIM per type of content, computed by including (WAVG) and excluding (AVG) user interactivity information on model views obtained under all camera layouts.	192
7.7	Percentage of viewpoints with non-zero weights under all camera layouts and average time of users inspection, per content.	193
8.1	Performance indexes computed over the entire data set.	201
8.2	Performance indexes computed over stimuli clustered per codec.	205
8.3	Performance indexes computed over stimuli clustered per type of content.	207
8.4	Performance indexes computed over each content's variations (mean \pm standard deviation).	209
9.1	Summary of content retrieval information, processing, and point specifications.	214
9.2	Performance indexes depicting the correlation between subjective scores from the participating test laboratories (Bold text represents the ground truth).	223
9.3	Results of the Welch's t-test performed on the scores associated with color encoding module Lifting and RAHT, for geometry encoder Octree and TriSoup and for every degradation level. The number indicates the ratio of contents for which the color encoding module of each row is significantly better than the module of each column.	228
9.4	Selected encoding parameters of G-PCC for experiment 2, for high and low target bit-rates. The depth parameter indicates the resolution of the Octree structure, whereas the level parameter indicates the TriSoup approximation.	230
9.5	Selected encoding parameters of G-PCC for experiment 3, for high and low target bit-rates. The depth parameter indicates the resolution of the Octree structure, whereas the QP parameter indicates the quantization parameter for the Lifting encoding module.	235
10.1	Selected values of λ_g and λ_c for the computation of the loss function as in Equation 10.4, to achieve various distortion allocation schemes with ratios $\lambda_g : \lambda_c$, for different bitrate values (from smallest to largest, R1 to R4).	250

1 Introduction

Innovations in emerging technologies to capture and consume immersive media today lay the foundation for a future of richer information exchange and enhanced quality of experience. Nowadays, 3D sensing technologies enable real-time acquisition of topological information of the scene-space with high accuracy. Such information can be exploited in application domains spanning from computer vision and robotics, to real-time communications. The significant progress witnessed in extended reality (XR)¹ technologies gives rise to applications ranging from entertainment and gaming, to education and psychology. The envision of experts on the field for the near future is the development of new formats based on the plenoptic function for the representation of 3-D visual information that will provide richer portrayals to better mimic real world sceneries (Ebrahimi et al., 2016). Such advancements will fuel the development of new-generation imaging and communication systems, bringing new challenges and exciting immersive experiences.

In this context, point cloud imaging denotes an attractive option for advanced content representation. A point cloud is defined by a set of points, which are determined by their x , y and z coordinates and characterize the content's topology in the 3-D space. The appearance and properties of the underlying surface can be determined by attributes that optionally accompany the location data, such as color, normals, curvatures and reflectance among others. In essence, a point cloud can be interpreted as an organized or unorganized data structure, which is obtained from a regular or irregular sampling of the surface of a 3D model. The coordinates indicate the spatial position of the samples, while associated attributes provide information that describes local surface properties (e.g., shape, texture, etc.).

The advent of high-quality depth sensors today provides the means for capturing depth-enhanced data. Moreover, the ample accessibility of high-performing graphics processing units (GPU) simplifies the processing, encoding, and rendering of 3D content. Such advances establish the ground for imaging modalities that are well-suited for real-time applications, such as point clouds, to thrive. This explains the reasons why this type of visual data has lately

¹Extended, or cross reality (XR) term is employed to collectively refer to augmented, mixed and virtual reality (AR/MR/VR) technologies.

attracted a strong interest by the scientific community and industrial partners. This interest can be confirmed from relevant activities of the JPEG and MPEG standardization bodies that have been taking place the last few years. As a result of such efforts, notable progress in compression technologies and quality evaluation frameworks has been attained, while MPEG has developed the first point cloud compression standards (Schwarz et al., 2019) and JPEG has recently issued a Call for Evidence in the context of the JPEG Pleno framework (WG1, 2020). Released standards are expected to simplify interoperability across devices and lubricate the integration of this technology in daily use-cases. The interested reader can refer to a recent JPEG document “Use cases and requirements” (Perry, 2018), for a comprehensive summary of target applications for point clouds.

Point cloud imaging offers a number of advantages mainly laying on the flexibility it offers in every processing step from acquisition to rendering. However, these qualities come at the cost of a vast amount of information that is required to faithfully represent 3D models. Thus, efficient data structures and compression algorithms are inevitable. Compression methods aim at reducing the data requirements for storage and transmission. However, lossy schemes, which grant larger reductions, result in visual degradations that affect the perceived quality of a model and, in turn, the user experience. Thus, it is of critical importance to define adequate and reproducible frameworks to accurately evaluate the impact of content distortions, providing us with the means to achieve a fine balance between visual quality and data size. For this purpose, quality assessment methodologies are required. Relevant methods can be classified as subjective, or objective, and are essential for perceptual quality assessment of degraded models and performance evaluation of encoding engines. Objective quality metrics rely on algorithms that provide predictions of the perceived quality of a distorted content. Subjective quality methods require the participation of observers in experiments to collect individual quality scores for the testing material. Although costly and time-consuming, subjective experiments are widely accepted to unveil ground-truth scores, since they depend on opinions of targeted end-users.

Considering the wide diversity of point cloud uses-cases, the posed challenges differ and should be tackled under different objectives. Specifically, in our work, point clouds are approached as a standalone 3D content representation that can be acquired in real-time for entertainment purposes; a representative example is 3D tele-immersive systems. Thus, from our perspective, aspects related to measuring and predicting perceptual quality, or the establishment of low-complexity rendering schemes with visually appealing results, are relevant and extensively explored in our studies. This is important to note because, for instance, when considering autonomous driving applications, the point cloud data quality is also extremely important, but in a different sense and under a different objective.

In this thesis, we address open questions arising in the field of point cloud quality evaluation and compression. In particular, we propose frameworks for both subjective and objective quality assessment of both colorless and colored point cloud models. For the former, we extend traditional approaches for conventional 2D imaging by incorporating interactivity and

introducing experimental settings that offer different degrees of freedom (DoF) to the end user. For the latter, we develop objective quality metrics that operate on the point cloud domain, in order to predict perceptual degradations that exhibit on topological and textural information. In addition, we evaluate widely popular predictors in 2D imaging by applying the corresponding algorithms on projected views of the models, and we propose practices to improve their performance. The state-of-the-art point cloud encoding engines are benchmarked, and rate allocation between geometry and texture encoding is investigated using subjective opinions from a carefully designed large-scale experiment. Finally, a flexible learning-based compression approach is proposed that can handle the encoding of both geometry information and color attributes of point cloud contents.

1.1 Contributions

Our contributions can be clustered in three main parts, which coincide with the organization of the manuscript. In the first part, we explore and define subjective methodologies to measure the perceptual quality of point cloud contents. In the second part, we design and evaluate objective quality metrics to predict the perceptual quality of point cloud contents. In the third part, we assess state-of-the-art encoding engines and introduce our own deep learning-based solution.

1.1.1 Measuring perceptual quality

In this part we experiment with several subjective evaluation frameworks and methodologies that are tailored to the richer nature of 3D content representations. We initiate our efforts by investigating the impact of adopting different test methods, which have been extended to incorporate interactivity, for quality assessment of point clouds. We then proceed and evaluate a radically different approach in an AR inspection scenario with 6DoF using the same testing material. In these experiments, the models are displayed as collections of point primitives. Provided the ambiguity in the perception of the underlying shape by the usage of raw point samples, we conduct a subjective experiment after converting the point cloud data to meshes for display purposes. In the aforementioned studies, point cloud topology is only considered in order to reduce the parameter space, since the textural information might act as distractor.

In a later stage, we experiment with higher resolution point clouds enriched with color attributes in order to account for more realistic use-cases of 3D modelling. For visualization purposes, we develop several point-based rendering schemes. In particular, a splat-based rendering method is initially implemented, which relies on a pre-processing step to assign a geometric shape and size to each point sample. Using this solution, we investigate the impact of the geometric shape selection based on human preferences. Moreover, this renderer is employed for quality evaluation of point cloud models compressed under a well-established codec using different combinations of geometric and textural distortions. The same stimuli are assessed using a second rendering solution, which relies on real-time voxelization, and

projection of the obtained voxel grid to a displayed image. In the first rendering scheme, each splat size can be adapted to local point densities, which ensures visualization of watertight models, but leads to coarser surface approximations for sparser content representations. In the second implementation, all voxels are mapped to fixed-size pixel neighborhoods, thus, visual artifacts in the form of missing pixels are perceived for sparser models. The subjective scores obtained from both experiments are compared to draw conclusions regarding their statistical equivalence and to identify potentially diverging rating trends, in response to the different nature of rendering artifacts.

In a third stage, we exploit VR systems to provide a fully-controlled testing environment for subjective evaluation of point cloud contents. For this purpose, we initially design a suitable scene that grants a high sense of realism with minimal distractions. The user consumes the virtual world by means of a headset and can interact with 6DoF via physical movements or using the VR controllers. The point clouds are displayed using a point-based rendering approach that can be configured off-line for visual adjustments per model. In this platform we conduct subjective experiments for quality assessment of state-of-the-art color encoding engines using two protocols; that is, an interactive extension of a conventional test method, and a newly introduced variant. Moreover, the same VR setting is employed in an eye-tracking experiment that is conducted in order to quantify the visual attention of users in immersive scenes under a task-dependent protocol. After integrating and calibrating a dedicated hardware in the headset, head and gaze data cues are recorded during inspection of point clouds by users. A methodology to utilize highest-quality gaze measurements is developed based on a per-session error profiling and a scheme to decide the fixation areas on the observed point clouds is proposed. Finally, a prototype of a virtual museum is implemented that allows more realistic human interactions, while also previously developed functionalities for recording and analysis of user behavior are improved and discussed.

1.1.2 Modelling perceptual quality

In this part we describe and evaluate proposed solutions of perceptual quality predictors for point cloud contents, which can be distinguished in point-based and image-based. Regarding the first class, we initiate by describing an algorithm based on the similarity of local surface approximations of underlying shapes, reflected through normal vectors. This algorithm is based on the angular similarity of tangent planes that correspond to associated point samples between the pristine and the distorted model. Provided that the method relies on normal attributes, our performance analysis is conducted using widely adopted normal estimation algorithms and different configurations. The impact of the estimated measurements is quantified and insights are obtained together with best practices for higher prediction accuracy. Our efforts are then focused on the design and performance evaluation of a family of features that are computed using estimators of a distribution's dispersion. These features are extracted from local neighborhoods and capture the local deviations of quantities that are defined per attribute, including topology and color information. To obtain a quality score for the level of

degradation of the corresponding attribute, an error value is computed by comparing features that are obtained from associated neighborhoods between the pristine and the distorted model, simulating the working principle of the structural similarity index (SSIM) (Wang et al., 2004). As part of the metric, a voxelization step is proposed, and can be optionally enabled prior to feature extraction, in order to simulate distant inspection. The performance of the proposed scheme is evaluated under different attribute selection, voxel resolution, dispersion estimator, and neighborhood size, using several subjectively annotated data sets as ground truth.

Regarding image-based approaches, we assess the prediction power of 2D imaging quality metrics on projected views of point cloud contents. For this purpose, we define an objective quality evaluation framework and acquire views of the point clouds from different camera layouts, using the same rendering scheme that was employed for user consumption. To examine generalization capabilities, the image-based metrics are benchmarked against two subjectively annotated data sets that are comprised of the same models, evaluated under two different rendering schemes. The impact of removing irrelevant background information is gauged, before proceeding to the exploration of potential benefits by enabling additional model views in the computation of the metrics. Moreover, we compute objective quality scores that consider the entire navigation experience of users during subjective evaluation, and evaluate their prediction power. Finally, we devise an ad-hoc methodology to weight model views that are captured from the camera layouts of our quality evaluation framework according to user inspection duration, showing promising results.

In the last chapter of this part, we rigorously evaluate the performance of current point cloud objective quality metrics. For this purpose, human scores that were obtained from a large-scale subjective quality evaluation campaign of the state-of-the-art MPEG point cloud compression solutions serve as the ground truth.

1.1.3 Towards efficient compression

In this part, we initiate by conducting a subjective experiment to evaluate the performance of the state-of-the-art encoding engines that have been developed in the context of the MPEG standardization activities. A point-based rendering scheme is developed and integrated in an evaluation framework that can support interactivity between the users and the displayed models. The evaluations are performed in two dislocated laboratories that participate in the efforts. The performance of the tested encoding algorithms is evaluated and useful insights are provided regarding strengths and weaknesses of each approach. Moreover, experiments to draw conclusions regarding the allocation of bits in geometry and color encoding of the MPEG compression solutions that operate on the point cloud domain are conducted.

Lastly, a deep learning-based approach is proposed to encode both the topology and the texture of point clouds. The network architecture is flexible in encoding one, or both of the aforementioned attributes. This gives rise to compare two different schemas; that is, a

unified network, which is trained on colored point clouds and encodes geometry and color as a holistic representation, against a combination of two separately trained networks, with the first dedicated to geometry and the second to color compression. Moreover, the impact of different network parameters is explored in the performance of our solution, such as the training data set, the color space, and the loss function selection.

2 Related work

In this chapter we report relevant work on point cloud imaging. We refer to technologies and methods from acquisition to rendering, with a particular focus on domains that are covered in a higher depth throughout this dissertation.

2.1 Acquisition

The acquisition of 3D point clouds has been widely investigated over the last decades. There is a large range of technologies for point cloud capturing, which can be clustered in a variety of ways. Hereafter, we refer to the most popular ones nowadays, and classify them based on the nature of sensors they require as (a) passive, and (b) active techniques.

2.1.1 Passive techniques

Passive techniques do not interfere with the model, rather, they rely on recorded information, most commonly in the form of 2D imaging, depicting the reflected energy from a model.

Stereo vision methods are falling in this category and rely on two or more cameras that capture a scene. In the simplest case, two instances are sufficient, however, techniques in order to increase the accuracy using multiple images have been available for several years (Okutomi and Kanade, 1993). For high quality results, the cameras need to be calibrated; that is, camera parameters need to be either known in advance or estimated. From the captured images, feature points are extracted and matched through automatic algorithms that fall in the category of the “correspondence problem”, which has been extensively investigated in the literature (Scharstein and Szeliski, 2002). To simplify the procedure, it is very common to apply image rectification in order to project the images onto a common plane, and from the disparity map that occurs, the depth is estimated through triangulation (Hartley and Sturm, 1997). The quality of the point cloud data depends on the solution to the correspondence problem (Dhond and Aggarwal, 1989), the camera calibration procedure, as well as properties of the scene space (e.g., shape, reflectivity). A relevant technique is the so-called structure-

from-motion (Westoby et al., 2012) that makes use of a large number of monocular images taken over different time frames from different viewpoints, typically, under the assumption of a stationary scene. In the general case, the camera pose and the scene geometry need to be estimated simultaneously, thus, resulting in a more challenging problem. When the computations are performed in real-time, often required in robotics community, this problem is referred to as Simultaneous Localization And Mapping (SLAM) (Durrant-Whyte and Bailey, 2006).

Photogrammetry techniques are similar, in the sense that they are based on the same principle of extracting a 3D point cloud from two or more 2D images that are taken from different positions and capture the same scene. In this case, the camera calibration and orientation is critical. Image measurements are identified and employed for matching purposes. These measurements can be obtained either through automatic, or semi-automatic, or manual processing with the intervention of an operator often assisting in obtaining more accurate results (Remondino and El-Hakim, 2006). The working principles of stereo vision and photogrammetry techniques share similarities. However, the relevant approaches are developed from different communities, namely, computer vision and photogrammetric, and reflect different objectives. In the former case the main goal is the automation of the procedure, which implies relaxations on the processing pipeline, whereas in the latter case, the main objective lays on obtaining highly accurate results (Hartley and Mundy, 1993).

There are several other less common methods that are based on captured images, such as *shape from texture*, *shape from focus/defocus*, *shape from specularity*, *shape from shadows*, *shape from shading*, *shape from silhouette*, *shape from contours*, and *shape from edge gradients* (Sansoni et al., 2009; Pavlidis et al., 2007; Remondino and El-Hakim, 2006; Mada et al., 2003). However, they typically lead to low resolution point cloud data and, thus, they are not further analysed.

2.1.2 Active techniques

Active techniques make use of the received energy after emission of properly formatted light, or any other form of electromagnetic energy that interacts with the model.

A *time-of-flight* camera is a range imaging camera system that estimates the distance of a surface based on the speed of light. In particular, a transmitter unit emits a laser pulse, and a receiving sensor detects its reflection (Kolb et al., 2010). The distance can be estimated based on the measured round-trip time. The emitted pulse may also be modulated, and in this case, a phase shift analysis is performed between the emitted and detected light to resolve the distance (Gokturk et al., 2004). The time-of-flight technology is employed in Light Detection And Ranging (LiDAR) scanners. The two terms are often confused and used as interchangeable. For a review on methods and applications of time-of-flight cameras, the interested reader can refer to (Hansard et al., 2012).

Laser scanning systems are based on emission of a laser from a source in the form of a point, or

a beam that impinges on the surface of the model, and its reflection is detected by a dedicated camera. The triangulation principle is employed in this technology in order to extract depth information (Chen et al., 2000). High accuracy and low vulnerability to illumination conditions are characteristic advantages of these systems.

The *structured light* technology is based on a projector that is producing specific patterns, and is working in cooperation with one or more cameras that capture the projected pattern. This method is also part of the triangulation-based family systems, however, the basic working principle lies on the deformation of the projected patterns due to the 3D shape of the objects. There is a multitude of different patterns proposed for this task, such as fringes, stripes, grid or dot designs, complex patterns with curves, which are time, space, or color coded. For a comprehensive review of structured light theory, patterns, applications and design, the interested reader may referred to (Geng, 2011; Salvi et al., 2004). A key advantage of these methods when compared to passive photogrammetry is the ability to determine planar surfaces, such as walls and floors, due to the usage of projected light patterns. It should also be noted that they are typically more sensitive to lighting conditions, when compared to laser scanners. Finally, for both of them, particular care should be devoted to transparent and highly reflective surfaces.

2.1.3 Discussion

The above-mentioned methodologies are not mutually exclusive. For instance, in one of the most widely cited surveys that was conducted for the digitization of exposed statues, the so-called Digital Michelangelo Project (Levoy et al., 2000), laser scanners, time-of-flight and digital still cameras were used in order to acquire, align, and merge scanned data. Nowadays, photogrammetry and computer vision algorithms are used by companies, such as Pix4d¹, to transform 2D imagery that may be captured even from drones into 3D maps.

For smaller-scale applications, the development of RGB-D cameras witnessed in the last few years, is also noteworthy. In particular, after 2010 and the release of Kinect v1, a significant interest from the industrial sector has been observed, which today is translated to the wide availability of low-cost depth sensing technologies. The Intel RealSense suite and the Kinect line of sensors are widely used as computer peripherals. These technologies make use of stereo vision, time-of-flight, structured light, or hybrid systems that involve more than one technologies, which collaborate with synchronized RGB cameras in order to append depth values to the RGB color of the pixels. For instance, Kinect v1 measures the depth using structured light and Kinect v2 is based on the time-of-flight technology (Wasenmüller and Stricker, 2017). The RealSense R200 makes use of one RGB camera that is synchronized with two left-and-right calibrated cameras and an infra-red projector that emits a dot pattern, in order to provide depth-enhanced RGB information (Keselman et al., 2017). Nowadays, depth sensing technologies are also integrated in mobile phones, such as the new-generation Apple,

¹<https://www.pix4d.com/>

Samsung and Huawei devices.

Point clouds can be also artificially generated either algorithmically or by hand, from various software applications available in consumer market. There is a wide range of relevant software packages that can be employed today, such as Blender, 3D CAD (Computer-Aided Design), AutoCAD, Rhino3d, Sketchup, MeshLab, Point Cloud Library (PCL), and MATLAB, to name a few. Finally, deep learning methods can be exploited for point cloud data production. Generative neural network approaches have been proposed that operate directly on the point cloud domain (Achlioptas et al., 2018), while also, 3D shape reconstruction using mesh modelling can be performed from a single RGB image (Wang et al., 2018a; Ge et al., 2019). Considering both ad-hoc generated models and deep learning applications that represent 3D visual information in the form of meshes, point clouds can be obtained either by exporting the vertex information, or by sub-sampling the faces of the exported mesh files.

In the context of this thesis, we employ well-established point cloud data sets that are captured from active sensing technologies in real-life, or from simulated acquisition scenarios. Moreover, for experimentation purposes, we create artificial models that depict simple geometric shapes using mathematical equations, and generate high-quality point cloud data by sub-sampling mesh contents.

2.2 Compression

The prevalent approaches that have been widely explored in the literature for point cloud compression can be clustered as model-based and projection-based. More recently, deep learning-based solutions were proposed and are expected to gain significant popularity in the near future.

2.2.1 Model-based encoding

Model-based approaches operate on the point cloud data domain and can be further subdivided to geometric and attribute encoding algorithms.

Geometry encoding: Compression of point cloud topology relies on efficient data structures, with octrees (Jackins and Tanimoto, 1980; Meagher, 1982) denoting the most common selection nowadays. One of the early works on the field presented in (Gumhold et al., 2005), is based on a prediction tree that is built using a greedy algorithm, which aims at minimizing residual errors. In (Merry et al., 2006), a single-rate encoding scheme is proposed and a spanning tree is constructed based on multiple geometry predictors to exploit correlations between neighbors. A multi-resolution predictive coding method is described in (Waschbüsch et al., 2004) that relies on identification of pairs, which is interpreted as a graph matching problem in their context. From these pairs, representative samples are defined, and residual errors from the

original points are estimated and encoded.

The kd-tree decomposition is introduced in (Gandoin and Devillers, 2002) to recursively split the space and encode the number of points that lies in one out of two partitions. This approach is adapted in (Peng and Kuo, 2003) to make use of octree structures and occupancy maps for the encoding of the points positions. In (Schnabel and Klein, 2006) and (Huang et al., 2006), the octree structure is exploited in progressive compression schemes that rely on approximations of the underlying surfaces to predict neighboring occupancy. The latter study is extended in (Huang et al., 2008) in order to handle compression of color attributes.

An extension paradigm to dynamic sequences using the exclusive disjunction operator for inter-frame prediction is presented in (Kammerl et al., 2012) and implemented in PCL (Rusu and Cousins, 2011), with each frame represented by an octree. In (Garcia and de Queiroz, 2017), temporal relationships among frames are additionally considered for lossless prediction by reordering each octree based on previous frames, before entropy coding. In (de Queiroz and Chou, 2017a), the motion-compensation encoding concept is extended from video to the dynamic point cloud domain. A voxelized point cloud is split into blocks and each block is encoded using an intra, or a motion-compensation mode based on a decision that is taken in the rate-distortion sense, respecting low complexity requirements for real-time operation.

A context-based lossless intra-frame encoding method is reported in (Garcia and de Queiroz, 2018), using the parent values and parent positions of an octree sequence, which denotes an ordered occupancy map of the octree data structure. In another recent line for static models compression, denser shape approximations are enabled after octree decomposition by reconstructing the underlying surface of a model using triangular primitives, also known as “Triangle Soup” (TriSoup), as described in (Pavez et al., 2018). Moreover, the usage of planar surfaces is proposed in (Dricot and Ascenso, 2019), graph-based geometric enhancements are introduced in (de Oliveira Rente et al., 2019) and volumetric functions are employed in (Krivokuća et al., 2020).

It is noteworthy that an octree-based and a TriSoup-based implementation are integrated in the MPEG Geometry-based Point Cloud Compression (G-PCC) codec (MPEG 3DG, 2019).

Color encoding: Color and potential compression of additional attributes were natively handled together with geometry by some of the early studies on the field (Waschbüsch et al., 2004; Schnabel and Klein, 2006; Huang et al., 2008). More recent algorithms are dedicated to topology compression, leaving space for the development of color-only encoding solutions. The latter are typically applied on either the uncompressed, or the restored (i.e., decompressed) geometry.

Color attribute encoding using Graph Fourier Transform (GFT) was initially presented in (Zhang et al., 2014) and further extended in (Shao et al., 2017), by enabling Laplacian sparsity, as well as in (Thanou et al., 2016) for inter-frame encoding of dynamic sequences. A 3-D intra

prediction scheme based on neighboring blocks is described in (Cohen et al., 2016b), followed by a modified version of the shape-adaptive Discrete Cosine Transform (DCT) to handle missing points, before quantization. The work is extended in (Cohen et al., 2016a) by devising an algorithm to compact the blocks before applying any transform, and modifying the neighborhood identification to create the graph. The Region Adaptive Hierarchical Transform (RAHT) based on the Haar wavelet transform is introduced in (de Queiroz and Chou, 2016) offering a high-performance solution with significant complexity reductions. In (de Queiroz and Chou, 2017b), the Gaussian Process Transform (GPT) is employed to exploit geometry correlations.

The algorithm described in (Zhang et al., 2018) is based on a hierarchical segmentation that is resolved by an initial global color-based and a subsequent local geometry-based segmentation to compile points in clusters that attain similar characteristics. A virtual adaptive sampling process is proposed in (Hou et al., 2017), enabling a sparse representation formulation for recovering the color values of occupied voxels in block partitions. The previous work is extended in (Gu et al., 2020b) by incorporating an inter-block prediction strategy and an entropy coding scheme for the transform coefficients. In (Gu et al., 2020a), based on the assumption that adjacent points share color similarities, representative points from previously encoded clusters are selected and employed to predict color values of points in the current cluster, making use of a graph structure that reflects the underlying geometry. A graph transform is used on top for the residuals. In (Chou et al., 2020), volumetric functions are employed to encode color attributes. A hierarchical structure is detailed in (Mammou et al., 2017) with points that belong to a lower layer being used to predict attributes at a higher layer of details. This scheme is further improved in (Mammou et al., 2018) by introducing a Lifting step.

The color codecs described in (de Queiroz and Chou, 2016) and (Mammou et al., 2018) are integrated in the MPEG G-PCC, and can be used in combination with any of the two available geometry encoding modules.

2.2.2 Projection-based encoding

Projection-based algorithms operate on the image domain and exploit the high performance of 2D imaging codecs, which are applied on projected views of point clouds.

In this category falls an early study on the field described in (Ochotta and Saupe, 2004), which is based on encoding of regularly sampled height fields from surface patches (i.e., point clusters that are partitioned), over base planes. A similar working principle of height fields decomposition is proposed in (Golla and Klein, 2015), with additional usage of occupancy maps. The voxel-based encoding scheme with adjustable level-of-details that is introduced, makes the algorithm suitable for real-time systems.

In (Houshiar and Nüchter, 2015), traditional 2D imaging compression algorithms, such as PNG, TIFF, JPEG and JPEG2000 are applied on panorama images that are produced after equirectangular projection of point cloud depth, color and reflectance values. Depth maps

are employed to represent point cloud data in (Bletterer et al., 2016) and a multi-resolution analysis is described for progressive encoding. An end-to-end tele-immersive system is proposed in (Mekuria et al., 2017a), exploiting the JPEG coding engine to encode the color of points that are projected onto planar surfaces in a depth-first tree traversal order.

In (Mammou et al., 2017), a patch-based point cloud projection is proposed, where the patches are assembled in a video sequence. This work essentially established the basis of the emerging MPEG Video-based Point Cloud Compression (V-PCC) test model (MPEG 3DG, 2020). The latter employs HEVC to encode the two video sequences that are generated to capture geometry and texture information of a point cloud. Additional metadata to reconstruct the model are compressed separately. In recent studies, algorithms to improve the encoding efficiency of V-PCC are proposed, based on appropriate padding of the projected patches (Li et al., 2020a), and better predictions of the motion vector (Li et al., 2020b).

2.2.3 Deep learning-based encoding

Deep learning architectures were recently employed for compression of visual data representations, showing promising results. The success and efficiency that has been observed in 2D imaging modalities has driven the interest for extending these approaches in point cloud imaging, which denotes a higher-dimensionality and irregular content representation.

The majority of deep-learning approaches for point cloud imaging are currently based on auto-encoding architectures that target compression of geometry-only information, which is realized in a block-by-block basis. In particular, one of the first attempts is reported in (Quach et al., 2019), proposing a shallow, yet efficient architecture composed of convolution and de-convolution layers for analysis and synthesis, respectively. Differentiable rate and quantization estimations are approximated (Ballé et al., 2016), and the focal loss is employed to obtain a quality score for the reconstructed model. In a more recent work (Quach et al., 2020b), the impact of several parameters added to the initial network version (Quach et al., 2019) is evaluated through a series of experiments. Among the additions, a hyper-prior model and deeper transforms, as well as the fine tuning of the balance weight employed in the focal loss and adaptive thresholding, were found to improve the performance.

Another early study on the field is presented in (Guarda et al., 2019b), which also adopts a small number of convolution and de-convolution layers for analysis and synthesis. Quantization is performed on the latent representation, and the result is entropy coded. The weighted binary cross entropy is used to measure the reconstruction error in the loss function. This study provides a detailed description of the key stages of an auto-encoder network architecture. Moreover, performance evaluation results show that a larger number of filters per layer is only beneficial at larger bit-rates. The same authors extend their efforts in (Guarda et al., 2019a) and conduct rate-distortion performance analysis on the latent space using the same network (Guarda et al., 2019b). In (Guarda et al., 2020), the network architecture is enriched with a hyper-prior and the possibility of explicit quantization via down-scaling and upscaling,

before and after feeding the latent representation to the Variational Auto-Encoding (VAE) module, respectively. Benchmarking results indicate the presence of a sweet spot when using implicit (governed by the selection of λ in the loss function) and explicit quantization that can lead to significant computational reductions.

In (Wang et al., 2019), a deeper auto-encoding architecture is proposed, based on 3D convolution layers stacked with Voxception-ResNet (VRN) structures and a hyper-prior implemented as a VAE. Several pre-processing steps are employed including voxelization, scaling and partition before feeding a point cloud in blocks to the network, and the weighted binary cross-entropy is used to estimate the reconstruction loss, similarly to (Guarda et al., 2019b). The performance of this network shows promising results, achieving comparable, if not better performance when compared to V-PCC. Experimentation with different partition sizes and adaptive thresholding for classification of a voxel as occupied or not, are part of the study. A multi-scale hierarchical encoder is proposed in (Huang and Liu, 2019) based on local features that are extracted at each layer. A sparsity term employed in the loss function enables sparse coding and higher efficiency in encoding point cloud geometry.

The aforementioned studies are handling point clouds as 3D occupancy maps on regular grids. In (Yan et al., 2019), raw point clouds are fed to the proposed architecture, which makes use of the PointNet (Qi et al., 2017) to extract features from unorganized coordinates in 3D space. The synthesis transform is represented by a generative fully-connected network, using the Chamfer distance in the loss function.

A study on the compression of point cloud attributes is introduced in (Quach et al., 2020a), relying on folding a 2D grid onto a point cloud and then mapping the attributes on top of it. Thus, the 3D is converted to a 2D encoding problem, provided the estimation of accurate parametric functions for folding and low-distortion attribute mapping techniques. An advantage of this approach is the usage of highly efficient 2D imaging techniques for point cloud compression; yet, a bottleneck is the low accuracy of the folding in the geometrically complex parts of a model. In (Alexiou et al., 2020a) geometry and/or color information is encoded directly in the 3D domain by extracting features from regular grids exploiting 3D convolutions; thus, spatial redundancies are captured for both types of information. Moreover, the influence of a series of encoding parameters is evaluated.

2.2.4 Discussion

The MPEG V-PCC model has recently obtained the Final Draft International Standard (FDIS) status (MPEG 3D Graphics Coding, 2020), whereas the MPEG G-PCC model has been promoted to the Committee Draft (CD) stage, and is expected to reach its final milestone by mid 2021 (MPEG Systems, 2020). The two models are widely considered the state of the art in point cloud compression for dynamic and static point cloud contents, respectively. For excellent recent reviews on coding approaches, the interested reader may refer to (Pereira et al., 2020; Cao et al., 2019).

In the context of this thesis, model-based and projection-based encoding engines are employed to denote realistic types of point cloud degradations. We initiate by using a simple octree encoding for the topology of colorless models, as implemented in PCL (Rusu and Cousins, 2011). The codec proposed in (Mekuria et al., 2017a) and served as the anchor implementation in the MPEG point cloud compression activities, while also the state-of-the-art MPEG V-PCC (Mammou, 2017) and G-PCC (Mammou et al., 2019) variants are extensively used in our experimentation efforts for colored models.

Contributions: Our contribution to the state-of-the-art in point cloud compression is the following paper: (Alexiou et al., 2020a), which introduces a learning-based approach for geometry and color encoding, and is detailed in chapter 10.

2.3 Rendering

Rendering solutions for point cloud data span over a wide range, which can be clustered in multitude ways. In this section we refer to technologies that have been developed for visualization of contents that are represented by point samples. We make a coarse distinction and split them in two of the most popular research lines in computer graphics community, namely, point-based and mesh-based rendering.

2.3.1 Point-based rendering

In this category fall several techniques that explicitly employ point rendering primitives without connectivity information to display a model.

In a pioneering work, Levoy and Whitted (Levoy and Whitted, 1985) were the first to propose the use of points in computer graphics, stating that points in 3D should be viewed analogously to pixels in 2D. Among the early studies, a point-based rendering technique is proposed in (Grossman and Dally, 1998), where objects are represented by points with depth and color information that are obtained after off-line sampling. During run-time, the points are projected onto a pixel grid using orthographic projection. In (Pfister et al., 2000), the use of viewer-facing discs, namely surfels, is introduced, extending the previous effort. In the simplest case, a surfel would be just a point in 3D space with a constant color value. Additional attributes can be optionally associated, such as normal, radius and shading. Appropriate sampling and filtering is performed in a pre-processing step, with the resulting samples being arranged in a hierarchical octree data structure. Note that in the above schemes, the reconstruction of the displayed image (i.e., hole filling) is performed in the image space. In the same line, a more recent work is presented in (Marroquim et al., 2007), proposing a new image reconstruction scheme with a hardware-accelerated implementation.

Zwicker et al. (Zwicker et al., 2001) proposed the use of surface splatting to mitigate sampling

issues and discontinuities in the rendered image. This technique is based on overlapping of elliptical primitives that are projected in image space and filtered using low-pass Gaussian kernels, which corresponds to a re-sampling operation. The pixel values incorporate aggregated contributions of surrounding neighbors from this processing before projection onto the screen. This texture filtering technique is also known as Elliptical Weighted Average (EWA) splatting. Appropriate adaptations for GPU-compliant implementations are described in (Ren et al., 2002) and (Guennebaud and Paulin, 2003). Further improvements are introduced in (Zwicker et al., 2004) by accurate splatting in terms of perspective in the image space, and by adding clip lines on the splats to improve sharpness. In (Botsch et al., 2004), Phong splats are proposed using a normal field per-pixel that varies linearly, reportedly leading to visual enhancements. A rendering framework in GPU is described in (Botsch and Kobbelt, 2003), relying on elliptical splat shapes with two-pass Gaussian filtering. A hierarchical point-based rendering method based on an octree decomposition is described in (Botsch et al., 2002), which can be applied to render either points or splats. Exploiting the proposed structure, computational reductions can be achieved with respect to the original implementation of EWA splatting.

Among the most widely cited papers are (Alexa et al., 2001; Alexa et al., 2003), which propose the use of smooth Moving Least Square (MLS) surfaces to approximate the model shape that is defined by point samples. In this case, re-sampling is performed in the object space to adjust the density of the projected points onto the image grid. In (Rusinkiewicz and Levoy, 2000), the QSplat system, one of the most broadly regarded studies is described. The rendering scheme relies on a hierarchy of spheres with different radii to display the model at different resolutions. Every node represents a part of the object's surface with a position, a radius and a normal. The development of QSplat was triggered by the need to render large-scale data sets obtained by the Digital Michelangelo project (Levoy et al., 2000).

An excellent survey on the field of early-developed techniques can be found in (Kobbelt and Botsch, 2004), while in (Sainz and Pajarola, 2004), an effort to compare the performance of primitives and algorithms for point-based rendering is detailed.

In more recent studies, an adaptive splatting approach tailored to facial point sets is described in (Kim et al., 2012) for realistic rendering. A feature detector is employed to exploit the prior information regarding the sensitivity of human perception in facial characteristics, in a re-sampling stage that is followed by splat optimization. A hole-filling algorithm is applied in the image space to avoid perception of missing information. This approach was favored over increasing the splat size, as being less computationally expensive for the context of the aimed application while also reducing the presence of bluriness artifacts. In (Preiner et al., 2012), a real-time rendering approach for dynamic point clouds is proposed. The scheme is based on rendering of surface-aligned splats with size and normal that are estimated on-the-fly in the screen space based on their k -nearest neighbors.

The development of out-of-core hierarchical data structures that make efficient use of GPUs to handle the immediate rendering of massive point clouds with fast responsiveness for

interactive applications has been a relevant research topic for the community. In fact, the studies (Pfister et al., 2000) and (Rusinkiewicz and Levoy, 2000) can be considered as early ancestors of multi-resolution schemes. Sequential data structures adjusted to the current level of detail are introduced in (Dachsbacher et al., 2003), while in (Gobbetti and Marton, 2004), a layered tree structure is demonstrated partitioning the point cloud space into chunks. In (Wimmer and Scheiblauer, 2006), nested octrees are proposed making use of memory optimized sequential data structures, with the same concept applied on a web-based rendering application described in (Schütz, 2016). In a recent study (Schütz et al., 2019), an algorithm that enables rendering with a continuous level of detail is proposed and evaluated in a VR scenario, which constitutes a challenging set-up considering the level of responsiveness that is required.

2.3.2 Mesh-based rendering

This category implies the application of a surface reconstruction algorithm on the point cloud data, which can be enabled either off-line or on-line, in order to display a polygonal mesh at execution time.

The reconstruction of a continuous surface from a set of point samples is an ill-posed problem, in the sense that there is no unique solution. It also denotes a well-studied topic in the literature. The points are typically interpreted as discrete samples and the objective of relevant algorithms is to extract the underlying surface. After surface reconstruction, a model is stored as a polygonal mesh.

One of the most popular early studies on the field is the marching cube algorithm (Lorensen and Cline, 1987). The objective is to reconstruct an iso-surface from a set of points that are regularly sampled. In each cube, the value of each vertex is compared with an iso-value, in order to determine which vertex is inside and outside the surface. A total of 15 unique cases of surface intersections are possible. As a result, the corresponding vertices are outputted. Surface reconstruction from unorganized points (Hoppe et al., 1992) denotes a pioneering work. The proposed algorithm is based on the signed distances of points from the underlying surface, which is locally approximated by corresponding tangent planes. The surface is estimated by the zero level set of the signed distance field.

The screened Poisson surface reconstruction (Kazhdan and Hoppe, 2013) is another well-established algorithm. It ensures watertight models and leads to high-quality smooth output surfaces for good quality input data. The working principle relies on the computation of an indicator function (i.e., a function that encloses an area) and the reconstructed surface is obtained by extracting an appropriate iso-surface. Normal vectors are required with the input data and serve as samples of the gradient for the indicator function. Under this logic, the problem reduces in finding a function whose gradient best approximates the normals of the points, considering also constraints related to the positions of the points. A GPU implementation of the Poisson surface reconstruction algorithm (Kazhdan et al., 2006), which

denotes an earlier version without the usage of positional constraints, is proposed in (Zhou et al., 2011).

The aforementioned, denote a few noteworthy algorithms among a vast number of surface reconstruction approaches that have been developed to infer the underlying shape of a model from point cloud data. The performance of existing methods depends on several irregularities that might be present in the provided sets of points. Sampling density, noise, outliers and missing data are some of the geometric factors that influence the outcome. Moreover, certain approaches depend on the presence of additional attributes, such as oriented, or unoriented normal vectors and how accurately they approximate the underlying shape. In (Berger et al., 2017), an excellent review of surface reconstruction algorithms is provided, clustering existing approaches per prior assumptions, point cloud artifacts, and input requirements, among others.

2.3.3 Discussion

Point-based rendering approaches offer more versatile and higher accuracy solutions to display point cloud data. Under conditions, they are spanning from advanced methods for visualization of highly realistic models, to low-complexity techniques that are friendly to content acquired in real-time. On the negative side, they do not grant continuous silhouettes. The distribution of points is crucial for their performance, and the majority of the schemes require algorithmic-dependent re-sampling before display. In addition, sophisticated techniques commonly introduce computational costs during run-time, for better visual representations.

On the contrary, mesh-based rendering can ensure watertight models under the appropriate selection and configuration of a surface reconstruction algorithm, with visual results, though, depending on the points arrangement. This class of methods is rather efficient in terms of rendering performance. However, they are less flexible in representing complex surface topology or dynamic scenes due to connectivity information, making them less suitable for real-time communication systems. Moreover, relevant algorithms often introduce losses in regard to the original points positions.

In the context of this thesis, we mostly experiment with point-based schemes, while also, a mesh-based rendering approach is used for point cloud visualization purposes. Concerning the latter, the screened Poisson surface reconstruction (Kazhdan and Hoppe, 2013) is selected, due to its ability to produce high-quality mesh models without discontinuities. Regarding the former, algorithms that rely on splats of fixed or adaptive size are employed, denoting low-complexity solutions of minimal overhead for the rendering pipeline. The deployed point-based methods do not alter the location data of the provided contents and can be used for both organized and unorganized point clouds. They might be sub-optimal in terms of visual appearance when compared to more sophisticated point-based or model-based rendering methods. Yet, they are advocated as more appropriate in the aspect of not introducing any ambiguous or lossy step before content display, and are better suited for real-time rendering. The

first is of crucial importance, since the adopted rendering schemes are commonly employed for subjective quality evaluation of point cloud codecs. Involving any additional processing after encoding and before subjective assessment might introduce biases, which should be thoroughly investigated and justified.

Contributions: Our contribution to the existing pool of point cloud renderers is the release of two point-based implementations that are developed as part of the following papers: (Alexiou et al., 2020a,b) for web and VR applications, respectively. The realizations are described in annexes D.3 and D.4, while the corresponding software is employed in experiments detailed in chapters 9 and 5, respectively.

2.4 Quality assessment

Quality evaluation methodologies for 3D model representations were initially introduced and applied on polygonal meshes, which has been the prevailing format in the field of computer graphics.

Subjective tests to obtain ground-truth data for visual quality of static geometry-only mesh models have been conducted in the past, subject to simplification (Watson et al., 2001; Rogowitz and Rushmeier, 2001; Yixin Pan et al., 2005), noise addition (Lavoué, 2009) and smoothing (Lavoué et al., 2006), watermarking (Gelasca et al., 2005; Corsini et al., 2007) and position quantization (Váša and Rus, 2012) artifacts. In (Guo et al., 2016), the perceived quality of textured models under geometric and color degradations is assessed. Subjective evaluation of compression artifacts is conducted in a VR setting and described in (Christaki et al., 2019), using non-textured mesh models that are clustered in two quality groups. In (Gutiérrez et al., 2020), a subjective experiment in MR with 6DoF is performed to assess both geometry and texture encoding distortions.

A significant amount of efforts has been also devoted on the development of objective quality metrics for mesh contents, which can be classified as: (a) image-based, and (b) model-based (Lavoué and Mantiuk, 2015). Widely-used model-based predictors rely on simple geometric projected errors (i.e., Hausdorff distance or Root-Mean-Squared error), dihedral angles (Váša and Rus, 2012), curvature statistics (Lavoué et al., 2006; Torkhani et al., 2012) computed at multiple resolutions (Lavoué, 2011), Geometric Laplacian (Karni and Gotsman, 2000; Sorkine et al., 2003), per-model roughness measurements (Corsini et al., 2007; Wang et al., 2012), or strain energy (Bian et al., 2009). Image-based metrics were initially introduced for perceptually-based tasks, such as mesh simplification in (Lindstrom and Turk, 2000; Qu and Meyer, 2008; Luebke and Hallen, 2001). Only recently their performance in predicting visual quality was evaluated and compared to model-based techniques in (Lavoué et al., 2016). The reader can refer to (Bulbul et al., 2011; Corsini et al., 2013; Lavoué and Mantiuk, 2015) for comprehensive reviews on subjective and objective quality assessment methodologies for mesh modelling.

The rest of this section is focused on the state-of-the-art in point cloud quality assessment. In a first part, subjective evaluation studies are detailed and notable outcomes are presented, whilst in a second part, the working principles of current objective quality algorithms are highlighted.

2.4.1 Subjective quality assessment

The first subjective evaluation study for point clouds is reported in (Zhang et al., 2014), which denotes an effort to assess visual quality of models at different geometric resolutions and different levels of noise that were introduced in both geometry and color. For the former, several down-sampling factors were selected to increase sparsity, while for the latter, uniformly distributed noise was applied to the coordinates, or the color attributes of the reference models. In these experiments, raw point clouds were displayed in a flat screen that was installed in a desktop set-up. The results showed an almost linear relationship between the down-sampling factor and the visual quality ratings, while color distortions were found to be less severe when compared to geometric degradations.

A 3D tele-immersive system is proposed in (Mekuria et al., 2017a) where the users are able to interact with naturalistic (dynamic point cloud) and synthetic (computer generated) models in a virtual scene. In this MR application, subjective experiments were conducted allowing the participants to navigate in the virtual environment through the use of the mouse cursor in a desktop setting. The proposed encoding solution that was employed to compress the naturalistic content of the scene was evaluated, among several other aspects of quality (e.g., level of immersiveness and realism).

In (Mekuria et al., 2017b), performance results of the codec presented in (Mekuria et al., 2017a) are reported, from a quality assessment campaign that was conducted in the framework of the Call for Proposals issued by the MPEG committee (MPEG 3DG and Requirements, 2017). Both static and dynamic point cloud models were evaluated under several encoding categories, settings, and bit-rates. Animated image sequences of the models captured from predefined viewpoints were generated and assessed under passive inspection using a single-stimulus test method. The point clouds were rendered using cubes as primitive elements of fixed size across a model. This study aimed at providing a performance benchmark for a well-established encoding solution and evaluation framework.

Interactive variants of existing test methods are proposed in (Alexiou and Ebrahimi, 2017a,b) to assess the quality of geometry-only point clouds in a desktop setting. In both studies, Gaussian noise and Octree-pruning was employed to simulate position errors from sensor inaccuracies and compression artifacts, respectively, and to account for degradations of different nature. The models were simultaneously displayed as point sets side-by-side, while human subjects were able to interact without timing constraints before grading the visual quality of the models. This is the first attempt dedicated to evaluate the prediction power of metrics existing at the time. In (Alexiou et al., 2017), the same authors extended their efforts by proposing an AR

evaluation scenario using a head-mounted display. In the latter framework, the observers were able to interact with the virtual assets with 6DoF by physical movements in the real-world. A rigorous statistical analysis between the two experiments (Alexiou and Ebrahimi, 2017b; Alexiou et al., 2017) is reported in (Alexiou and Ebrahimi, 2018b), revealing different rating trends under the usage of different test equipment as a function of the degradation type under assessment. Moreover, influencing factors are identified and discussed.

A quality assessment study of position de-noising algorithms is performed in (Javaheri et al., 2017a). To this aim, impulse noise was initially added to the models in order to simulate outlier errors. After outlier removal, different levels of Gaussian noise were introduced to mimic sensor imprecisions. Then, two de-noising algorithms, namely Tikhonov and total variation regularization, were evaluated. For rendering purposes, the screened Poisson surface reconstruction (Kazhdan and Hoppe, 2013) was employed. The resulting mesh models were captured by different viewpoints from a virtual camera, forming video sequences. The reference and the degraded models were shown sequentially to human subjects in order to rate the perceived level of impairment of the latter.

The visual quality of colored point clouds under octree- and graph-based geometry encoding was evaluated in (Javaheri et al., 2017b), both by subjective and objective means. The color attributes of the models remained uncompressed to assess the impact of these geometry-only degradations; that is, sparser content representations are obtained from the first, while blocking artifacts are perceived from the latter. Static models representing inanimate objects and human figures were selected and assessed at three quality levels. Cubic geometric primitives of adaptive size based on local neighborhoods were employed for rendering purposes. A spiral camera path moving around a model (i.e., from a full view to a closer look) was defined to capture images from different perspectives. Animated sequences of the stimuli were generated and passively consumed by the subjects using the double-stimulus sequential test method. This is the first study with benchmarking results on more than one compression algorithms.

In (Alexiou et al., 2018) subjective experiments were conducted in five different test laboratories to assess the visual quality of colorless point clouds, enabling the screened Poisson surface reconstruction algorithm (Kazhdan and Hoppe, 2013) as a rendering methodology. The point cloud contents were degraded using Octree-pruning, and the observers visualized the mesh models side-by-side in a passive way. Although different 2D monitors were employed by the participated laboratories, the collected subjective scores were found to be strongly correlated. Moreover, statistical differences between the quality scores obtained from this experiment and the study conducted in (Alexiou and Ebrahimi, 2017b), indicated that different visual representations of the same point clouds might lead to different conclusions. In (Alexious et al., 2018), an identical experimental design is used, with human subjects consuming the reconstructed mesh models through various 3D display types/technologies (i.e., passive, active, and auto-stereoscopic). The results show very high correlation and very similar rating trends with respect to previous efforts (Alexiou et al., 2018), suggesting that human judgements on this data set and under the adopted test method are not significantly affected by the display

equipment.

The visual quality of voxelized colored point clouds was evaluated in (Torlig et al., 2018a) in subjective experiments that were performed in two intercontinental laboratories. Orthographic projections after real-time voxelization of both the reference and the distorted models were simultaneously shown to the subjects by means of an interactive renderer that was developed and described. Point clouds representing both inanimate objects and human figures were selected and compressed by the codec described in (Mekuria et al., 2017a), using combinations of geometric and color degradation levels. The results showed that subjects rate more severely distortions on human models. Moreover, using this codec, marginal gains are brought by color improvements at low geometric resolutions, indicating that the visual quality is rather limited at high sparsity. This is the first study reporting performance evaluation results of image-based quality metrics for point cloud contents.

In (Alexiou and Ebrahimi, 2019), identically degraded models as in (Torlig et al., 2018a) were assessed using a different rendering scheme. In particular, the point clouds were rendered using cubes as primitive geometric shapes of adaptive sizes based on local neighborhoods. Reference and impaired models were simultaneously displayed in an interactive platform for the evaluation of the latter, with the user's behavior being recorded. The rating trends found to be very similar to (Torlig et al., 2018a). The logged interactivity information was further analyzed and used to identify important perspectives of the models under assessment. The performance of image-based quality metrics was improved by proposed modifications in the relevant computational pipeline and a weighting scheme that exploits interactivity data was additionally proposed.

In (da Silva Cruz et al., 2019), the results of a subjective evaluation campaign that was issued in the framework of the JPEG Pleno (Ebrahimi et al., 2016) activities, are reported. Subjective experiments were conducted in three different laboratories in order to assess the visual quality of point clouds under an octree- and a projection-based encoding scheme. A passive evaluation protocol in conventional monitors was selected and different camera paths were defined to capture the models under assessment. The reference and impaired stimuli were rendered side-by-side using points of fixed size, which was specified per model and degradation level. This is reported to be the first study aiming at defining test conditions for both small- and large-scale point clouds. The former class corresponds to models that are normally consumed outer-wise, whereas the latter represent scenes which are typically consumed inner-wise. The results indicate that regular sparsity introduced by octree-based algorithms is preferred by human subjects with respect to missing structures that appeared in the encoded models from the projection-based counterpart, due to occluded regions.

Subjective evaluations of a volumetric video data set that was acquired and released was performed in (Zerman et al., 2019), under compression artifacts from the MPEG V-PCC. Two point cloud sequences sampled at four different resolutions were encoded under four quality levels of geometry and color distortions, leading to a total of 32 volumetric videos.

The stimuli were subjectively assessed in a passive way using two test methods; that is, a side-by-side evaluation of the distorted model and a pairwise comparison. The point clouds were rendered using primitive ellipsoidal elements of fixed size, determined heuristically to result in visualization of watertight models. The results showed that the visual quality was not significantly affected by geometric degradations, as long as the resolution of an encoded model allows adequate representation. Moreover, color impairments from V-PCC were found to be more annoying than geometric artifacts.

Subjective quality evaluation of different types of degradations, including Gaussian noise in both topology and texture, octree down-sampling and compression artifacts from the MPEG test models was conducted in (Su et al., 2019), using a wide set of colored models that were generated in the framework of the study. A passive inspection protocol with simultaneous visualization of the reference and distorted stimuli was employed for quality assessment. Point primitives of minimum size were employed for display purposes, with a virtual camera orbiting around each model at a fixed viewing distance to capture views. Among the outcomes, results showed that V-PCC outperforms the alternative codecs, especially at low bit-rates.

In (Alexiou et al., 2019a), a quality evaluation campaign was conducted in order to benchmark both subjectively and objectively the state-of-the-art MPEG test models, including V-PCC and all G-PCC variants. Several point clouds with diverse characteristics were employed and compressed following test conditions dictated by MPEG experts. The encoded versions were evaluated in an interactive platform with side-by-side visualization of the reference and the distorted models. The stimuli were displayed using splats of adaptive size based on local sparsity. As part of the study, subjective experiments under a pairwise comparison protocol were additionally performed, in order to conclude on preferable rate-allocation strategies for geometry, and geometry-plus-color encoding.

Static point clouds were evaluated in (Javaheri et al., 2019) subject to geometric compression artifacts under different rendering approaches. That is, geometry-only point primitives of fixed size with shading, geometry-plus-color point primitives of fixed size using the original color after a re-coloring step without shading, and geometry-only meshes after screened Poisson surface reconstruction (Kazhdan and Hoppe, 2013) with shading. For each rendering solution, a different evaluation session was established using the double-stimulus sequential test method. Regarding the selected encoders, the V-PCC and G-PCC using the TriSoup module were considered, together with the PCL octree-based codec. Results show that different scoring behaviors might be observed for the same compression impairments, as a function of the rendering approach. Moreover, the scoring deviations might vary per codec. Finally, it was suggested that texture information might mask underlying geometric distortions.

Visual quality assessment of dynamic point cloud contents visualized in a virtual reality scenario, both in 3DoF and 6DoF, is presented in (Subramanyam et al., 2020). Human figures from real-life acquisition and artificially generated were recruited, and encoded using the V-PCC and the anchor codec of the MPEG studies (Mekuria et al., 2017a). The models were

displayed in the virtual scene using quads of fixed size and were consumed by means of a head-mounted display. The users were allowed to navigate by physical movements in the 6DoF scenario, while remained sited for the purposes of the 3DoF counterpart. The subjects were required to rate the visual quality of each sequence using an absolute category rating protocol with a hidden reference. Results showed the superiority of V-PCC at low bit-rates, while statistical equivalence was found with the MPEG anchor at higher bit-rates as a function of the content. Finally, the inability of the codecs to achieve transparent visual quality was remarked.

Perceptual quality of static point clouds in VR was also evaluated in (Alexiou et al., 2020b). The users were able to interact with the stimuli in a 6DoF inspection scenario inside a virtual scene that was specifically designed to avoid distractions. The color encoding modules of the MPEG G-PCC test model were evaluated under octree-based geometry compression. For this purpose, two double-stimulus protocols with sequential inspection were adopted and compared. The models were displayed using quads of adaptive size that were interpolated before rendering to smooth the surfaces. The user behavior during evaluation was also analysed to provide further insights.

A study on the comparison of point cloud against mesh representations for compression of volumetric video is conducted in (Zerman et al., 2020). The Google Draco and JPEG encoding engines were employed for geometry and texture of mesh, respectively, while V-PCC and G-PCC were recruited to encode geometry and color of point cloud versions. As part of the study, the efficiency of the latter MPEG point cloud codecs was also analysed. All models were evaluated in a passive protocol using absolute category rating with hidden references from both content representations, while point clouds were displayed using fixed-size point primitives. Results show that the point cloud encoding-plus-rendering pipeline leads to better performance at low bit-rates, whereas higher quality levels are achieved by the mesh-based counterpart. However, the latter is attained for bit-rates that well-exceed the point cloud ones. Finally, among the MPEG alternatives, the superiority of the V-PCC was confirmed.

Similarly, a subjective evaluation of volumetric videos using both point cloud and mesh technologies is detailed in (Cao et al., 2020). Several additional factors were considered in the experimental design, among which the target bit-rate, the content resolution, and the viewing distance. To decrease the parameter space, for every target bit-rate, a manual identification of the optimal combination for model resolution and compression parameters per viewing distance was performed in a perceptual sense. The selected stimuli were evaluated following passive inspection protocols in two experiments that were carried out. In the first, the subjects rated the visual quality of models that were displayed using both types of content representations under a single-stimulus test method. In the second, a pairwise comparison between the same models represented as point clouds and meshes was issued. Based on the results, subjects favored the point cloud alternative at lower bit-rates. Moreover, the viewing distance was found to be an important factor, and mesh modelling was preferred at closer distances. At higher bit-rates and distant inspection, human opinions expressed equal

preference.

Subjective quality assessment of dynamic point clouds is conducted in (van der Hooft et al., 2020) in an adaptive streaming scenario hosted by the system described in (van der Hooft et al., 2019). For the purposes of the study, volumetric video sequences were selected and encoded at different quality levels using V-PCC. More than one models were placed in the same scene under different arrangements, and were visited with different navigation paths. The streamed cues were subjectively evaluated after passive consumption in a desktop setting. Among the experimental parameters, different bandwidth conditions, bit-rate allocation schemes, and prediction strategies were examined.

In (Perry et al., 2020), the performance of the MPEG codecs was assessed in terms of bit-rate against quality, using static colored point clouds. In this framework, the V-PCC and certain combinations of geometry and color encoding modules from G-PCC reference software were selected. The experiments were performed in four independent laboratories that participated in the relevant JPEG Exploration Study activities. A passive inspection protocol with a side-by-side visualization was employed, using fixed-size point primitives to display the models. The experimental set-up of each laboratory varied. Yet, the collected subjective scores exhibited high inter-laboratory correlation.

In (Yang et al., 2020), subjective quality assessment was issued on a large set of widely-employed colored models. Several degradation types affecting both the geometry and the color information were introduced, consisting of octree-pruning, noise injection in the coordinates and the RGB values, random down-sampling, and combinations of the above to further augment the visual impairments. The experiments were conducted using a single-stimulus, interactive evaluation protocol under a fixed inspection distance between the virtual camera and the model's origin. Among the main objectives of this study was to establish a large-scale subjectively annotated data set and to introduce a new objective quality metric.

In Table 2.1, a summary of existing subjective evaluation studies is attempted considering several experimental factors, in order to provide an informative synthesis of the current approaches.

2.4.2 Objective quality metrics

Objective quality metrics for point cloud contents can be distinguished as: (a) point-based, and (b) image-based approaches, which is very similar to the corresponding classification in mesh modelling (Lavoué and Mantiuk, 2015). The idea of converting point clouds to meshes prior to application of relevant algorithms was discarded quickly, as this additional processing step is commonly lossy.

Chapter 2. Related work

Table 2.1 – A categorization of subjective studies in terms of **model** [G: geometry, C: colored], **motion** [S: static, D: dynamic, B: both], **distortion** [GN: geometry noise, CN: color noise, DN: de-noise, D: down-sampling, O: octree pruning, C: compression, S: streaming], **rendering** [Min-P/T: minimum size points/triangles, Fix-P/Q/E/C: fixed size points/quads/s/ellipsoids/cubes, Adp-P/Q/C: adaptive size points/quads/cubes, Prj-V: projected voxels, Mesh: reconstructed mesh], **inspection** [I: interactive, P: passive], and **protocol** [DS-Seq/Sim: double stimulus sequential/simultaneous, SS: single stimulus, PC: paired comparison]. *Unk* stands for unspecified.

Paper	Model	Motion	Distortion	Rendering	Inspection	Protocol
(Zhang et al., 2014)	C	S	D & GN & CN	Min-P	<i>Unk</i>	<i>Unk</i>
(Mekuria et al., 2017a)	C	D	C	Min-P	I	SS
(Mekuria et al., 2017b)	C	B	C	Fix-C	P (zoom)	SS
(Alexiou and Ebrahimi, 2017a)	G	S	O & GN	Min-P	I	DS-Sim
(Javaheri et al., 2017a)	G	S	DN	Mesh	P	DS-Seq
(Alexiou and Ebrahimi, 2017b)	G	S	O & GN	Min-P	I	SS & DS-Sim
(Javaheri et al., 2017b)	C	S	C	Adp-C	P (zoom)	DS-Seq
(Alexiou et al., 2017)	G	S	O & GN	Min-T	I (AR)	DS-Sim
(Alexiou et al., 2018)	G	S	O	Mesh	P	DS-Sim
(Alexious et al., 2018)	G	S	O	Mesh	P	DS-Sim
(Torlig et al., 2018a)	C	S	C	Prj-V	I	DS-Sim
(da Silva Cruz et al., 2019)	C	S	C	Fix-P	P	DS-Sim
(Alexiou and Ebrahimi, 2019)	C	S	C	Adp-C	I	DS-Sim
(Zerman et al., 2019)	C	D	C	Fix-E	P	DS-Sim
(Su et al., 2019)	C	S	CN & GN & O & C	Min-P	P	DS-Sim
(Alexiou et al., 2019a)	C	S	C	Adp-P	I	DS-Sim & PC
(Javaheri et al., 2019)	G & C	S	C	Fix-P & Mesh	P	DS-Seq
(Subramanyam et al., 2020)	C	D	C	Fix-Q	I (VR)	SS
(Alexiou et al., 2020b)	C	S	C	Adp-Q	I (VR)	DS-Seq
(Zerman et al., 2020)	C	D	C	Fix-P & Mesh	P	SS
(van der Hooft et al., 2020)	C	D	C & S	Fix-P	P (zoom)	SS
(Cao et al., 2020)	C	D	C	Min-P & Mesh	P	SS & PC
(Perry et al., 2020)	C	S	C	Fix-P	P	DS-Sim
(Yang et al., 2020)	C	S	CN & GN & D & O	Min-P	I (no zoom)	SS

Point-based metrics

Current point-based predictors evaluate the level of impairment based on geometry and/or color properties of a point cloud model.

The majority of the proposed methods consist of full-reference approaches; this is, the presence of the original content is required for the computations. In full-reference metrics, a correspondence between the pristine and the impaired stimuli is essential. For this purpose, one model is selected as the reference and the other is set under evaluation. An association for each point of the latter model is then founded. Most commonly, for each queried sample of the model under evaluation, the nearest point or neighborhood that belongs to the reference is identified. Yet, there are metrics that follow different association algorithms, as it will be explained below.

After establishing correspondence, for every point of the model under evaluation, an individual error is obtained. A global degradation score for the entire model is computed via pooling across the individual values, with the most common choices being a simple average, the Mean Square Error (MSE), the Root-Mean-Square (RMS) error, and the Hausdorff distance. Note that by choosing one model as the reference, a specific correspondence and, in turn, a particular global degradation score is derived. Thus, for quality prediction that is independent of the reference selection, the so-called symmetric error is used. One way to export such a measurement is to set both the pristine and the impaired models as a reference, compute both global degradation scores and apply a function that grants symmetry, such as the max or the average pooling. Hereafter, the aforementioned procedure is implied when we refer to the symmetric error.

Early-developed predictors rely on simple distances between pairs of points that are associated under the nearest neighbor rule and assess geometry-only distortions. The point-to-point metric measures the Euclidean distance that separates the corresponding samples in the 3D space. Thus, an individual error value reflects the geometric displacement of a point from its reference position. The point-to-plane metric (Tian et al., 2017b) relies on the projected error of a queried point across the normal vector that corresponds to the reference sample. Hence, an error value indicates the deviation of a point from its linearly approximated reference surface. Using any of the above metrics, the MSE or the Hausdorff distance are more often used to obtain a global degradation score, while the symmetric error with max pooling is adopted to provide the final prediction.

The geometric Peak-Signal-to-Noise-Ratio (PSNR) measurement is proposed for the aforementioned metrics in (Tian et al., 2017a) to account for differently scaled contents. This is computed based on the ratio of a squared peak constant value, potentially multiplied by a scalar, divided by the symmetric squared error (e.g., MSE or squared Hausdorff distance). The peak can be set equal to the maximum nearest-neighbor distance of the original content. Alternatively, when the metrics are applied on voxelized models, the square root of the voxel grid diagonal, or the voxel grid resolution can be used instead. Note that, provided the symmetric error and the peak value, the PSNR is straightforwardly computed.

A pooling method referred to as the generalized Hausdorff distance is proposed in (Javaheri et al., 2020a), to improve the performance of the point-to-point and point-to-plane variants. The rationale is to exclude a percentage of the largest individual error values from the computation of the global degradation score, in order to mitigate the sensitivity of the Hausdorff distance to the presence of outlying points. Moreover, the same authors revise the computation of the geometric PSNR in (Javaheri et al., 2020b). In particular, the peak value in the numerator is replaced by estimators of the content's intrinsic resolution, which is computed as the maximum, or the average of nearest neighbors distances. A second formulation involves the regularization of the numerator using the previously defined intrinsic, or the so-called rendering resolution of the content. The latter is obtained by computing the average over projected distances of nearest neighbors onto the plane that is perpendicular to the normal

vector of each point.

The point-to-distribution geometry-based metric is introduced in (Javaheri et al., 2020c). In this case, the correspondence is realized between a point of the model under evaluation and a nearest neighborhood that belongs to the reference. An individual error for a queried point is computed using the Mahalanobis distance, in order to take under consideration properties of the local distribution of the reference samples. A degradation score for a model under evaluation is obtained by means of a simple average, with the symmetric error based on max pooling.

The plane-to-plane metric (Alexiou and Ebrahimi, 2018c) is based on the angular similarity of unoriented normals, or equivalently tangent planes, that correspond to pairs of nearest points that belong to the reference and the model under evaluation. Each individual error quantifies the difference in orientation between the linear local surface approximations of the corresponding models' shapes. A global degradation score is obtained using the average, or the MSE over individual angular similarity scores of the model under evaluation, and the symmetric error provides a final quality prediction.

In the same category of geometric predictors falls the PC-MSDM (Meynet et al., 2019). This is an extension of the well-known mesh-based MSDM metric (Lavoué et al., 2006; Lavoué, 2011) to point cloud contents. It relies on features that exploit local curvature statistics between pairs of associated points. The pairs are composed of reference samples and their projections onto surfaces that are fitted to the distorted model. The curvature values are computed after applying least-squares fitting of quadric surfaces in local neighborhoods defined around the associated points. A global degradation score is obtained using the Minkowski distance over individual error values, while a symmetric error is computed using average pooling. This metric was recently enhanced to include textural information by extracting additional color-based features. Moreover, a recommended weighting function was established in order to provide a global degradation score that considers both geometry and color distortions (Meynet et al., 2020). In the case of PC-MSDM, an asymmetric error is used.

In (Diniz et al., 2020b), local binary pattern (LBP) descriptors applied on the luminance channel are employed to estimate texture distortion. Voxelized point clouds are required as inputs, and the descriptors are computed on the k -nearest neighborhood of each point, thus, extending the 2D approach in the 3D space. Histograms of the extracted feature maps (i.e., labels) are obtained for both the reference and the distorted models. The histograms are then compared through a distance metric (i.e., Euclidean distance) before applying a regression algorithm to provide a quality score. This work was later extended in (Diniz et al., 2020a) to additionally take under consideration the point-to-plane distance between the point clouds, and the point-to-point distance between the corresponding feature maps in the computation of the prediction value.

In (Alexiou and Ebrahimi, 2020), statistical local features are proposed to compute a quality score, similarly to SSIM (Wang et al., 2004) in the image domain. The features are extracted

from local neighborhoods around each point, and are applied on quantities that are defined per attribute, considering location, color, normal and curvature information. A correspondence between samples from the model under evaluation and the reference is achieved using the geometric nearest neighbor. A global degradation score is obtained by pooling across an error map that reflects differences of the feature values between associated points, per attribute. Moreover, a voxelization step is proposed and optionally enabled prior to the feature extraction, which can lead to better predictions. Both asymmetric and symmetric errors using max pooling are exported.

A reduced-reference metric, namely PCM_RR, is described in (Viola and Cesar, 2020), relying on a diverse set of global features from location, color and normal information. In particular, 1D histograms are obtained from the coordinates of a model considering each axis, and the luminance values from the color attributes. Histograms reflecting shape uniformity, which is estimated based on angular similarity between normal vectors per local neighborhood, are also employed. Relevant distances are applied to compare the approximated distributions of the reference and the distorted models, in order to provide a quality score. Note that only the aforementioned statistics are required for the computation of this metric, thus, placing it to the reduced-reference class.

Simple point-based methods that assess the color of an impaired model make use of conventional formulas from 2D content representations. In particular, the formulas are applied on pairs of associated points. Similarly to the geometry-only case, the nearest neighbor rule is typically employed to achieve correspondence. A global degradation score is then estimated based on the MSE, or the corresponding PSNR, from individual error values that are obtained for the model under evaluation. The computations can be performed either in the RGB or the YCbCr color spaces. Finally, the symmetric error using max pooling is typically adopted to provide a prediction value.

In (Viola et al., 2020), metrics for color distortions that are based on histograms and an extension of correlograms to point cloud data, are introduced and benchmarked. Luminance-only and a weighted average including luminance and chroma components are evaluated to characterize the color distribution of a model. Several distances are also examined for comparison purposes. Finally, evaluation analysis of a weighted combination between the best-performing color-based predictor that is proposed and an existing geometry-based metric (i.e., point-to-plane) is reported.

Image-based metrics

In the image-based approaches, firstly used in (de Queiroz and Chou, 2017a) for point clouds, the rendered models are mapped onto planar surfaces, on which conventional 2D imaging metrics are applied to provide a quality score (Torlig et al., 2018a).

Requirements for the presence of the original model at run-time depend on the working

principle of the selected 2D imaging metric. So far, only full-reference approaches have been examined for quality evaluation of point clouds. In this case, views of the pristine and the impaired models are captured under identical camera parameters in order to compute an objective quality score. A global degradation is then estimated as an average, or a weighted average of the objective values that are derived under the adopted camera settings.

There are several factors that may influence the results of image-based metrics' computations. First and foremost is the rendering scheme that is employed to display point cloud data, together with the environmental and lighting conditions that are adjusted in the virtual scene. Note that the aforementioned specifications have a strong impact on the visual appearance of a model and can be set differently for users consumption and metrics execution. Moreover, the number of cameras and the configuration of each camera's parameters for the acquisition of model views, also affect the obtained scores. For this reason, image-based metrics are considered as rendering-dependent and view-dependent solutions (Lavoué et al., 2016; Alexiou et al., 2019a).

Nonetheless, image-based approaches are capable of simultaneously capturing both geometric and color degradations, as reflected in the selected renderer. In some cases, the realization of a simple rendering method might be part of the implementation of an objective metric, such as voxelization at a manually-defined voxel grid resolution as described in (Torlig et al., 2018b) and implemented by respective software². In principle, though, reproducing the rendering methodology and conditions that are set during consumption is preferred, since it allows to capture views of the content as experienced by users. For this purpose, snapshots of the models are commonly acquired from the corresponding application used for visualization.

Independently of the rendering scheme, the number of viewpoints and the parameters of the virtual camera can be set arbitrarily in order to capture the stimuli. Naturally, it is desirable to cover the maximum surface of a model, thereby incorporating as much visual information as possible in the extracted views. Yet, enabling a large number may lead to redundancies and extra computational costs, without guaranteeing performance improvements, as indicated in (Alexiou and Ebrahimi, 2019). Excluding pixels from the views that don't belong to the effective part of the displayed model (i.e., background information), was found to improve the accuracy of the predicted quality in (Alexiou and Ebrahimi, 2019). Moreover, the estimation of the global degradation score by incorporating importance weights based on the time of inspection of human subjects was proposed in the same study, and was found to increase the prediction performance.

Finally, in (Yang et al., 2020), an image-based metric is introduced based on a weighted combination of global and local features, which are extracted from texture and depth images that are captured after projecting the point cloud onto the 6 faces of a surrounding cube. The Jensen-Shannon (JS) divergence on the luminance channel serves as the global feature, whereas the local features consist of a depth edge map that reflects discontinuities, texture

²<https://github.com/digitalivp/ProjectedPSNR>

similarity that is applied on color components, and an estimated content complexity factor.

In Table 2.2, a categorized outline of current objective quality metrics is reported.

Table 2.2 – A categorization of objective quality metrics in terms of **class** [FR: full-reference, RR: reduced-reference], **domain** [P: point-based, I: image-based], **inputs** [L: location, N: normals, C: curvatures, RGB.: red-green-blue color, TI: texture image, DI: depth image], and **features**. The character “&” indicates AND, while “|” denotes CONDITION with optional inputs referred on the left side.

Paper	Class	Domain	Inputs	Features
(Tian et al., 2017b)	FR	P	L & N	Projected error
(Alexiou and Ebrahimi, 2018c)	FR	P	L & N	Angular similarity
(Torlig et al., 2018a)	FR	I	TI	Metric-dependent
(Alexiou and Ebrahimi, 2019)	FR	I	TI	Metric-dependent
(Meynet et al., 2019)	FR	P	L	Curvature local statistics
(Diniz et al., 2020b)	FR	P	L & RGB	LBP histograms
(Javaheri et al., 2020a)	FR	P	N L	Euclidean distance, or projected error
(Meynet et al., 2020)	FR	P	L & RGB	Curvature and luma-chroma-hue local statistics
(Viola et al., 2020)	FR	P	RGB	Luma histogram
(Alexiou and Ebrahimi, 2020)	FR	P	N, C, RGB L	Location, angular similarity, curvature, or luma local statistics
(Javaheri et al., 2020c)	FR	P	L	Mahalanobis distance
(Viola and Cesar, 2020)	RR	P	L & N & RGB	Location, angular similarity and luma histograms
(Diniz et al., 2020a)	FR	P	L & RGB	LBP histogram, distance, and projected error
(Javaheri et al., 2020b)	FR	P	N L	Euclidean distance, or projected error
(Yang et al., 2020)	FR	I	TI & DI	Depth-edge and texture similarity, and luma JS divergence

2.4.3 Discussion

Subjective quality assessment methodologies unveil the ground truth for quality characterization of impaired models. However, they denote expensive procedures in terms of time and computational costs that depend on the recruitment of human subjects to consume and evaluate the degradation level of the stimuli under evaluation. Objective quality metrics aim at providing accurate predictions for the visual quality of the impaired models after executing the corresponding algorithms. Yet, their validity needs to be verified through benchmarking against subjective opinions. It is noteworthy that the performance of a predictor might depend on the selection of contents and the types of degradation.

In the context of this thesis, we extensively experiment with both types of evaluation means for point cloud contents. In particular, we employ and extend well-established subjective quality

assessment methodologies. Moreover, we develop new point-based algorithms for quality prediction and experiment with the performance of image-based approaches. Finally, we recruit state-of-the-art objective quality metrics for benchmarking, and discuss their strong and weak points in order to draw conclusions regarding their performance and best-practices for their usage.

Contributions: Our contributions to the state-of-the-art in subjective quality evaluation studies are the following papers: (Alexiou and Ebrahimi, 2017a,b; Alexiou et al., 2017, 2018; Alexious et al., 2018; da Silva Cruz et al., 2019; Alexiou and Ebrahimi, 2019; Alexiou et al., 2019a, 2020b; Perry et al., 2020), which essentially form part I of this dissertation and chapter 9 of part III. Note that we haven't referred to relevant work in visual attention, which is the topic of our interest in (Alexiou et al., 2019b) and, in principle, can fall in the broader class of subjective experimentation. Pertinent studies are detailed in section 5, where we report our work. In the field of objective quality evaluation, the following papers have been published: (Alexiou and Ebrahimi, 2018a,c; Torlig et al., 2018a; Alexiou and Ebrahimi, 2019, 2020), which are detailed and evaluated in part II. Finally, publicly open software and data sets are provided to further facilitate research on the field (see annex E).

Measuring perceptual quality **Part I**

3 Quality evaluation of point clouds geometry

Visual quality of content representations is strongly linked to the user experience. Thus, the foundation of adequate methodologies for its quantification is crucial for human-centric applications. The perceptual quality of a content is evaluated through either objective, or subjective methods. In the first case, algorithms are designed to estimate the impact of signal degradations in terms of visual artifacts, and aim at providing accurate predictions of perceived quality. In the second case, evaluation experiments with the participation of human observers are conducted in order to rate its visual quality. Subjective quality assessments provide the ground truth since they depend on opinions of targeted end-users. Yet, they require an explicit design that allows repeatability and ensures reliability of the results. The Recommendations ITU-R BT.500-13 (ITU-R BT.500-13, 2012), ITU-T P.910 (ITU-T P.910, 2008) and ITU-T P.913 (ITU-T P.913, 2016) denote well-established and widely adopted manuals tailored for this purpose. In particular, they specify test methods, experimental designs and evaluation procedures, among other, for conventional 2D imaging modalities. However, it is unclear whether these standards should be employed as such, or the provided guidelines should be extended and adjusted to the richer features of 3D contents.

In this chapter we examine and define alternative methodologies for subjective quality assessment of point cloud contents. Our objective is to explore more realistic, yet, adequate paradigms that exploit the higher levels of interactivity that are offered by 3D visual data representations. To this aim, as a first step, we describe the stimuli that were collected and prepared to assemble the point cloud data set that will be used throughout our experimentation. Then, we focus on the evaluation of testing parameters, such as the test methods, display devices, and rendering schemes, after extending protocols from 2D to 3D imaging to integrate human interaction. In particular, experiments are performed to understand the impact of including explicit references in subjective quality assessment of point clouds. Moreover, an evaluation scenario is designed and conducted in AR, where participants inspect and interact with the queried stimuli through 6DoF by means of a head-mounted display (HMD). Correlation of subjective opinions from the previous experiments leads to useful insights regarding the effect of the display equipment in rating the same testing material. Finally, an experiment is per-

formed to evaluate the quality of point cloud contents after conversion to polygonal meshes. The latter is a common rendering approach that allows visualization of watertight models. We compare ratings from subjects inspecting the same stimuli under these two different visual data representations, namely point clouds and meshes, to draw conclusions regarding their statistical equivalence. In these efforts, we opt for geometry-only models to limit the parameter space of our experiments, since color information might be cited as distracting.

This chapter is based on material that has been published in (Alexiou and Ebrahimi, 2017a,b; Alexiou et al., 2017, 2018; Alexious et al., 2018; Alexiou and Ebrahimi, 2018b).

3.1 Data set preparation

3.1.1 Content selection

A total of 7 geometry-only models are selected to assemble our data set. *Cube*, and *sphere* are synthesized using corresponding mathematical formulas, while *torus* is artificially produced in MeshLab (Cignoni et al., 2008) to represent synthetic point clouds with perfect geometry. *Vase* is a model captured by Intel RealSense R200¹ and constitutes a representative content with irregular structure that can be acquired from a low-cost consumer market device. *Egyptian_mask*² is a model employed in relevant point cloud compression activities and the Call for Proposals issued by the MPEG standardization committee (MPEG 3DG and Requirements, 2017), also denoting a content with irregular topology. Finally, *bunny* and *dragon* are selected from the Stanford 3D Scanning Repository³ and represent point clouds with less irregular geometry and smooth underlying surfaces.

These models are selected to form a representative data set, considering the following properties: (a) Simplicity, as it would have been difficult for complex scenes to be distinguishable in the absence of color. Although simple, the complexity of contents covers a reasonable range. (b) Diversity of geometric structure, as different artifacts may be observed by applying different types of degradations. The selected models are generated by different means, resulting in different levels of geometric irregularity. (c) Similarity of point density⁴, as the number of points comprising a point cloud directly affects the faithfulness of the represented model.

3.1.2 Content preparation

The selected contents are scaled to fit in a minimum bounding cube of size 1 in order to normalize the impact of the applied distortions. To restrict the point density levels to a narrow range, a sparser version of the *dragon* is employed (i.e., namely, *dragon_vrip_res3*), while the initially captured *vase* and the originally released *egyptian_mask* models are downsampled

¹<https://intel.ly/2IID8FB>, last accessed 12/2020

²<http://mpegfs.int-evry.fr/MPEG/PCC/DataSets/pointCloud/CfP/>, last accessed 01/2020

³<http://graphics.stanford.edu/data/3Dscanrep/>, last accessed 12/2020

⁴Point density is defined as the number of points divided by the volume of the minimum bounding box.

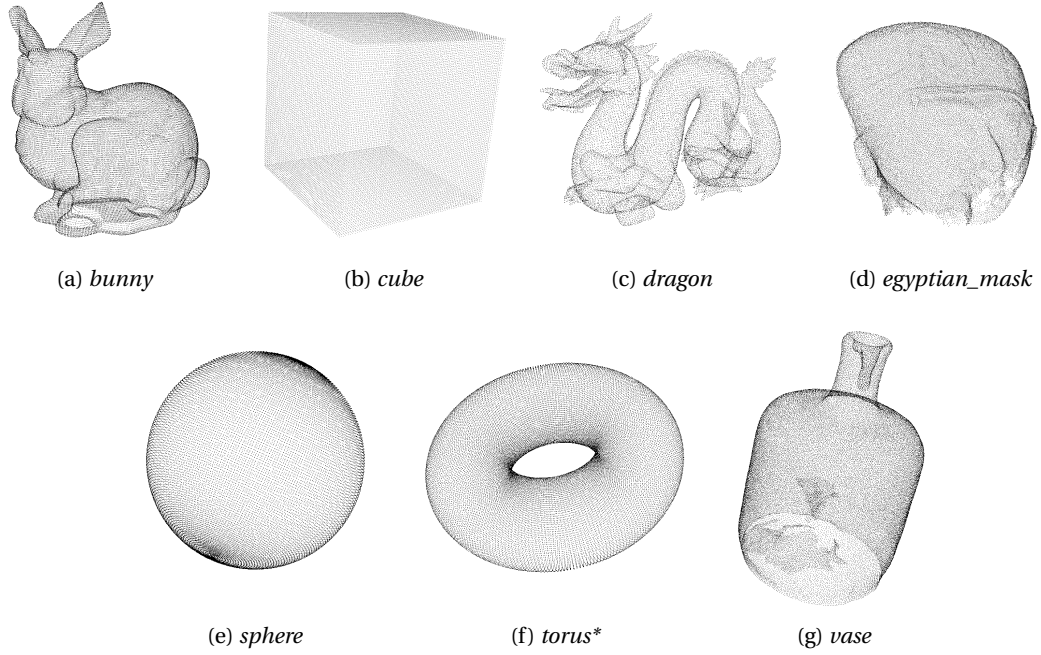


Figure 3.1 – Reference point cloud contents.

Table 3.1 – Geometric characterization of the reference point cloud contents.

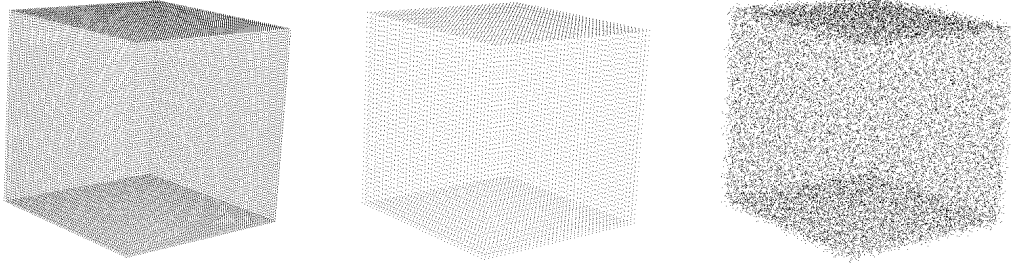
	<i>bunny</i>	<i>cube</i>	<i>dragon</i>	<i>egyptian_mask</i>	<i>sphere</i>	<i>torus</i>	<i>vase</i>
Points	35947	30246	22998	31601	30135	31250	36022
Min D	$3.79 \cdot 10^{-5}$	0.0101	$7.58 \cdot 10^{-4}$	0.0070	$1.48 \cdot 10^{-4}$	0.0042	0.0055
Max D	0.0144	0.0141	0.0113	0.0497	0.012825	0.0084	0.0104
X/Y/Z	1/0.99/0.78	1/1/1	1/0.71/0.45	0.99/1/0.82	1/1/1	1/0.33/1	0.68/1/0.68

using the CloudCompare software. The downsampling is performed by discarding points, such that no neighbors at the sparser version are closer than a specified threshold distance. This algorithm avoids displacements of the original coordinates, thus, maintaining the default structure of a content.

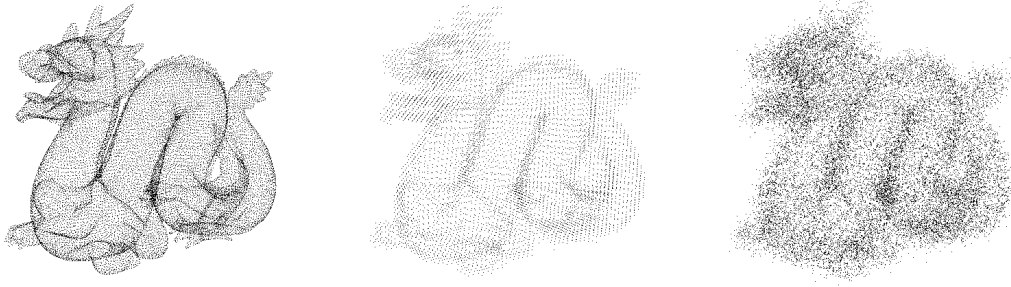
The prepared models are used as the reference contents in our experiments; in Figure 3.1 their snapshots are illustrated, while in Table 3.1 corresponding information regarding their geometric composition is reported.

3.1.3 Degradation types

In this study, two radically different types of geometric degradations are employed and assessed: (a) Gaussian noise, which results in points' displacements, and (b) Octree-pruning, which leads to point reduction and mapping to a regular grid. In both cases, the distortion



(a) *Cube*: (left) original, (middle) Octree-pruning with $p = 50\%$, (right) Gaussian noise with $\sigma = 0.008$



(b) *Dragon*: (left) original, (middle) Octree-pruning with $p = 50\%$, (right) Gaussian noise with $\sigma = 0.008$

Figure 3.2 – Illustrative examples of the visual artifacts that are introduced by the selected types of degradation.

levels are selected to cover a wide range of subjective scores, spanning from the lowest to the highest. In Figure 3.2, indicative examples of the types of artifacts that are generated under both types of degradation are presented.

Gaussian noise: It is widely used in the literature to simulate position errors due to depth sensor imperfections, or errors occurred after stereoscopic triangulation. In this case, we assume that the noise affects the coordinates of every point of a model, across each X, Y and Z axis. The injection of Gaussian noise, leads to the perception of scattered geometric arrangements, which are becoming more evident as the level of the noise is increasing.

For our experiments, this type of degradation is implemented using custom MATLAB scripts. The distortion level is determined by a target standard deviation, σ , that takes a value from the set $\{0.0005, 0.002, 0.008, 0.016\}$. Note that the coordinates of the point clouds are ranging over $[0, 1]$.

Octree-pruning: This type of impairment is obtained after octree decomposition at a selected level-of-detail (LoD), or a tree-depth. It leads to points' removal and displacement, giving rise to visible distortions in the form of structural loss. In principle, larger LoDs, or lower tree-

Table 3.2 – Configuration parameters for Octree-pruning. With *, we annotate contents that are employed only during the training.

Content	LoD values	Input points	Actual percentage	Target percentage
<i>bunny</i>	0.007	32957	8.32%	10%
	0.010	25209	29.87%	30%
	0.012	17763	50.59%	50%
	0.016	10870	69.76%	70%
<i>cube</i>	0.015	27541	8.94%	10%
	0.017	20888	30.94%	30%
	0.020	15002	50.40%	50%
	0.025	9602	68.25%	70%
<i>dragon</i>	0.008	20847	9.35%	10%
	0.010	16487	28.31%	30%
	0.013	11539	49.83%	50%
	0.017	7026	69.45%	70%
<i>egyptian_mask</i>	0.008	28393	10.15%	10%
	0.010	22061	30.19%	30%
	0.013	15790	50.03%	50%
	0.017	9466	70.04%	70%
<i>sphere</i>	0.004	27298	9.41%	10%
	0.011	21100	29.98%	30%
	0.015	15168	49.67%	50%
	0.020	8977	70.21%	70%
<i>torus*</i>	0.005	30566	2.19%	2%
	0.007	27968	10.50%	10%
	0.010	21901	29.92%	30%
	0.012	15715	49.71%	50%
	0.017	9539	69.47%	70%
<i>vase</i>	0.007	32454	9.90%	10%
	0.009	25217	30.00%	30%
	0.011	17963	50.13%	50%
	0.015	10693	70.31%	70%

depths are resulting in lower number of output points. See annex B.1 for further information.

For our experiments, Octree-pruning is implemented using the PCL software (Rusu and Cousins, 2011) version 1.8.0, selecting LoD values per content in order to discard a target percentage of the original number of points, p , that takes a value from the set $\{10\%, 30\%, 50\%, 70\%\}$. A deviation of $\pm 2\%$ is allowed for p . The actual percentages that are achieved are reported in Table 3.2, per stimulus. Note that for *torus*, which is a content that serves for training purposes, an additional version with $p = 2\%$ is prepared.

3.2 Evaluation methodologies

In this section we examine the influence of adopting different test methods in subjective quality assessment of point cloud contents. In particular, we conduct two separate subjective experiments following two of the most popular evaluation protocols, namely, the absolute category rating (ACR) and the degradation category rating (DCR), commonly and hereafter referred to as Double-Stimulus Impairment Scale (DSIS). The former is a test method that better simulates real-life consumption of visual data, whereas the latter is commonly employed to rate the fidelity of a distorted content with respect to its original version. In this study, we extend the aforementioned protocols by allowing the observers to interact with the queried stimuli, exploiting the 3D nature of the represented models. We employ the same rating scale, asking the participants to rate the level of impairment of the distorted stimulus with respect to their implicit, and a provided explicit reference, for the ACR and the DSIS experiment, respectively. To avoid additional influencing factors that might act as distractors such as texture or shading, geometry-only models are employed and displayed as raw point clouds. To fairly compare the selected test methods, the same desktop arrangement in a controlled testing environment is configured, and the same contents, degradation types, and degradation levels are evaluated by human observers. The objective is to address whether the subjects rate the testing material in the same way and, more generally, what is the impact of adopting different test methods in subjective quality assessment of point cloud representations.

3.2.1 Data set

In this study, the contents *bunny*, *cube*, *dragon*, *sphere*, and *vase* are selected. As detailed in section 3.1, the contents are pre-processed and scaled in a bounding cube of size 1, with their coordinates spanning in the range $[0,1]$. Both types of degradation are evaluated, namely, Gaussian noise and Octree-pruning, to account for substantially different artifacts, as shown in Figure 3.2. A target standard deviation, $\sigma = \{0.0005, 0.002, 0.008, 0.016\}$, is employed in the former case, whereas a target percentage of discarding points, $p = \{10\%, 30\%, 50\%, 70\%\}$, is defined in the latter case to account for different degradation levels.

3.2.2 Methodology

Test methods

Two widely known test methods are selected for this study, namely, (i) simultaneous DSIS, and (ii) ACR, using the same 5-rating impairment scale (5 - *imperceptible*, 4 - *perceptible, but not annoying*, 3 - *slightly annoying*, 2 - *annoying*, 1 - *very annoying*). The first method is commonly preferred for its high discriminative power and reliability, since subjects are able to visualize the degraded and the reference models side-by-side, facilitating the identification of differences. Moreover, the DSIS is renown for eliminating biases from personal preferences of observers over particular contents, as opposed to the ACR counterpart. However, the

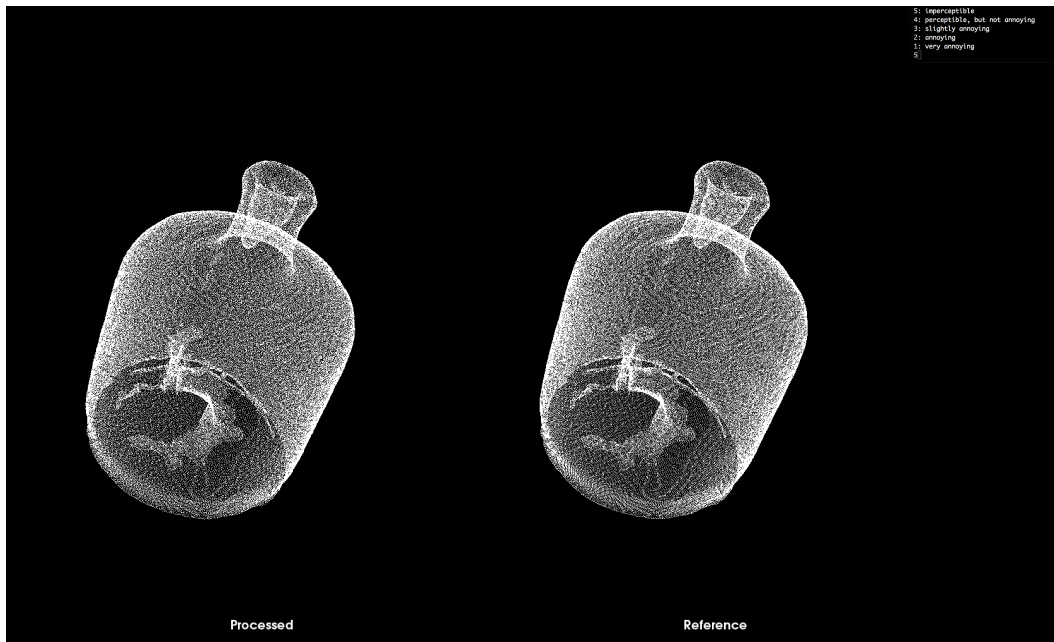


Figure 3.3 – Rendering application screen-shot showing the original *vase* on the right and the hidden reference on the left.

latter accounts for a more realistic type of media consumption. In the DSIS test method, the participants were instructed to rate the level of impairment of the distorted model with respect to an explicit reference that was provided and clearly annotated. In the ACR test method, the subjects were asked to rate the level of impairment of the distorted model with respect to their implicit/internal reference.

Both protocols are tailored to allow interactivity. In particular, through the renderer, the subjects were able to inspect, rotate, translate and zoom in/out to the point clouds in real-time using the mouse, and provide their scores using the keyboard. Despite the fact that passive evaluations enable identical viewing experience among users, it was considered important to let subjects get familiar with this type of visual data representation by allowing them to naturally interact, until they feel certain for their judgement. Hence, no time limitation was imposed during evaluations. Finally, a free viewing protocol was followed, meaning that the users were allowed to adjust their sitting position with respect to the screen.

Rendering

To render the stimuli the PCL visualizer is employed. The point clouds are displayed as collections of points, with each point represented by a single pixel onto the screen. The background color of the visualizer was set to black, while the color of the stimuli was set to white in order to increase the contrast and avoid distractions. A snapshot of the renderer is depicted in Figure 3.3.



Figure 3.4 – Participant inspecting the point clouds under assessment in the desktop set-up.

Testing environment

The experiments were conducted in the MMSPG laboratory, which fulfils the ITU-R Recommendation BT.500-13 (ITU-R BT.500-13, 2012) for subjective evaluation of visual data. A 30-inch Apple Cinema Display with a resolution of 2560x1600 was installed in the room. The luminance of the foreground and the background of the renderer was measured on the screen as 354 and 0.5 nits, respectively, using a luminosity sensor⁵. In Figure 3.4, a participant interacting with the stimuli under evaluation in our testing environment is presented.

Experimental design

Provided that the nature of artifacts that are introduced by Gaussian noise and Octree-pruning drastically differs, 4 separate sessions were held in total: (i) DSIS with Gaussian noise, (ii) DSIS with Octree-pruning, (iii) ACR with Gaussian noise, and (iv) ACR with Octree-pruning. Each session was launched after a training phase, where special care was given in order for the subjects to well-understand the impact of the corresponding type of impairment. Moreover, subjects were familiarized with the interaction part. During the testing, the same content was never displayed consecutively in order to avoid contextual effects. A different permutation of

⁵X-Rite i1 Display Pro - <http://www.xrite.com/>

the presentation order of stimuli was deployed per session and per subject, while for the DSIS test method, the side of the reference in the screen was selected randomly for every subject.

In each session, a total of 5 contents and 4 degradation values were used, along with a hidden reference for sanity check, leading to 25 stimuli per session. A total of 28 naive subjects (17 males and 11 females) participated in the experiments; 12 of them were involved in both while 16 participated in just one experiment, leading to 20 scores per stimulus. The age was ranging from 20.56 to 37.4 with an average of 28.18 and a median of 28.04 years of age.

Data processing

The MOS and CIs are computed to characterize the quality level and uncertainty of a particular stimulus, as described in annex A.1.1. Moreover, to compare the test methods, performance indexes described in annex A.2 are employed, which are issued per type of degradation. In particular, the ACR is compared against the DSIS test method after Gaussian noise and Octree-pruning, separately.

3.2.3 Results

Subjective results

Subject screening was applied on the collected scores of every experiment (i.e., test method) and each session (i.e., Gaussian noise and Octree-pruning). In the ACR experiment, no outlier was detected in none of the two sessions leading to 20 out of the 20 scores for both cases, while in the DSIS experiment one outlier was detected in the second session leading to 20 out of 20 and 19 out of 20 scores for Gaussian noise and Octree-pruning, respectively.

In Figures 3.5 and 3.6 the MOS and the corresponding CIs are indicated against the degradation levels for every type of impairment for the DSIS and ACR experiments, respectively. The markers with faces indicate the scores for the distorted point clouds, while the markers without faces (i.e., at the top-left corner) correspond to the ratings of the hidden references.

Results from both experiments indicate similar rating distributions with a general tendency of increasing MOS as the level of impairment is decreasing. It can be seen that, in the case of Gaussian noise, the subjective scores are decreasing following a logarithmic trend as the target standard deviation is increasing, for every content. However, this is not the case with Octree-pruning, where the shape of the model seems to influence the perceived distortions (e.g., *cube*).

Using the DSIS protocol in the presence of Gaussian noise, very similar scores are observed for every degradation level independently of the content, thus, suggesting that the subjects are able to recognize the amount of introduced noise. Using the ACR protocol, the *dragon* and *vase* are rated slightly but consistently lower than the other models, for every target standard

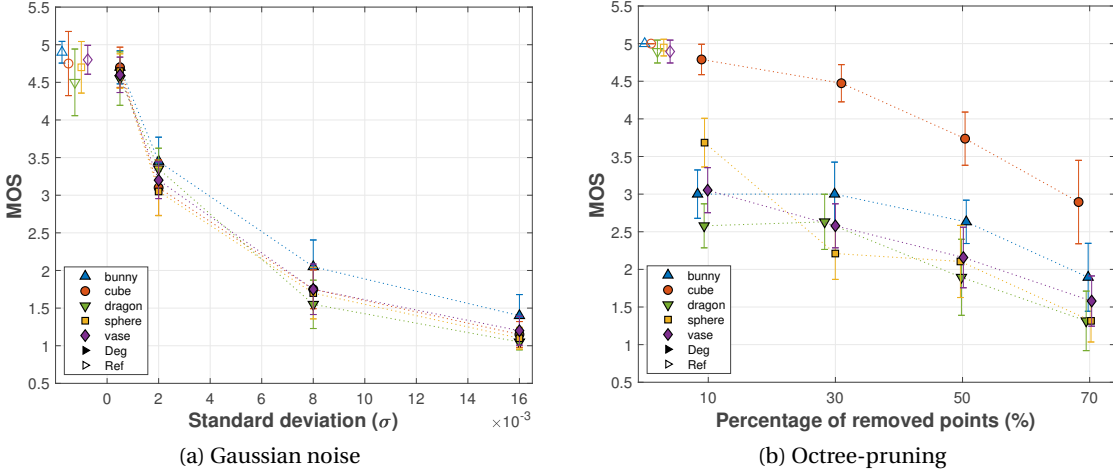


Figure 3.5 – Subjective scores against degradation levels using the DSIS test method.

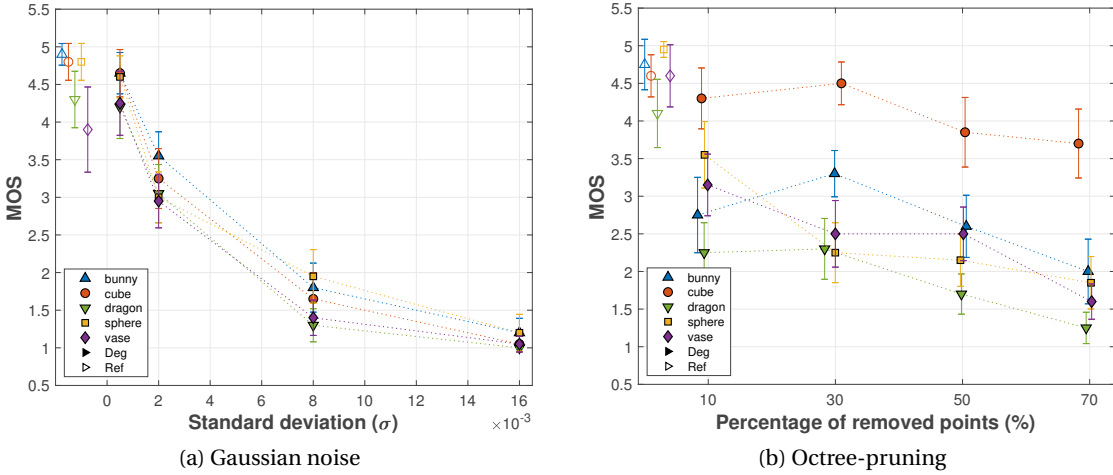


Figure 3.6 – Subjective scores against degradation levels using the ACR test method.

deviation. In fact, this tendency is even clearer in the ratings of the hidden references. For *dragon*, we assume that this behavior is observed due to the complexity of the object. In particular, although the density of is similar across contents, *dragon* is a substantially more complex object. This speculation could be partially verified by how the subjects rate the hidden reference in the Octree-pruning session, which is notably lower with respect to other hidden references, meaning that subjects tend to believe that the reference *dragon* is already simplified. Regarding *vase*, the difference in subjective scores is assumed to be observed due to its irregular geometric structure. There are two main reasons for this: (a) the training phase, which was conducted involving a content with more regular geometry; thus, when irregular topology was observed, participants tended to associate it with the existence of noise, and (b) a general preference of subjects towards regularly placed coordinates. Note though, that in the case of Octree-pruning, the reference version of *vase* is rated similarly to the other hidden

references; this is due to the absence of geometric regularity that was expected under this type of degradation, but not perceived by the subjects.

Using both DSIS and ACR protocols with Octree-pruning, we observe that *cube*, is rated remarkably higher than any other content, for every degradation level. The octree decomposition leads to elimination of high frequency components, which results to visible artifacts in the form of structural loss. Thus, more severe distortions are observed in point clouds with high curvature values and irregular structures, whereas the perception of the geometric arrangement of regular contents with planar underlying surfaces, such as *cube*, is not significantly impacted. Moreover, a steep increase of the MOS of the *sphere* can be observed for $p = 10\%$, for both ACR and DSIS experiments. This phenomenon can be explained by the non-uniform distribution of points; that is, the density of points in the poles is much higher and, thus, for $p = 10\%$, limited artifacts is perceived in the remaining surface. Finally, *bunny* is rated remarkably lower for $p = 10\%$ when compared to $p = 30\%$ in the ACR session. This is because, indeed, additional artifacts are perceived in the latter case. Yet, in the DSIS test method, this MOS decrease is not observed. This leads to the assumption that subjects tend to rate based on the number of points of the processed point cloud when the reference content is provided. The aforementioned observations suggest that simplifying objects without considering their underlying geometric properties may lead to enhanced visual distortions.

Comparison of test methods

In Table 3.3, the performance indexes of the MOS obtained from the DSIS (i.e., which is considered as ground truth) against the MOS from the ACR test methods are provided. No fitting, linear and cubic fitting are applied on the scores obtained from the ACR. Moreover, in Figure 3.7 the scatter plots comparing the subjective scores of the DSIS against the ACR are presented. The horizontal and vertical bars indicate CIs that are as computed by the scores of the test method indicated in the corresponding label. The linear and cubic fitting curves are also included.

In Table 3.4, the performance indexes of the MOS obtained from the ACR (i.e., ground truth) against the MOS from the DSIS are reported under all fitting functions, while in Figure 3.8, corresponding scatter plots are depicted.

A comparison of the average CIs shows that, in general, the uncertainty regarding the quality level of each stimulus is higher when using the ACR protocol, which can be explained by the absence of an explicit reference. In particular, the CIs were found to be 5.67% and 5.06% larger when using the ACR with respect to the DSIS in the presence of Gaussian noise and after Octree-pruning, respectively.

Based on our results, it is clear that the subjective scores from both test methods are strongly correlated when evaluating point clouds subject to Gaussian noise. The linear fitting function achieves an angle of 45° in Figure 3.7a and 44.5° in Figure 3.8a, indicating almost perfect

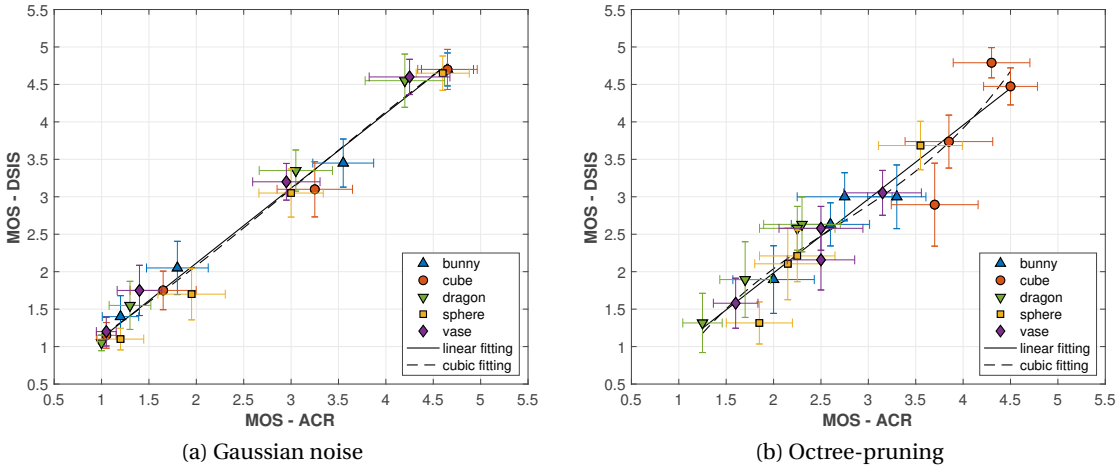


Figure 3.7 – Comparison of subjective scores obtained from both test methods (DSIS is set as the ground truth).

Table 3.3 – Performance indexes to compare the test methods (DSIS is set as the ground truth).

Gaussian noise											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.992	0.976	0.211	0.20	100%	0%	0%	95.79%	0%	1.58%	2.63%
Linear fitting	0.992	0.976	0.175	0.10	100%	0%	0%	95.79%	0%	1.58%	2.63%
Cubic fitting	0.992	0.976	0.174	0.05	100%	0%	0%	95.79%	0%	1.58%	2.63%
Octree-pruning											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.948	0.955	0.305	0.20	100%	0%	0%	84.74%	0%	8.95%	6.32%
Linear fitting	0.948	0.955	0.304	0.20	100%	0%	0%	84.74%	0%	7.89%	7.37%
Cubic fitting	0.952	0.955	0.291	0.20	100%	0%	0%	86.84%	0%	8.42%	4.74%

linear correlation. The intercepts of 1.11 and 0.96, though, in the former and the latter case respectively, indicate a slight tendency of higher ratings in the DSIS test method, consistently. The high PLCC and SROCC indexes, as well as the low RMSE and OR values, confirm the close correlation. Furthermore, a CE of 100% indicates no statistically significant difference between the MOS obtained from the two test methods. Finally, the FR of 0% and the marginal FD and FT percentages verify that the two protocols lead to almost identical conclusions for the assessment of two stimuli.

Results after Octree-pruning show that when scores from the DSIS experiment are considered as ground truth, an angle of 44.6° with an intercept of 1.24 are achieved, whereas when the ACR scores are set as the ground truth an angle of 42.3° with an intercept of 1.46 are observed. The linearity and monotonicity coefficients are high in both cases while the RMSE and OR values are still relatively low. Moreover, the CE is 100% and the FR is 0%. The high FD percentages

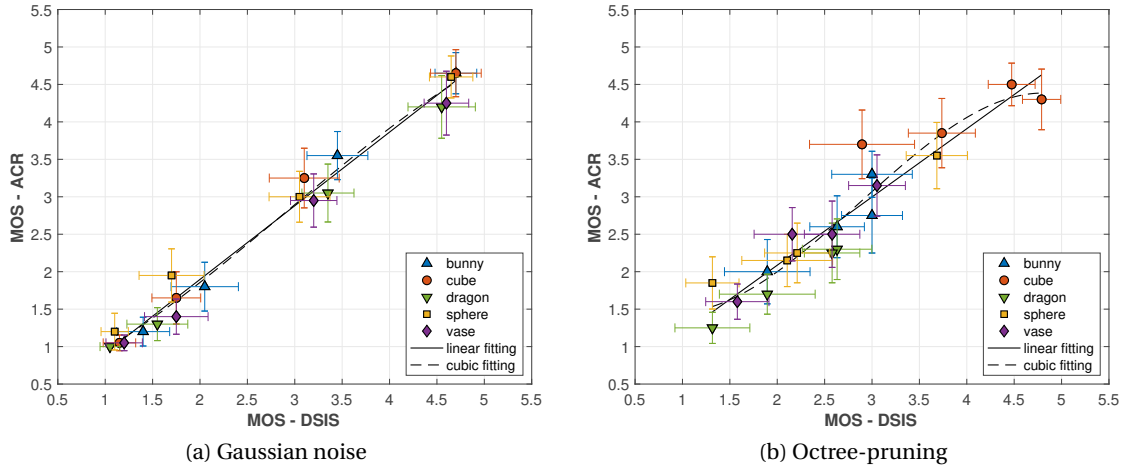


Figure 3.8 – Comparison of subjective scores obtained from both test methods (ACR is set as the ground truth).

Table 3.4 – Performance indexes to compare the test methods (ACR is set as the ground truth).

Gaussian noise											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.992	0.976	0.211	0.25	100%	0%	0%	95.79%	0%	2.63%	1.58%
Linear fitting	0.992	0.976	0.173	0.10	100%	0%	0%	95.79%	0%	2.63%	1.58%
Cubic fitting	0.992	0.976	0.171	0.05	100%	0%	0%	95.79%	0%	1.58%	2.63%
Octree-pruning											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.948	0.955	0.305	0.15	100%	0%	0%	84.74%	0%	6.32%	8.95%
Linear fitting	0.948	0.955	0.292	0.20	100%	0%	0%	83.68%	0%	4.74%	11.58%
Cubic fitting	0.954	0.955	0.275	0.10	100%	0%	0%	83.16%	0%	4.21%	12.63%

observed in Table 3.3 when setting the DSIS scores as ground truth, though, suggest that this test method does not differentiate two stimuli, while the ACR methodology decides that they are statistically different. This is confirmed through the high FT percentages of Table 3.4. These results indicate that stimuli with equivalent scores in the DSIS, are rated differently in the ACR experiment, and is mainly observed here for distorted versions of different contents. This can be explained by the presence of the reference model in the former case, thus, enabling relative comparisons per content.

3.3 Display devices

Richer content representations, such as point clouds, can be consumed in conventional monitors, smart-phones and HMDs. Depending on the device, different levels of interactivity are offered to the users, which may affect the perception and, by extension, the perceived

quality of the displayed visual data. In this study, we explore the prospect of conducting subjective quality assessment of point cloud representations in AR. The subjects are able to inspect and interact with the models that are placed as virtual assets in the real world through physical movements and can be consumed by means of an HMD. We then compare the scores collected in this experimental set-up to human opinions obtained from a desktop arrangement, to identify potentially deviating rating trends. To limit the parameter space of our experiment while exploiting previous efforts, colorless point clouds are assessed without enabling any illumination or shading during rendering. To ensure fair comparison, identical contents, degradation types and levels were chosen for both experiments. The objective of this study is to, first, define a subjective evaluation methodology that exploits the full potentials of advanced content representations and, second, to examine whether utterly different display devices that create different user experiences lead to the same conclusions for the visual quality of point clouds, even when only the geometry of the models is evaluated.

3.3.1 Data set

In this study, the contents *bunny*, *cube*, *dragon*, *sphere*, *torus* and *vase* were selected for the purposes of our experiment. *Torus* was used in the training stage, thus, it was excluded from the test. As detailed in section 3.1, the contents are pre-processed and scaled in a bounding cube of size 1, with their coordinates lying in $[0,1]$. Both types of degradation are evaluated, namely, Gaussian noise and Octree-pruning. In the first case, a target standard deviation, $\sigma = \{0.0005, 0.002, 0.008, 0.016\}$, is employed, whereas in the second case a target percentage of discarding points, $p = \{10\%, 30\%, 50\%, 70\%\}$, is defined to simulate different degradation levels.

3.3.2 Methodology

Test method

The simultaneous DSIS protocol, which was found to be consistent for rating the level of impairment of point clouds according to the results of section 3.2.3, is adopted in the HMD AR experiment. Both the reference and the distorted models are visualized side-by-side, under the same 5-rating impairment scale (5 - *imperceptible*, 4 - *perceptible, but not annoying*, 3 - *slightly annoying*, 2 - *annoying*, 1 - *very annoying*), with the subjects instructed to rate the level of impairment of the distorted model with respect to the provided reference.

In both experiments, the subjects interacted with the stimuli under evaluation in a free viewing fashion, while exploiting the DoF that were offered by the corresponding display device. For the desktop experiment, this means that the subjects were able to change their initial distance between their position and the screen, while inspecting the stimuli from the selected viewpoint specified by mouse movements. For the HMD AR experiment, the subjects were free to interact with the queried stimuli by changing their physical position with 6DoF interactions in the real

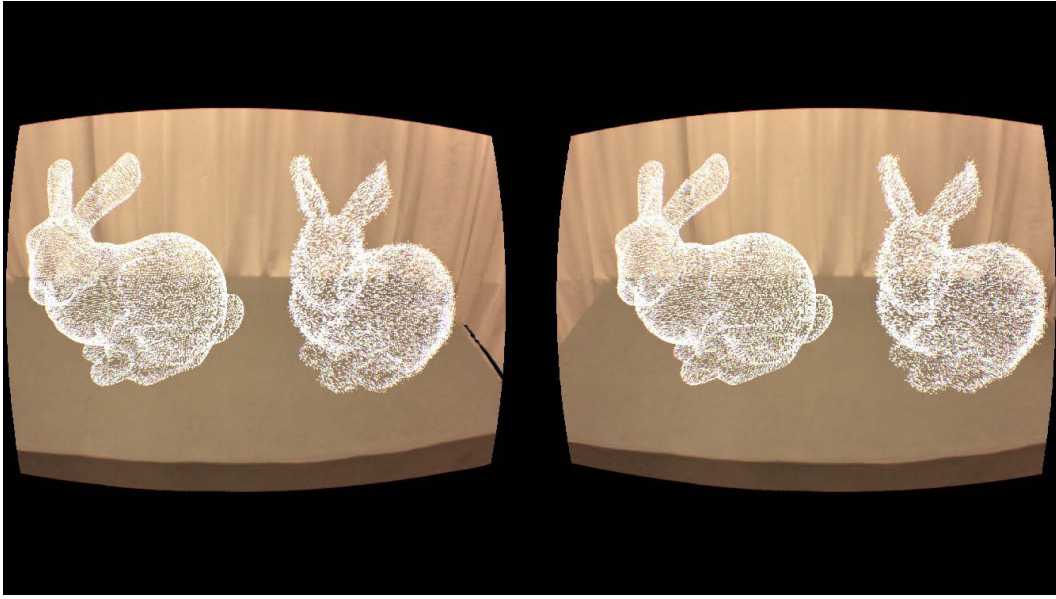


Figure 3.9 – Rendering application screen shot showing the reference *bunny* on the left and its impaired version with Gaussian noise of $\sigma = 0.008$ on the right.

world (e.g., coming closer, changing their point of view, etc.). No restrictions were introduced in terms of time duration, or viewpoint selection for the evaluation of the stimuli. At the end of each individual evaluation, the subjects were informing the operator and were providing a score orally after listening to the rating scale that would select from. The order of the rating scale was provided identically after each pair of stimuli and for every subject.

Rendering

As described in section 3.2.2, the PCL visualizer was employed in the desktop set-up. The background color was set to black, and the foreground was set to white.

In the HMD AR set-up, as in the desktop arrangement, the models were displayed as collections of points. In this case, though, each point was represented by an atomic triangle of minimum size. Provided that these atomic triangles were significantly smaller than the dimensions of a point cloud model, they were perceived as points by an observer. The virtual assets were added on top of the real scenery, defining an AR scenario. Thus, the background of the renderer was the real-world environment with colors involving different shades of grey, while the color of the points was set to white. The models under assessment were placed in a fixed location on top of a test table, which was covered by a medium grey tissue. In Figure 3.9, a snapshot of the renderer is provided showing a pair of virtual models in the real-world scene.



Figure 3.10 – Participant inspecting the point clouds under assessment in the HMD AR set-up.

Testing environment

The desktop experiment is detailed in section 3.2.2, where the reader may refer for more details. In brief, it was conducted in a controlled environment in the MMSPG laboratory with a desktop set-up using an Apple Cinema Display.

The HMD experiment was also conducted in the MMSPG laboratory. The subjects were observing the stimuli by means of the Occipital Bridge AR headset⁶, using an iPhone 6S as the screen with a resolution of 326 pixels per inch. The Occipital Bridge software development kit libraries allow rendering of a real world scene captured by the phone's camera with an attached wide angle lens of 120 degree field of view. The luminance values of the points and the test table surface were measured on the screen as 595.28 and 38.91 nits, respectively. In Figure 3.10, a participant interacting with the virtual testing models is illustrated.

Experimental design

Similarly to the evaluation procedure followed in the desktop experiment, the HMD AR experiment was split in two sessions; that is, one session was held for the assessment of stimuli

⁶<https://bridge.occipital.com/>

subject to Gaussian noise and a separate session was conducted for Octree-pruning distortions. Each session was launched after a training phase, where the subjects were informed about the general characteristics of the type of noise they would assess, and got familiarized with the HMD set-up. During the testing, the subjects were asked to stand in front of the test table that was installed in the scene at the distance of 1 meter at the beginning of each evaluation, and then, they were free to interact with the queried stimuli at will. For the presentation of the stimuli, particular care was given in order to randomize the order of the observed pairs per session and subject. Moreover, the position of the reference was picked randomly per subject and remained fixed across a session.

In each session of the HMD AR experiment, 5 contents and 4 degradation values were used along with a hidden reference resulting in 25 stimuli per session. A total of 24 naive subjects (14 males and 10 females) participated in the experiments; 18 of them were involved in both sessions while 6 participated in just one, leading to 21 scores per stimulus. The age was ranging from 25 to 32 with an average of 27.66 and a median of 28 years of age.

Data processing

The MOS and CIs are computed to quantify the perceived quality of a particular stimulus. To compare the two visualization methods, the performance indexes described in annex A.2 are applied on the derived scores per type of degradation.

3.3.3 Results

Subjective results

The analysis of the subjective scores from the desktop set-up is reported in section 3.2.3. In this section, we present the results from the HMD AR experiment, following the same procedure to allow comparisons. In particular, subject screening according to the ITU-R Recommendation BT.500-13 (ITU-R BT.500-13, 2012) was applied on the scores collected from each session (i.e., Gaussian noise and Octree-pruning), revealing 3 and 1 outliers for Gaussian noise (i.e., 18/21) and Octree-pruning (i.e., 20/21), respectively.

In Figure 3.11 the MOS along with the CIs against the degradation values are depicted, with markers without faces (i.e., at the top-left corner of the Figures 3.11a and 3.11b) indicating the scores of the hidden references. It can be observed that as the standard deviation of the Gaussian noise is increasing, the MOS is decreasing. The subjects seem to be able to recognize easily the amount of noise introduced, independently of the displayed content. The particular test method that is adopted (i.e., simultaneous DSIS), also, assists to obtain such results, since the subjects are always aware of the reference content and can base their judgements on relative geometric differences.

Conversely, when the contents are subject to Octree-pruning distortions, the underlying

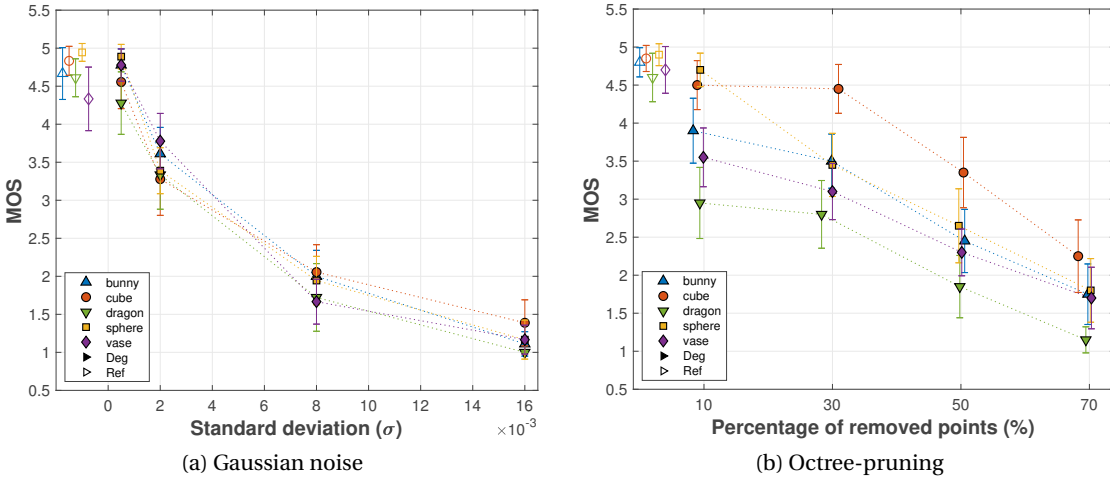


Figure 3.11 – Subjective scores against degradation levels in the HMD AR experiment.

surface and the shape of the content seems to play a significant role. As the number of points is decreasing, less details and more rough representations of curved edges are observed, which leads subjects to penalize distorted stimuli of higher curvature with lower scores. For instance, *cube* is rated remarkably higher than any other content (except for *sphere* at $p = 10\%$). The structural loss occurred after octree decomposition is not perceived as truly annoying for this particular content, because it does not affect its geometric structure. Moreover, the models *sphere*, *bunny* and *vase* are similarly rated, whilst *dragon*, which is the most complex, is notably under-rated. Any removal of points for this particular object has higher impact, and even for $p = 10\%$ the MOS is much lower than the MOS of the hidden reference. Another reason for *dragon*'s scores is its geometry. The shape of this content and, specifically, the ratio between height and length is such that it looks remarkably smaller than the other models, despite the fact that all point clouds were scaled to fit in the same bounding cube. Since the subjects mostly kept a fixed distance during evaluation, perceiving one object as smaller than the others may have affected their ratings. *Vase*'s irregular structure is transformed to a regular representation after Octree-pruning. Provided that subjects tend to rate based on relative differences, the MOS of the *vase* is systematically lower than the MOS of *bunny* and *sphere*, whose geometry is more regular. Finally, as *sphere* is artificially generated, the density of points in poles is much higher. For $p = 10\%$, no remarkable impairments occur in the remaining surface and, thus, it is rated similarly to the hidden reference.

Based on observations extracted during the experimental procedure, most of the subjects prefer to stay static. Although the performance of the device is sufficient, and just a few subjects experienced motion sickness, there were very few cases of users that were feeling confident with the interactivity part. The level of interaction and the viewing position are important factors, and in order to compensate their impact on the MOS and CIs, we would suggest to use more than 15 subjects, which is a number proposed for quality assessment of conventional content. Furthermore, subjects tend to rate models based on the number of

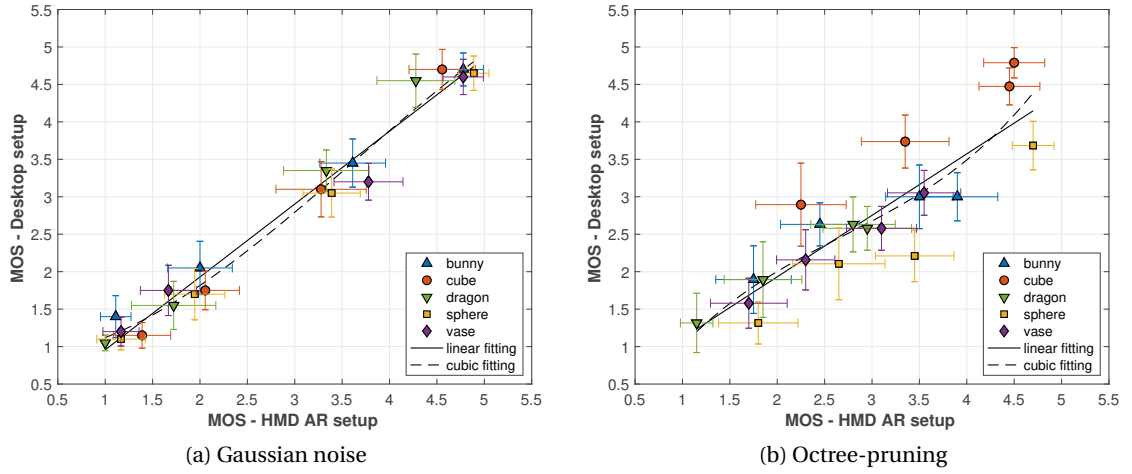


Figure 3.12 – Comparison of subjective scores obtained under both display devices (Desktop scores are set as the ground truth).

Table 3.5 – Performance indexes to compare display devices (Desktop scores are set as the ground truth).

Gaussian noise											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.988	0.959	0.234	0.30	100%	0%	0%	94.21%	0%	3.68%	2.11%
Linear fitting	0.988	0.959	0.210	0.25	100%	0%	0%	94.21%	0%	3.68%	2.11%
Cubic fitting	0.990	0.959	0.192	0.15	100%	0%	0%	96.32%	0%	0.53%	3.16%
Octree-pruning											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.879	0.872	0.544	0.60	90%	0%	10%	73.68%	0%	17.89%	8.42%
Linear fitting	0.879	0.872	0.454	0.45	95%	0%	5%	72.11%	0%	15.26%	12.63%
Cubic fitting	0.885	0.872	0.443	0.45	100%	0%	0%	76.32%	0%	14.21%	9.47%

points and, in general, they prefer regular representations. For instance, in the case of *vase* for $p = 10\%$, a few subjects asked why there is no option to rate the processed content higher than the reference.

These results are quite similar to the conclusions obtained in section 3.2.3. In order to quantify potential rating differences between the two experiments, namely, desktop and HMD AR, we statistically analyse the corresponding subjective scores.

Comparison of display devices

In Table 3.5, the performance indexes of the MOS obtained from the desktop set-up (i.e., which is considered as ground truth) against the MOS from the HMD AR set-up are reported after applying no fitting, linear and cubic fitting on the latter, while in Figure 3.12 scatter

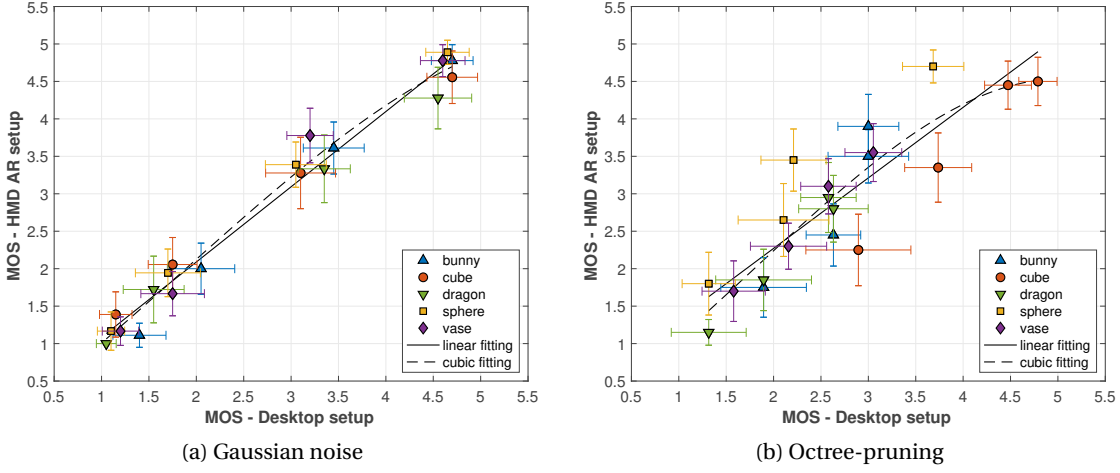


Figure 3.13 – Comparison of subjective scores obtained under both display devices (HMD AR scores are set as the ground truth).

Table 3.6 – Performance indexes to compare display devices (HMD AR scores are set as the ground truth).

Gaussian noise											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.988	0.959	0.234	0.25	100%	0%	0%	94.21%	0%	2.11%	3.68%
Linear fitting	0.988	0.959	0.214	0.15	100%	0%	0%	94.21%	0%	2.11%	3.68%
Cubic fitting	0.990	0.959	0.195	0.15	100%	0%	0%	94.21%	0%	2.63%	3.16%
Octree-pruning											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.879	0.872	0.544	0.45	90%	10%	0%	73.68%	0%	8.42%	17.89%
Linear fitting	0.879	0.872	0.486	0.45	95%	5%	0%	73.68%	0%	8.42%	17.89%
Cubic fitting	0.891	0.872	0.463	0.35	100%	0%	0%	74.74%	0%	6.84%	18.42%

plots comparing the subjective scores from the two set-ups are illustrated, for both types of degradation. Similarly, in Table 3.6 and Figure 3.13, the performance indexes and scatter plots of the MOS collected from both tests are depicted, considering the HMD AR set-up as ground truth.

Based on our results, it is evident that for both types of degradation, the CIs associated with scores obtained from the HMD AR experiment are larger. In particular, for Gaussian noise and Octree-pruning, the CIs from the HMD AR are on average 11.54% and 7.31% bigger when compared to the desktop set-up, respectively. The latter values are obtained by averaging over all possible combinations of valid subjects, when the cardinality of one set of scores is higher than the other's. For Gaussian noise, the standard error is 0.0168 for desktop and 0.0261 for HMD AR, respectively, while for Octree-pruning, the standard error is 0.0202 for the desktop and 0.0184 for the HMD AR set-up. This behavior is partially due to the higher level of interac-

tivity that is offered by an HMD device, leading to in principle different viewing experiences among subjects and across stimuli. Moreover, the real environment scenery as a background is an additional factor that could have influenced the perception of the virtual objects. Despite the fact that no issues were reported by the subjects, the level of discomfort when wearing an HMD is admittedly higher and could lead to rating inconsistencies throughout a session. To average such statistical uncertainties, a larger number of participants is proposed.

Concerning the results in the presence of Gaussian noise, in Figures 3.12a and 3.13a, the linear fitting function achieves an angle of 44.2° and 45.1° , with an intercept of 0.96 and 1.14, respectively. This indicates that although highly correlated, the scores obtained in a desktop arrangement are consistently slightly lower. The strong correlation is verified by the high PLCC and SROCC values of Tables 3.5 and 3.6. A CE percentage of 100% implies that the MOS of the distorted contents, as rated in both testbeds, are statistically equivalent. The FR is 0%, while the FD and FT percentages are also rather low (below 3.7%).

Regarding the results after Octree-pruning, in Figures 3.12b and 3.13b the linear fitting function achieves an angle of 39.4° and 43.3° with an intercept of 1.23 and 1.63, respectively, without revealing any particular trend. Based on the performance indexes of Tables 3.5 and 3.6, the selection of a different testbed may lead to different conclusions regarding the visual quality of Octree-pruning artifacts. In particular, the PLCC and SROCC values are lower, while the RMSE and OR coefficients are remarkably higher with respect to the Gaussian noise case. A CE below 100% indicates that there is a percentage of distorted models for which the MOS values are statistically distinguishable, with the subjects over-estimating the visual quality using the HMD AR set-up. In fact, *sphere* is clearly under-rated in the desktop set-up, which is assumed to be a result of the structured-type of loss that is perceived when this content is displayed in a flat monitor. In the HMD AR experiment, these artifacts are not visible, potentially, due to the lower phone's resolution and the short eye-to-screen distance. A substantial difference is observed at a degradation level of $p = 10\%$ for this content. As mentioned in section 3.2.3, in this case the main distortions exhibit on the *sphere*'s poles. Thus, the users couldn't easily detect them in the HMD AR setting, as it would have been practically difficult to assess the contents from the top or the bottom, and maintain the same distance as if they were standing in front of the content. Finally, despite the fact that the FR index remains at 0%, high percentages of FD in Table 3.5 and FT in Table 3.6 suggest that models that are not differentiated in the desktop set-up, may be rated differently in the HMD AR test. The poor performance can be verified by the large spread of the distribution of data points in the scatter plots 3.12b and 3.13b.

3.4 Rendering schemes

In many applications, point cloud data are not meant to be displayed directly, rather processed to extract information about the imaged scenery; an indicative example is the usage of LiDAR technologies in autonomous driving. However, in other application scenarios, such as in entertainment industry, humans are targeted as the end-users to consume the point cloud

contents. In the latter case, it is rather common to apply surface reconstruction for rendering purposes. This is, algorithms that treat point clouds as sets of discrete samples and aim at inferring continuous underlying surfaces. The reconstructed models are represented as polygonal meshes, which have been extensively exploited for 3D modelling in computer graphics. In this study, we investigate such an alternative visualization approach in the context of subjective quality assessment of point cloud geometry. In particular, we employ a popular surface reconstruction algorithm and convert point cloud to mesh representations before rendering. Then, we ask from human observers to rate the visual quality of the latter. The tests are conducted in five independent research laboratories employing different display equipment in desktop arrangements. Initially, an analysis of the correlation between the human opinions collected from all participated laboratories is issued. Then, we compare the scores derived from this experiment with subjective ratings obtained from a prior experiment using the same stimuli, but this time rendered as collections of points. The main objective of this study is to identify whether similar conclusions regarding the visual quality of point cloud data can be drawn when using these two different rendering schemes.

3.4.1 Data set

In this study, the contents *bunny*, *cube*, *dragon*, *sphere*, *torus* and *vase*, are selected and evaluated in our experiments. *Torus* is used for training purposes, thus, it was excluded from the test. The contents are pre-processed as described in section 3.1, and translated at the origin (0, 0, 0). The Octree-pruning is only employed in this experiment to degrade the models, as it is considered more relevant, using the same target percentage of point removal, $p = \{10\%, 30\%, 50\%, 70\%\}$.

3.4.2 Methodology

Test method

The simultaneous DSIS test method is adopted with a 5-level impairment scale (5 - *imperceptible*, 4 - *perceptible, but not annoying*, 3 - *slightly annoying*, 2 - *annoying*, 1 - *very annoying*), including a hidden reference for sanity check. Thus, both the reference and the degraded stimuli were simultaneously shown to the observer, side-by-side, and every subject rated the visual quality of the processed with respect to the reference stimulus.

The subjects consumed the testing material passively by inspecting generated 2D video sequences that show different views of the models. In every experiment, a free viewing (FV) scenario was adopted; that is, after the initial position, every subject was free to move closer or further from the screen during the evaluation. This is because different objects could be perceived of different volume. For instance, from a fixed distance between the observer and the screen, the *dragon* is perceived smaller with respect to the *sphere*, due to the different ratio between height and length. The initial viewing distance that was adopted together with the

Table 3.7 – Equipment details per laboratory.

	EPFL	UBI	UC	UNIN	UP
Monitor	Apple Cinema	ASUS	Sony	Sony	Dimenco
Model	M9179LL/A	PB287Q	KD-49X8005C	KD55x8505	DM504MA5
Inches	30"	28"	49"	55"	50"
Resolution	2560x1600	3840x2160	3840x2160	3840x2160	1920x1080
View distance	0.7 m (FV)	1.5 m (FV)	1.8 m (FV \pm 30 cm)	1.5 m (FV)	1.5 m (FV)

limitation of movements that was optionally imposed in every laboratory can be found in Table 3.7. The video sequences were consumed through the MPV⁷ video player, for which a custom interface was developed to allow subjects to provide their scores, either during or after the completion of the animation.

Rendering

For rendering purposes, the Screened Poisson surface reconstruction algorithm (Kazhdan and Hoppe, 2013) is selected. The CloudCompare implementation is employed with a tree-depth of 8 and default parameters. This method is popular due to: (i) high performance, (ii) availability in open source software, (iii) guaranteed generation of watertight objects, (iv) adjustable complexity, as a function of the tree-depth, and (v) reproducibility of the generated meshes. Yet, it relies on the presence of normal vectors. In the absence of this type of attributes from the point cloud data, normal vectors need to be estimated. To this aim, the CloudCompare software was used with default settings; that is, the radius to identify a nearest neighborhood was selected automatically per stimulus, and a plane fitting was employed. Then, on the same tool, the normal vectors were oriented using a Minimum Spanning Tree of 6 nearest neighbors. After visual inspection, the orientation of a model's vectors was flipped, if needed. In Figure 3.14, the reference models after conversion to mesh representations are illustrated. In Figure 3.15, each reconstructed version of the *bunny* model is depicted, in order to provide visual examples of the distortions that occur in mesh representations from Octree-pruning.

To render the meshes, the default VTK⁸ visualizer, as integrated in PCL was employed. The default lighting conditions were set to the scene and flat shading was applied in order to avoid masking visible artifacts. A white color was used for the foreground and black color for the background in order to enhance contrast and reduce distractions. To capture projected views, the models were placed at the origin of the virtual environment and a fixed distance from the camera was set to avoid changes of the model's size that may be perceived as the virtual camera is circularly moving around it. The camera rotated around the horizontal first and, then, around the vertical axis of the center of the object in steps of 1°. In every step, a still

⁷<https://mpv.io/>

⁸<https://vtk.org/>

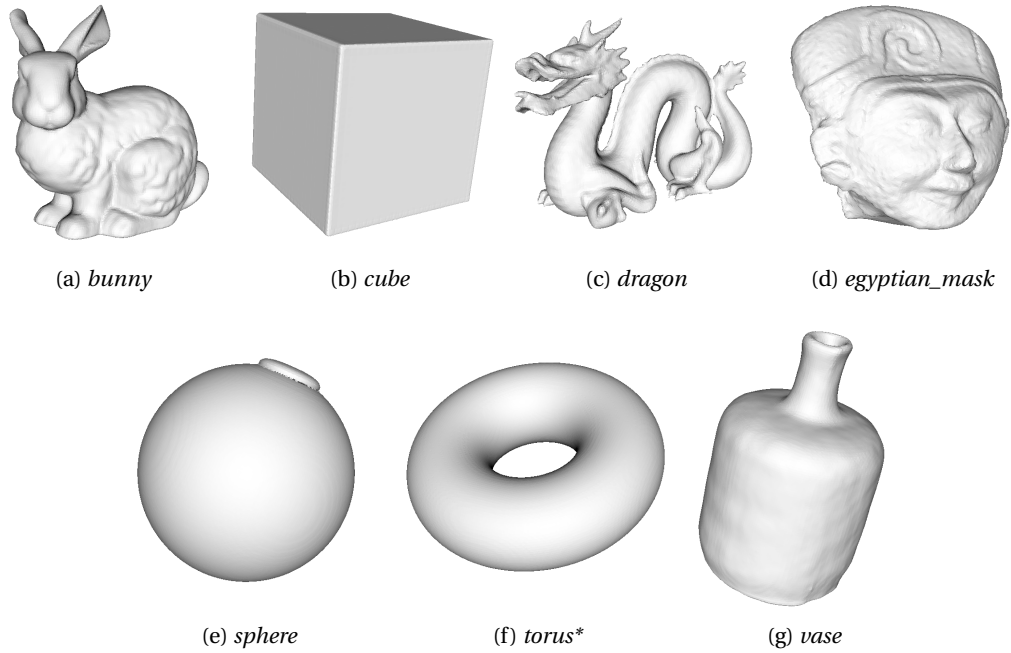


Figure 3.14 – Reference point cloud contents after conversion to meshes.

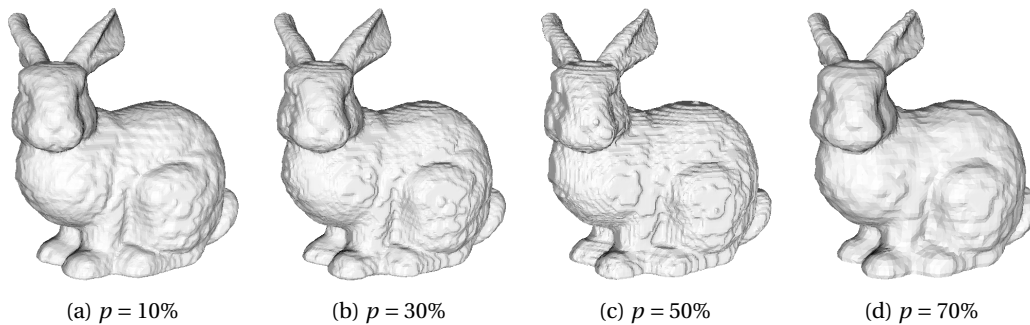


Figure 3.15 – Reconstructed *bunny* under all degradation levels from Octree-pruning.

frame was captured, leading to a total of 720 frames. The still images were then losslessly compressed with an H.264/AVC encoder, producing an animated video of 30 fps with a total duration of 24 seconds, which were inspected by the subjects to rate the visual quality of the models under assessment.

Testing environment

The subjective experiments were conducted in 5 laboratories: École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; University of Beira Interior (UBI), Covilhã, Portugal; University of Coimbra (UC), Coimbra, Portugal; University North (UNIN), Varaždin, Croatia

Table 3.8 – Subjects information per laboratory.

	EPFL	UBI	UC	UNIN	UP
Males	11	17	9	14	30
Females	9	5	11	6	14
Overall	20	22	20	20	44
Year span	21-37	21-50	21-54	19-57	19-59
Average age	28.88	30.59	29.45	26.45	23.32
Median age	28.39	28	23	21.5	22

and Univeristy of Patras (UP), Patras, Greece. The conditions of every test environment were adjusted to follow the ITU-R Recommendation BT.500-13 (ITU-R BT.500-13, 2012). The equipment that was installed in each laboratory can be found in Table 3.7, for every university.

Experimental design

At the beginning of each evaluation session, a training session took place in order to familiarize the subjects with the artifacts under assessment. The *torus* was selected for this purpose and, hence, it was excluded from the actual subjective tests. The training was performed using 3 animated video sequences that represented 3 different levels of degradation in order to indicatively illustrate the range of visible distortions. To avoid biases during testing, for half of the subjects the reference was placed on the right and the degraded content on the left side of the screen, and vice-versa for the rest. The presentation order of the stimuli was randomly picked per subject, while particular care was given to avoid displaying the same model consecutively.

An overall of 30 scores were obtained per evaluation session, considering that each subject assessed 6 test contents degraded in 4 distinct levels along with the hidden references. Information regarding the demographics of the test subjects are provided in Table 3.8.

Data processing

The MOS and CIs are employed to characterize the visual quality per stimulus. To compare the two rendering schemes, the performance indexes described in annex A.2 are employed.

3.4.3 Results

Subjective results

The results of the experiment conducted under raw point cloud rendering are reported in section 3.2.3. Hereafter, we present the subjective scores from the current experiment, which is based on visualization of the reconstructed mesh models. In particular, outlier detection

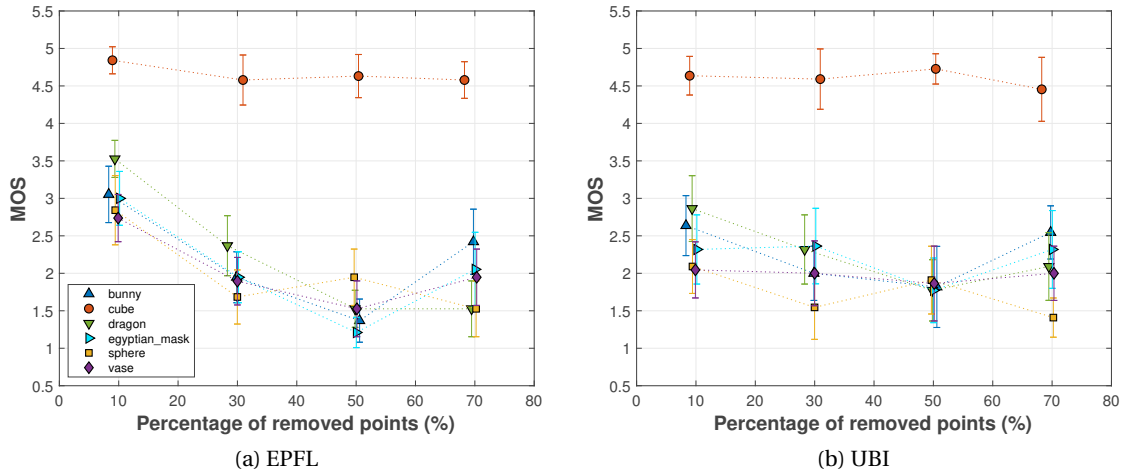


Figure 3.16 – Subjective scores against degradation levels per laboratory.

based on the Recommendation (ITU-R BT.500-13, 2012) was initially applied on the human opinions from each test laboratory, separately. In both EPFL and UNIN 1 outlier was found, in UP 6 outliers were identified, while no outliers were revealed for the rest of the laboratories.

The subjective scores for the distorted versions of the 6 contents under evaluation are shown in Figure 3.16, with the caption of each sub-figure indicating the laboratory from which they were derived. Notice that we provide plots only from two participants, since very similar curves are obtained for the rest. Notably, it can be observed that the MOS for *cube* remains high, independently of the level of distortion. For the other meshes the MOS is increasing as the target percentage of removed points is decreasing, with the exception of the lowest degradation level, where the MOS is stable or even slightly higher. This can be explained by the lower degree of polynomial functions that were used by the reconstruction algorithm to produce the surfaces of the mesh, leading to smoother surfaces. This was caused by the vast reduction of the number of points for these distorted contents. An example can be viewed in Figure 3.15, where less artifacts are visible in the stimulus of Figure 3.15a when compared to the stimulus of Figure 3.15b. Another observation is the generally low scores that are given to all contents, excluding *cube*. It is evident that the reconstructed models are not rated as visually appealing, which might be the result of a sub-optimal configuration for the reconstruction algorithm. Yet, in the absence of best practices, an exhaustive manual search for parameter adjustment per stimulus is too tedious, risking to introduce bias in the scoring distributions by favoring a particular content/stimulus over another. Thus, it was decided to employ the same, default configuration across all stimuli.

Inter-laboratory correlations

In Table 3.9, performance indexes for every combination of participated laboratories are provided without applying any regression model, showing in principle strong correlation

3.4. Rendering schemes

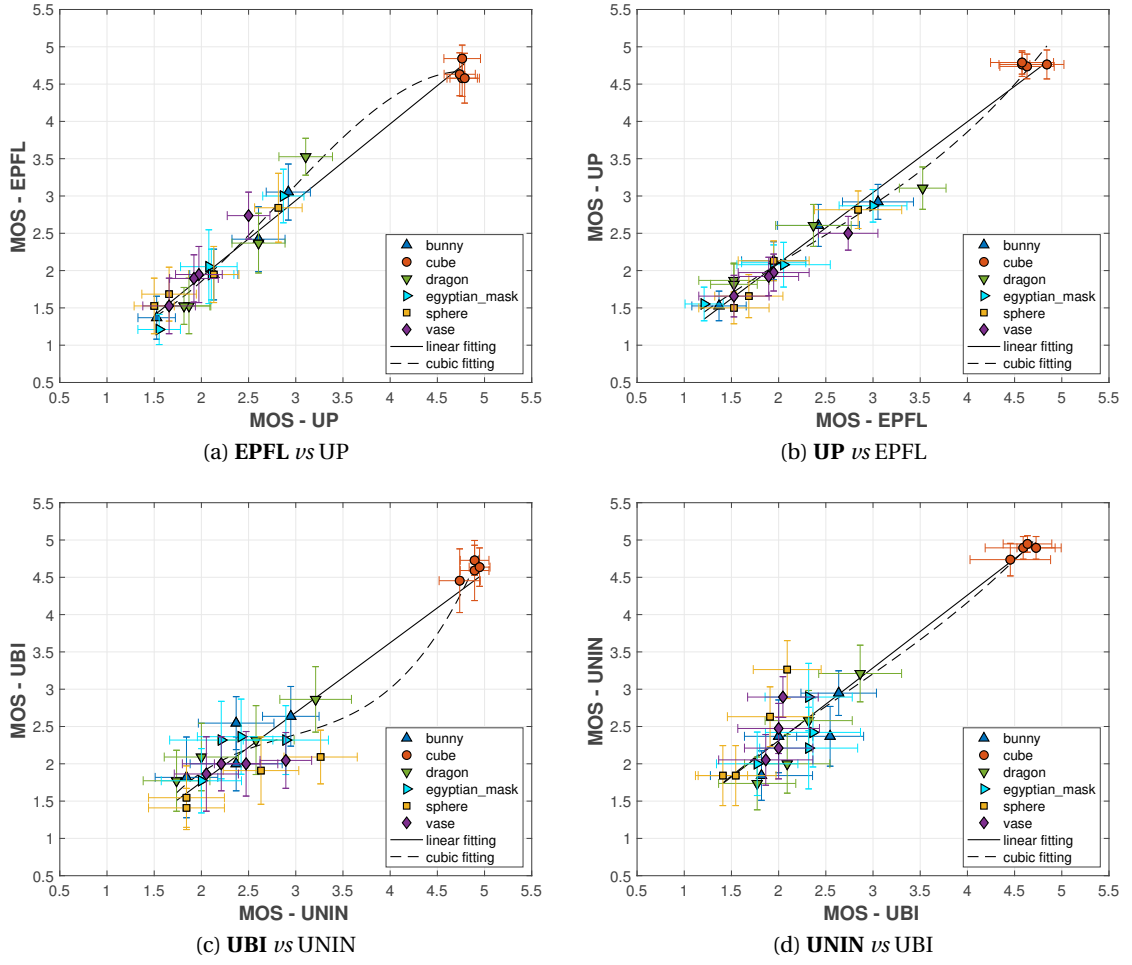


Figure 3.17 – Comparison of subjective scores obtained from different laboratories (Bold text represents the ground truth).

Table 3.9 – Performance indexes without using any fitting function for inter-laboratory correlation (Bold text represents the ground truth)

	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
EPFL vs UBI	0.940	0.890	0.391	0.417	100%	0%	0%	83.33%	0%	1.09%	15.58%
EPFL vs UC	0.967	0.922	0.298	0.292	100%	0%	0%	88.77%	0%	0.72%	10.51%
EPFL vs UNIN	0.975	0.927	0.384	0.375	100%	0%	0%	88.41%	0%	0.72%	10.87%
EPFL vs UP	0.987	0.969	0.199	0.125	100%	0%	0%	92.75%	0%	6.16%	1.09%
UBI vs UC	0.973	0.869	0.281	0.208	100%	0%	0%	94.93%	0%	3.62%	1.45%
UBI vs UNIN	0.954	0.838	0.435	0.333	95.83%	0%	4.17%	94.57%	0%	4.35%	1.09%
UBI vs UP	0.971	0.904	0.284	0.125	100%	0%	0%	80.80%	0%	19.20%	0.00%
UC vs UNIN	0.973	0.903	0.338	0.208	100%	0%	0%	92.39%	0%	4.71%	2.90%
UC vs UP	0.985	0.948	0.195	0.125	100%	0%	0%	85.87%	0%	14.13%	0.00%
UNIN vs UP	0.984	0.938	0.288	0.333	100%	0%	0%	84.06%	0%	15.94%	0.00%

between subjective scores in each case. Provided that different desktop equipment was installed in each university, as reported in Table 3.7, a general conclusion that can be drawn is that subjective evaluations using this data representation do not highly depend on the specifications of the monitor. In general, the PLCC and SROCC coefficients are high, whilst the RMSE and OR values remain low, indicating high accuracy and consistency of the ratings, respectively. The FR, which is the most severe type of error remains 0%, while the CD is above 83.3%, which indicates that for a big percentage of pairs of stimuli, the same conclusions can be drawn by two test labs. In UP we observe smaller CIs with respect to the rest of the test laboratories, as result of the higher number of participants; specifically, the CIs of EPFL, UBI, UC and UNIN are on average 41.39%, 70.96% and 61.91% and 48.79% larger than the CIs of UP, respectively. This explains why the FD percentages are consistently high when the UP is not set as the ground truth, meaning that there are cases where scores obtained from the ground-truth laboratory suggest that two stimuli are rated statistically equivalently, whereas scores from the UP indicate statistical distinction. Based on the performance indexes, there is a small percentage of stimuli that is over-estimated in UBI when compared to the corresponding scores in UNIN. However, in the rest of the cases, a CE of 100% is achieved. In Figure 3.17, scatter plots showing the comparison of MOS between two pairs of universities are indicatively presented along with every fitting function, to visually interpret the strong correlation results. Notice that the pair UBI and UNIN, essentially, denotes the worst combination according to the performance indexes; yet, the correlation is still good. Very similar graphs are obtained for the rest of the combinations.

Comparison of rendering schemes

Finally, the subjective scores from this test (excluding *egyptian_mask*) are compared to ratings derived in a previous experiment, where the visual quality of the same degraded point clouds was assessed without enabling any reconstruction algorithm for rendering. The latter was conducted in the EPFL laboratory using an interactive variant of the same test method (i.e., simultaneous DSIS with 5-rating impairment scale), under identical environmental conditions and test equipment, as described in section 3.2. Thus, the scores of EPFL from the current experiment are used to issue the statistical analysis. No fitting, linear and cubic regression models were applied, as reported in Table 3.10, while in Figure 3.18, scatter plots indicating the correlation between the two experiments are provided.

Based on our analysis, the correlation between these two tests is poor. Despite the fact that a relatively high percentage of CE is observed, there is no consistent trend, with some stimuli being over-estimated and some others under-estimated when using one rendering scheme with respect to the other. Moreover, the low percentages of CD indicate that scores from the two experiments frequently lead to statistical differences for the visual quality of two stimuli. The weak correlation is also evident from Figure 3.18, where the scattered data points indicate different ranking order, lack of monotonicity and accuracy. Our results suggest that the visual quality of identically distorted contents is affected by the use of a surface reconstruction

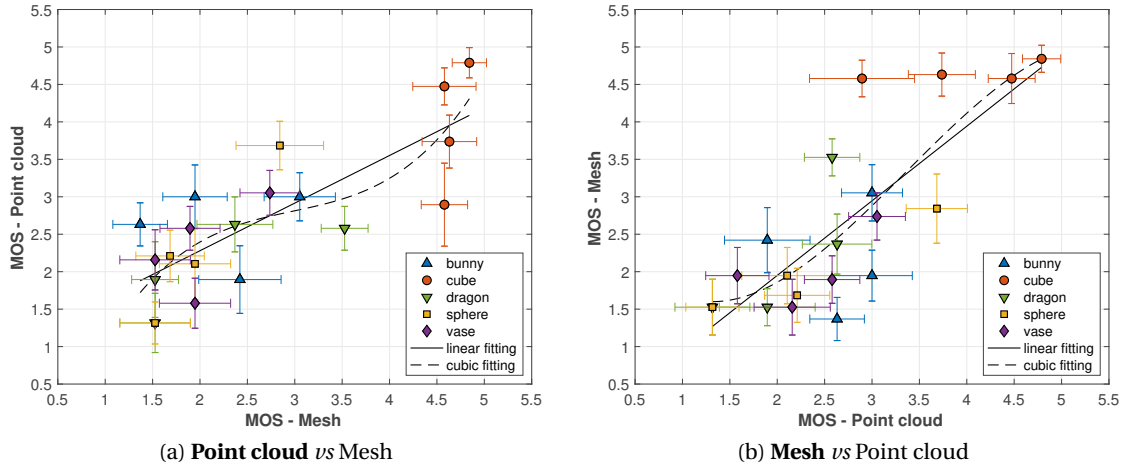


Figure 3.18 – Comparison of subjective scores obtained under both rendering schemes (Bold text represents the ground truth).

Table 3.10 – Performance indexes to compare the rendering schemes (Bold text represents the ground truth).

Point cloud <i>vs</i> Mesh											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.796	0.729	0.723	0.600	80%	10%	10%	67.37%	0.00%	18.95%	13.68%
Linear fitting	0.796	0.729	0.576	0.650	90%	5%	5%	63.68%	0.00%	13.16%	23.16%
Cubic fitting	0.804	0.729	0.565	0.600	85%	10%	5%	68.42%	0.00%	15.26%	16.32%
Mesh <i>vs</i> Point cloud											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.796	0.729	0.723	0.600	80%	10%	10%	67.37%	0.00%	13.68%	18.95%
Linear fitting	0.796	0.729	0.721	0.600	75%	15%	10%	67.37%	0.00%	13.68%	18.95%
Cubic fitting	0.808	0.729	0.702	0.500	80%	10%	10%	68.95%	0.00%	5.79%	25.26%

algorithm for rendering purposes.

To obtain a watertight model, commonly, the coordinates of the points are modified to best match the fitting surfaces. This lossy procedure leads to geometric deviations that might reduce or enhance artifacts as perceived on the originally distorted point cloud data, while an extra ambiguous step is introduced by judging the visual quality of the latter using a different 3D visual data representation. Thus, it can be deduced that quality scores after enabling and disabling surface reconstruction to render point cloud data, might lead to different conclusions on the visual quality of a particular stimulus. Finally, provided that different reconstruction algorithms and configurations typically lead to different mesh representations, it becomes clear that experimental set-ups for quality evaluation of point cloud contents using such a rendering scheme should be cautiously designed.

3.5 Conclusions

In this chapter we examine and propose different evaluation methodologies and protocols for subjective quality assessment of point cloud contents. We compare different test methods, display equipment and rendering schemes, to draw conclusions regarding their statistical relevance. Our results show that despite point cloud being a modality with ambiguous notion of visual quality, high correlation is achieved between two of the most widely-adopted test methods, namely, ACR and DSIS. The former allows explicit assessment of the visual quality of the displayed model, however, it gives space to biases arising from implicit inter-content comparisons and personal preferences. For instance, adopting ACR, the *dragon* and *vase* were rated consistently lower in the presence of Gaussian noise in our experiment, due to higher complexity and geometric irregularity, respectively. The DSIS protocol found to be more consistent in terms of identification of the level of impairment of a degraded content, allowing though subjects to rate based on relative differences (i.e., geometric distances, or number of points) between the queried stimulus and the provided reference. The smaller CIs that were observed in the DSIS case, suggest the participation of more subjects in ACR experiments.

Using the more consistent DSIS approach, we proceed by investigating the impact of adopting different display devices of different degrees-of-freedom in the context of subjective quality evaluation. Our results reveal that different rating trends are observed under the usage of different equipment as a function of the degradation type under assessment. In particular, although in the presence of Gaussian noise, scores obtained from the desktop and the HMD AR set-ups were found to follow very similar trends, this is not the case for Octree-pruning. Our study suggests that the former is a type of degradation that can be easier and more consistently identified even across different devices, while the latter is challenging in terms of identification, whose perception is affected by the display means. Thus, leading to the conclusion that visual quality assessment should be conducted using the target equipment for consumption.

In the last part, we aimed at investigating the impact of applying a different rendering mechanism for the testing material. In particular, using Octree-pruning as the sole degradation type, we enabled surface reconstruction for visualization of distorted point cloud data. This experiment was performed on five independent laboratories revealing high correlation among the participants, despite the variety of the displays that were employed. Using the scores of the same laboratory, a comparison between the ratings of subjects visualizing the same stimuli using two different visual data representations, namely sets of points and reconstructed surfaces, showed that the human opinions on visual quality are affected by the usage of surface reconstruction.

The generated data set and subjective scores collected from our experimentation efforts are made publicly available for research purposes. For more information, see annex E.

4 Quality evaluation of colored point clouds

The rendering approach that is employed to display point cloud data determines the visual outcome and strongly affects the perceived quality. Moreover, knowing how a content is rendered, may also determine the way it is acquired, transmitted and compressed. For point cloud contents, several rendering schemes have been proposed in the literature. The most common can be clustered as either (a) point-based, or (b) mesh-based techniques. In Figure 4.1, a point cloud model is presented at its original form, after point-based rendering and surface reconstruction, to provide an indicative illustration.

Reconstruction algorithms generate a polygonal mesh from a point cloud model. This 3D visual data representation is defined by a set of vertices together with associated faces expressed through connectivity information. Point clouds can be converted to polygonal meshes using a wide range of methodologies, with simple triangles being typically used as faces. Although meshes are the prevailing representation for 3D objects, they denote costly rendering approaches in terms of time and computational complexity for real-time applications. Moreover, the processing to infer high-quality underlying surfaces from discrete samples is typically lossy, meaning that points displacements and reductions might occur. Such lossy procedures make impossible the 1-1 mapping between point clouds and meshes, which is rather problematic in applications where inverse conversions are required. Furthermore, the types and the levels of visual distortions turn out to be very different in converted meshes with respect to the original point cloud representations, governed by the configuration and the selection of the reconstruction scheme. Consequently, it is clear that using meshes for deciding the quality of point cloud data is a sub-optimal approach, confirmed also from correlation results that are presented in section 3.4.

The concept of using points as primitive elements to represent 3D models has been proposed in a pioneering work presented in (Levoy and Whitted, 1985). In its simplest form, each point is represented by a single pixel. However, fidelity-wise, this is rather inefficient, while also it denotes an unnatural way of 3D modelling consumption, if no additional processing takes place to fill holes that appear in case the resolution of the model is sparser than the image grid that is projected. Furthermore, the extrapolation of the model surfaces from individual point



Figure 4.1 – Point cloud content using different rendering methods.

samples, introduces high levels of ambiguity in perception, making also the definition of visual quality a challenging task. Thus, in point-based rendering techniques, points are typically replaced by splats to acquire volumetric dimension and more efficiently approximate the surface. Additional attributes of points such as color, or normals can be additionally reflected on the splats. The size of each splat can be either fixed, or adaptive across a model, while the geometric shape can be either 2-D or 3-D. In the former case, the orientation of 2D splats can be either predetermined in the world coordinate system, or adjustable to face the camera of the user. Moreover, it is common to take under consideration the distribution of points in a neighborhood to orient and stretch the corresponding rendering primitive in order to fill the local region accordingly (e.g., major and minor axis of an ellipsoidal). More sophisticated techniques enable surface splatting and texture filtering, or additional processing in the image space for hole filling, in order to produce high-quality surface approximations and water-tight models. Yet, these approaches despite the high performance in terms of visual appearance, introduce extra workload in the rendering pipeline, leading essentially to a trade-off between complexity and visual quality.

In this chapter we employ simple point-based rendering approaches in the context of subjective quality assessment of colored point clouds. We initially experiment with different primitive elements that are employed to display the point cloud data, in order to understand whether splat properties, such as the shape and orientation, influence the preference of human subjects. Then, we proceed with the design and realization of two subjective quality

evaluation experiments under substantially different rendering schemes, and we investigate their impact in deciding the visual quality of point cloud contents subject to geometry and color compression artifacts based on statistical analysis. As part of the study, conclusions regarding the efficiency of the selected codec, and insights related to the rating trends from the participants are provided.

It should be noted that our implementations should not be confused with the well-known surface splatting algorithms (Zwicker et al., 2004). The term splat, in our context, refers to a simple geometric shape that replaces a point in order to give a dimensional aspect with the possibility to re-size, orient and color. Thus, the rendering schemes denote low-complexity solutions, which are suitable for real-time communication systems, at the expense of lower visual quality.

This chapter is based on material that has been published in (Torlig et al., 2018a; Alexiou and Ebrahimi, 2019).

4.1 Point rendering primitives

The usage of splats provides an efficient way to represent point clouds with pleasing visual results under conditions, with the main advantage of not introducing any lossy intermediate processing in point cloud topology. There are multiple geometric shapes that can be used to represent a point. However, it remains unknown whether advantages that are coming from the properties of a particular selection are preferred over others.

In this study, we aim to shed some light on the issue by determining users preference under different point rendering primitives in terms of visual appearance. In particular, we conduct an experiment using three out of the most popular 2D and 3D splat shapes to render a point cloud, namely, disks, cubes, and spheres. As a result, different visual representations are obtained and shown to human subjects, which are then asked to choose the one they prefer. The assessments are conducted in a desktop set-up and the participants are allowed to interact with the stimuli under evaluation in a side-by-side fashion, essentially enabling a pairwise comparison protocol. Note that the usage of different geometric shapes might lead to the perception of different artifacts. For instance, the less computationally demanding 2D splats lead to a more refined representation of curves by better approximating the underlying surface. This is achieved provided that, in our implementation, the orientation of the splats is defined by the normals of the points. At the same time, mis-oriented 2D splats may lead to visible holes. On the contrary, 3D shapes are better suited for perception of watertight surfaces and do not depend on normal vectors. Yet, they introduce higher load during rendering and lead to rougher approximations of curves. Thus, in this experiment, the objective is to understand whether particular visual artifacts occurring under the usage of corresponding point rendering primitives are perceived as more annoying than others, and under which conditions. To enable a fair comparison between the geometric shapes to the maximum possible extent, we opt splat sizes that lead to the same maximum projected area onto the screen.



Figure 4.2 – Original point cloud contents.

4.1.1 Data set

Content selection

A total of 8 static colored contents, namely, *amphoriskos*, *head*, *egyptian_mask*, *shiva*, *longdress*, *loot*, *queen*, and *redandblack* are employed in this experiment. Every model belongs to either the JPEG¹ or the MPEG² point cloud repositories, except of *amphoriskos*, which is recruited from the Sketchfab³ platform. In Figure 4.2, the original versions of these point clouds are depicted.

Content preparation

The contents were selected and prepared in order to obtain categorical influencing factors that may affect the results. For instance, four contents were selected to represent “Objects”, and four others to represent “Human” figures. Moreover, considering that the original number

¹<https://jpeg.org/plenodb/>, last accessed 12/2020

²<http://mpegfs.int-evry.fr/MPEG/PCC/DataSets/pointCloud/CfP/>, last accessed 01/2020

³<https://bit.ly/3nekULm>, last accessed 12/2020

Table 4.1 – Point cloud contents characterization.

Content	Type	Sparsity	Input points	CC Distance	Output points
<i>amphoriskos</i>	Object	High	147,420	0.002	49,440
<i>head</i>	Object	Low	14,025,710	0.032	187,282
<i>egyptian_mask</i>	Object	Low	272,689	0.007	189,439
<i>shiva</i>	Object	High	1,010,591	0.045	54,308
<i>longdress</i>	Human	Low	857,966	1.5	189,398
<i>loot</i>	Human	High	805,285	3	55,004
<i>queen</i>	Human	Low	1,000,993	1.7	218,681
<i>redandblack</i>	Human	High	757,691	3	51,318

of points lies in a wide range, down-sampling was applied aiming at two different levels of sparsity, namely, “Low” and “High”. The down-sampling was implemented in CloudCompare software, by setting a maximum allowed distance between nearest neighbors and discarding intermediate points. Note that the models that belong to the type “Human” were originally voxelized at a voxel-depth of 10 bits, whilst the initial topology of the rest was more irregular with coordinates spanning in an arbitrary range. In Table 4.1, we present geometric characteristics, as well as the classification of every content regarding the influencing parameters of interest.

4.1.2 Methodology

Test method

A pairwise comparison was used in this experiment giving the option of tie in order to avoid forced decisions. The participants were asked to inspect both representations and choose their preferred one based on the following scale: (i) “A is better than B”, (ii) “No difference”, and (iii) “B is better than A”, where A and B refer to the presented stimuli, which were clearly annotated on the screen. This test method is well-known for its high discriminatory power and accuracy. It can be particularly useful when evaluating abstract perceptual dimensions, such as in this case, where the same models under different visual representations are assessed. Moreover, the fact that the users are not asked to provide grades, rather their preferences, implies less cognitive load and less bias introduced to the votes.

The deployed protocol allowed interactivity between the subjects and the displayed stimuli; that is, the participants were able to interact with the stimuli through the renderer by zooming, rotation, and translation using the mouse. The models were inspected simultaneously, while their views were in synch and adjusted to the selections of the user in real-time. Radio buttons were also installed in the graphical layout implemented in QT library⁴ to display the adopted

⁴<https://www.qt.io/>

ternary scale and allowed users to submit their preference through mouse-clicks. Moreover, there were no restrictions imposed in terms of time duration, or user navigation during evaluations. Finally, a free viewing inspection was adopted, with the subjects being able to adjust their sitting position with respect to the screen.

Rendering

The renderer described in annex D.2 was employed in this study, using three primitive geometric objects to represent splats, namely, disks, cubes and spheres. For fairness purposes, the default size of the source elements are set with the aim of attaining the same maximum projected area onto the screen. For disks, this is achieved when the normal vector is aligned to the camera direction, whereas for cube, when the space diagonal is aligned to the camera direction, forming the regular hexagon as a projected shape. In more details, for cubes, we use the unit cube, which leads to a projected area that spans between 1 and $\sqrt{3}$, depending on the camera direction. For disks and spheres, we set a radius of 0.743, which results to a projected area between ~ 0 and 1.73 for the former, and 1.73 for the latter.

For this experiment, we enable an adaptive splat size policy; that is, after the construction of each source element as described above, the size of each rendered splat is adjusted to local neighborhood sparsity considering $k = 10$ nearest neighbors. To orient the 2D disks, the normal vectors are estimated using the least-square plane fitting algorithm proposed in (Hoppe et al., 1992) over $k = 10$ nearest neighbors.

The background color of the renderer was set to black, while for display purposes and to avoid masking or enhancing visual artifacts, the default lighting conditions and flat shading were enabled in the scene. Finally, a perspective projection was enabled to provide users with a more realistic view.

Testing environment

The experiment was conducted in MMSPG laboratory, which follows the ITU-R Recommendation BT.500-13 (ITU-R BT.500-13, 2012). An ambient light of 15 lux was adjusted in the testing room, and a luminance of 115 cd/m^2 was set and measured on the screen. The monitor that used is an EIZO ColorEdge CG318 of 31.1 inches and 4096x2160 resolution.

Experimental design

The experiment was split in two stages: (a) the training and (b) the actual test. In the training, the subjects got familiarized with the rating scale and the interactivity part. For this purpose, another dummy content was selected, which was not used in the actual tests.

For half of the subjects, the position of a stimulus from a specific pair under comparison was set at the right side, while for the other half, it was displayed on the left side of the screen.

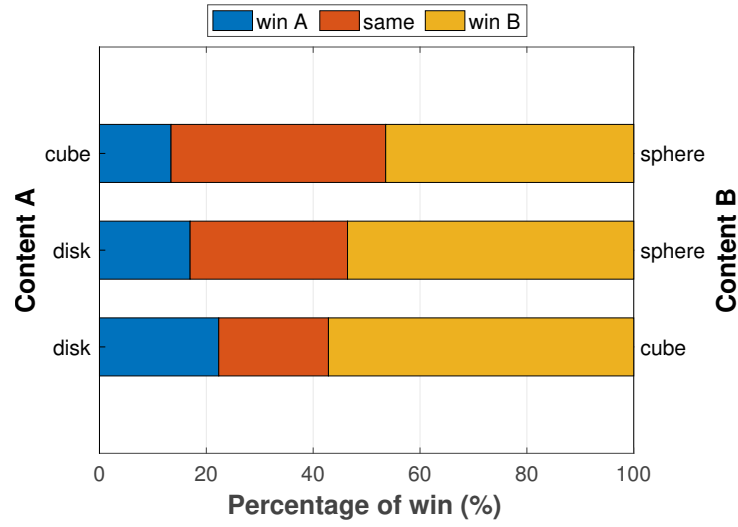


Figure 4.3 – Pair comparison of point rendering primitives across all contents.

Moreover, the presentation order of the pairs was randomly picked for every session. To reduce temporal references, it was intentionally avoided to show the same content consecutively. Moreover, the physical distance between a viewer and the monitor was adjusted at per each subject's preference.

For each of the 8 point clouds, we have 3 different point rendering primitives, which were all compared to each other per content, leading to 24 comparisons that were made from every subject. A total of 14 naive participants (12 males and 2 females) were recruited in the experiment after passing acuity and color vision tests based on Snellen chart and Ishihara plates. Their age was ranging between 20.9 and 27.7, with a mean of 23.2 and a median of 22.7.

Data processing

The votes that were obtained from the experiment are presented in the form of bars. Win and ties percentages are clearly separated and reported in order to draw conclusions regarding the preferred point rendering primitive. Moreover, we extract a MOS and a corresponding CI for each alternative representation using the BTL model, as described in annex A.1.2, by equally splitting the ties.

4.1.3 Results

In Figure 4.3, we present the votes of participants in the form of win percentages computed over all contents. These bars effectively summarize the aggregated preference of the subjects for the point rendering primitives that were considered and compared. From this graph, we observe a trend for subjects preferring 3D cube and sphere primitives against the 2D disk, as implemented in our rendering application. Moreover, when comparing models represented

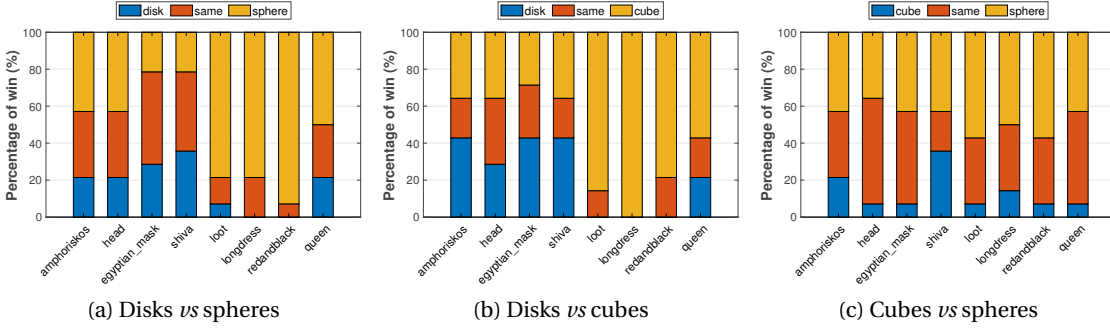


Figure 4.4 – Pair comparison of point rendering primitives per content.

with spheres and cubes, a high percentage of subjects does not indicate a clear preference; yet, when excluding the ties, the spheres are more frequently selected.

In Figure 4.4 the participants' selections are illustrated as win percentages for each pair combination of primitives, per content. When comparing disks against spheres, we remark that the latter are favoured by large margins for point clouds that represent human figures. Higher win percentages for disks are observed in the case of *egyptian_mask* and *shiva*, which are still limited below 36%, with most of the subjects declaring that they don't prefer one representation over the other, for these two contents. Similar, yet more polarized votes are observed when comparing models using disk against cube primitives. Specifically, favouring disks over cubes is even less frequent for point clouds depicting human figures, whereas for models representing objects, the participants state a preference for the disks with win percentages of up to 43% (against a maximum of 36% for cubes). When comparing cubes against spheres, we generally note a very small percentage of a clear win for the former over the latter. Moreover, it is very frequent for subjects to submit no preference among the two content representations. However, when participants differentiate the two versions, they clearly prefer spheres.

In Figure 4.5, the normalized quality scores and the associated CIs computed from the participants votes are presented for each point rendering primitive, after equally splitting the ties to the two classes under comparison. Results from this analysis are in alignment with our earlier observations. In particular, the highest score is associated to the sphere, thus, indicating the supremacy of this particular shape to represent point clouds models. Second follows the cube, confirming that 3D primitives are in principle favoured, and last comes the disk shape.

In Figures 4.6, we depict visual examples of point clouds that were shown to participants of this experiment, under all point rendering primitives that were considered. For demonstration purposes, we use the same visualizer that was employed in the subjective evaluations, while the models are captured from the same viewpoint to allow comparisons. In the case of *longdress*, displayed in Figure 4.6a, we remark that despite the more refined edges that are attained when using disks (e.g., face shape, nose width), impairments in the form of blurriness

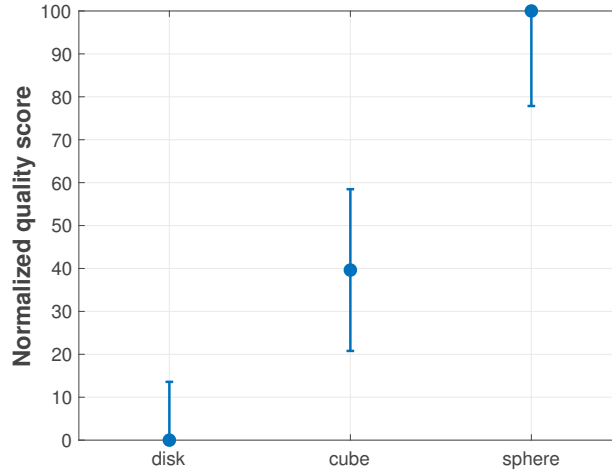


Figure 4.5 – Normalized quality scores from subjective preferences for the adopted point rendering primitives.

are noted especially in the facial characteristics of the model, which are clearly more annoying. Using cubes these distortions are limited, and they are further reduced by the usage of spheres. Moreover, color details are also sharper in the latter, which denotes another potential reason to explain the votes distribution indicated in Figure 4.4 for this particular content. In the case of *egyptian_mask*, shown in Figure 4.6b, marginal differences can be observed between the three content variants, which is also reflected on the equivalent win percentages, or the large number of ties that are observed in Figure 4.4. Similar observations are made for the rest of the contents.

It should be noted that, assuming a frontal view of a model and a normal vector that is parallel to the camera direction, the disk splat will be shown larger than the cube. Between disks and spheres, despite the theoretical equivalence of the projected area, the depth dimension in the latter case generally leads to better preservation of details. These remarks are more evident in the corresponding content representations of *longdress* in Figure 4.6a.

To identify whether the sparsity level of a content is a factor that influences the opinions of subjects regarding the preferred primitive shape, we repeat the same analysis by computing the normalized quality scores and the CIs over all models that are clustered as of “Low” and “High” sparsity per Table 4.1, separately. Additionally, to understand whether the type of content affects the subjects’ preference, the same procedure is repeated after aggregating the preference scores for all point clouds that depict “Human” figures and “Object” models.

Based on our results, illustrated in the first row of Figure 4.7, we observe that similar scores are associated to the point rendering primitives for both cases of sparsity levels. However, when splitting the point clouds per type of represented content, indicated in the second row of Figure 4.7, we observe different trends. In particular, when showing “Human” figures, subjects prefer cubes over disks, however, when presenting “Objects”, there is a tendency of



(a) *longdress*



(b) *egyptian_mask*

Figure 4.6 – Content representations using disks, cubes and spheres as point rendering primitives from left to right.

favouring disks over cubes, indicating that different shapes might be preferred as a function of the displayed model and its intrinsic geometric properties. In this result, it should be

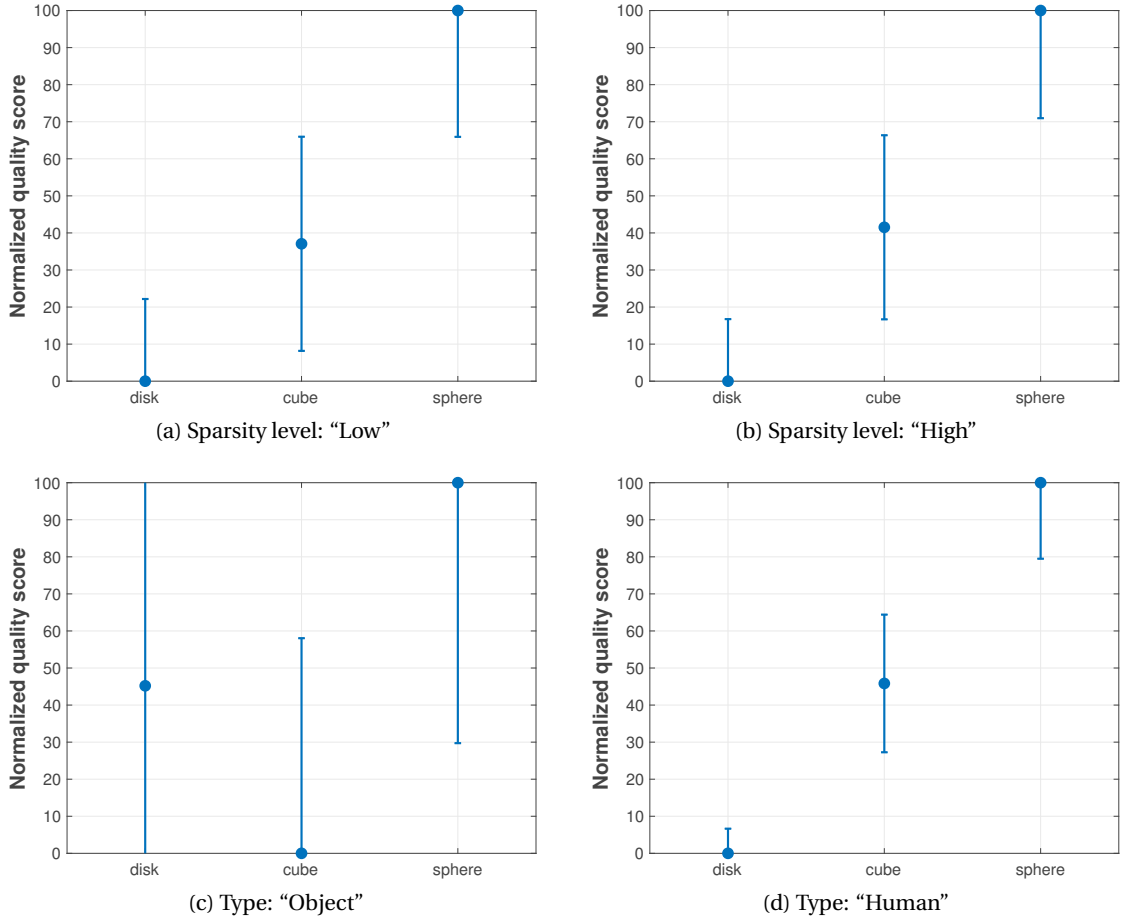


Figure 4.7 – Subjective scores after aggregating preferences over point clouds of the same sparsity level in the first row, and the same content type in the second row.

recalled and accounted the fact that the topology of models composing the “Objects” sub-set includes more outlying points and missing regions, with respect to models that belong to the “Human” sub-set. Finally, the CIs of the normalized scores of Figure 4.7c are large, thus, no safe conclusions can be drawn.

4.2 Point-based rendering schemes

In this section we examine two ad-hoc point-based rendering alternatives that are developed for point cloud subjective quality evaluation purposes. In particular, one rendering scheme stems from previous efforts detailed in section 4.1 and makes use of primitive geometric objects that replace point samples with a custom shape and adaptive size. This technique will be referred to, hereafter, as splat-based. The second rendering scheme, namely voxel-based, quantizes the topology of the point cloud under inspection in real-time, with the outcome being orthographically projected on a 2D image plane that is consumed by the user. The two

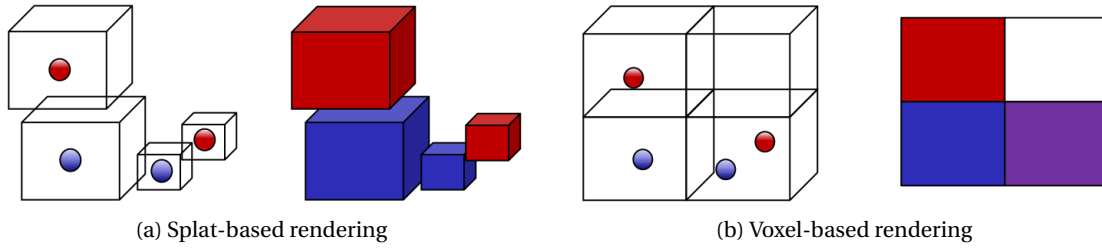


Figure 4.8 – Operational logic of the rendering schemes under evaluation.

rendering schemes follow different rationales on delivering the content to the observers, which are summarized in Figure 4.8. The first, is based on displaying the entire volumetric element, which is represented by a cube of an arbitrary size in this experiment and is employed to represent a point sample, whereas in the latter, the point is projected to a fixed neighborhood of pixels. Thus, visible artifacts of different nature occur. Using the splat-based approach, the point cloud models are perceived as watertight. However, during user inspection and as the camera is rotating, different splat sizes might be observed due to orientation changes, triggering small-scale flickering artifacts. More importantly, at lower geometric resolutions, the amplification of the splat sizes that is enabled to avoid the perception of hollow regions, leads to coarser surface approximations. On the contrary, in the case of the voxel-based renderer, artifacts in the form of missing pixels are perceived for sparser content representations. For the purpose of this study, a state-of-the-art codec is employed and multiple encoding configurations using different combinations of geometric and color degradation levels are applied. The degraded stimuli are evaluated using the rendering solutions in separate experiments that are conducted under identical set-ups. Using both sets of quality scores, the performance of the encoder is analysed, the preferences of subjects are statistically determined, and influencing factors are identified. Finally, the two rating distributions are compared in order to examine whether the developed rendering approaches lead to the same conclusions regarding the subjective quality characterization of the same stimuli.

4.2.1 Data set

Content selection

A total of 7 static contents with diverse characteristics were selected for the experiments. In particular, both human figures and inanimate objects were considered, each having different levels of geometry and color details. The *longdress*, *loot*, *redandblack*, and *statue_klimt* contents were chosen from the MPEG repository⁵ and belong to the first type. The *romanoillamp* and *biplane* models were selected from the JPEG Pleno repository⁷, while the *amphoriskos*

⁵6

⁷<https://jpeg.org/plenodb/>, last accessed 12/2020

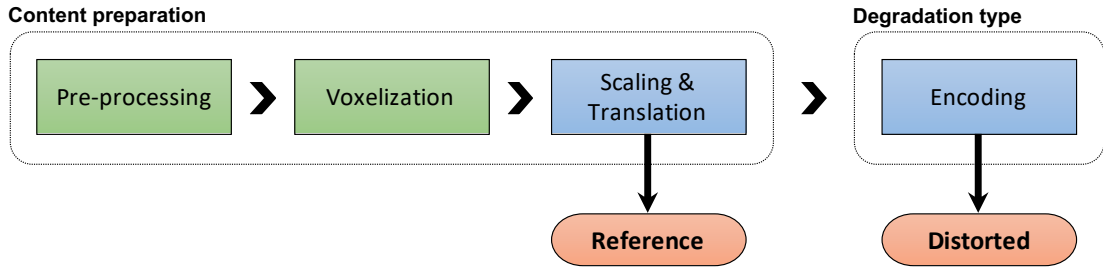


Figure 4.9 – Pre-visualization processing workflow.

point cloud was found in the online platform Sketchfab⁸. Such point clouds are typically scanned through depth sensors that provide either directly or indirectly a cloud of points representing their 3D shape. Typical use cases involve applications where such models are consumed from the outside.

Content preparation

The point cloud contents were initially prepared based on the work-flow indicated in Figure 4.9. In particular, pre-processing and voxelization stages were enabled in order to reduce influencing factors, such as the number of points and the geometric structure. The scaling and translation step ensure that the coordinates of the models are spanning in the same range at the input and the output of the selected encoder. Please note that with green color we annotate preparation steps that were issued on a subset of the test contents, whilst with blue color we specify processing that was enabled on the whole data set. Below, we provide implementation details for every stage of preparation.

Pre-processing: This step was enabled in order to ensure a narrow range for the number of points across contents. Specifically, *biplane* is provided in multiple versions that correspond to different scans. In this experiment, we used a combined version that provides a fully reconstructed model (i.e., *1x1_Biplane_Combined_000*), which consists of approximately $106 \cdot 10^6$ points and introduces heavy workload in the rendering pipeline. To reduce this number to acceptable limits, we applied subsampling using CloudCompare, by setting a maximum allowed distance between nearest neighbors equal to 0.009. For *amphoriskos*, the original model is represented by approximately $200 \cdot 10^3$ points. To increase the resolution for a better content representation, we initially applied the Poisson Surface Reconstruction algorithm (Kazhdan and Hoppe, 2013), as implemented in CloudCompare and using the default configurations with 1 samples per node. The normal vectors associated with the coordinates of the original model were employed for this purpose. From the reconstructed mesh, $1 \cdot 10^6$ points were sampled by randomly picking a given number on each triangle, using the same software. Regarding *romanoillamp*, the associated connectivity information that is originally carried

⁸<https://bit.ly/3nekULm>, last accessed 12/2020



Figure 4.10 – Reference test contents.

Table 4.2 – Geometric description of every reference content.

	<i>amphoriskos</i>	<i>biplane</i>	<i>longdress</i>	<i>loot</i>	<i>redandblack</i>	<i>romanoillamp</i>	<i>statue_klimt</i>
Points	828,820	773,447	857,966	805,285	757,691	636,097	482,941
Min D	$9.78 \cdot 10^{-4}$	$9.78 \cdot 10^{-4}$	$10.11 \cdot 10^{-4}$	$10.20 \cdot 10^{-4}$	$10.35 \cdot 10^{-4}$	$9.78 \cdot 10^{-4}$	$9.78 \cdot 10^{-4}$
Max D	$23.94 \cdot 10^{-4}$	$470.83 \cdot 10^{-4}$	$22.61 \cdot 10^{-4}$	$20.39 \cdot 10^{-4}$	$25.36 \cdot 10^{-4}$	$693.76 \cdot 10^{-4}$	$100.17 \cdot 10^{-4}$
X/Y/Z	0.60/1/0.68	0.65/0.23/1	0.40/1/0.20	0.35/1/0.41	0.44/1/0.30	1/0.45/0.51	0.30/1/0.29

with this content was discarded, while all the vertexes were kept to represent the point cloud. For the rest of the contents, no pre-processing was applied.

Voxelization: With this operation, we ensure a regular-spaced geometric structure for the point clouds in order to avoid biases that may be introduced by either the encoder or the rendering scheme. In particular, given that a subset of our data set (i.e., human figures) was already voxelized, we converted the continuous geometric representations of the rest of the contents (i.e., object models) into sets of voxels that lie in three-dimensional lattices of octree-depth equal to 10, in order to remove this influencing factor (i.e., irregular geometric structure) from our results. See annex B.2 for implementation details.

Scaling & Translation: This step ensures that the geometry of both the reference and the distorted contents lies in the same dynamic range. Specifically, the selected codec produces point clouds with output coordinates that are proportionally located in the range $[-0.5, 0.5]^3$

4.2. Point-based rendering schemes

Table 4.3 – Percentage of discarded points, and geometry and color bpp per encoded stimulus.

Content	Octree-depth	Percentage of discarded points	Geometry bpp	Color bpp		
				QP = 10	QP = 50	QP = 90
<i>amphoriskos</i>	OD = 10	0%	5.006	0.301	1.004	2.889
	OD = 09	46.08%	1.561	0.188	0.612	1.764
	OD = 08	83.39%	0.400	0.078	0.234	0.652
<i>biplane</i>	OD = 10	0%	2.890	0.589	2.101	4.926
	OD = 09	67.31%	0.618	0.209	0.686	1.623
	OD = 08	91.96%	0.142	0.069	0.191	0.430
<i>longdress</i>	OD = 10	0%	2.520	0.347	1.169	3.423
	OD = 09	70.37%	0.649	0.125	0.414	1.178
	OD = 08	92.24%	0.169	0.047	0.134	0.358
<i>loot</i>	OD = 10	0%	2.556	0.182	0.561	1.716
	OD = 09	70.01%	0.662	0.073	0.213	0.636
	OD = 08	92.16%	0.173	0.034	0.078	0.210
<i>redandblack</i>	OD = 10	0%	2.694	0.199	0.632	2.037
	OD = 09	68.91%	0.699	0.084	0.249	0.773
	OD = 08	91.87%	0.182	0.039	0.093	0.258
<i>romanoillamp</i>	OD = 10	0%	3.827	0.289	1.124	3.492
	OD = 09	57.53%	1.059	0.136	0.491	1.488
	OD = 08	87.86%	0.282	0.055	0.159	0.447
<i>statue_klimt</i>	OD = 10	0%	4.552	0.413	1.392	3.889
	OD = 09	49.44%	1.384	0.240	0.792	2.147
	OD = 08	85.00%	0.324	0.098	0.286	0.722

with respect to the input. To avoid the perception of different dimensions across contents, the voxelized point clouds were first scaled to the range $[0, 1]^3$ and then centered to the origin (0, 0, 0). The output of this step produces the reference contents of this experiment, as indicated in Figure 4.9. In Figure 4.10, the reference point clouds employed in the study are illustrated, while in Table 4.2, their intrinsic geometric characteristics is provided.

Degradation type

After preparation, the contents are encoded to produce the distorted versions evaluated by human subjects, as indicated in the last step of the work-flow as per Figure 4.9. For this purpose, a well-established point cloud codec is employed to account for representative compression distortions that are introduced in both geometry and color information.

Encoding: For encoding engine, the open source software that was employed as the anchor in the Call for Proposals issued by MPEG on point cloud compression (MPEG 3DG and Requirements, 2017) was selected. It denotes a typical octree-based compression scheme, with the

color attributes encoded using the JPEG algorithm, after they are mapped to an image grid using a depth-first order tree traversal; a detailed description can be found in (Mekuria et al., 2017a). To obtain a wide range of impairments, 3 quality levels for geometry and 3 quality levels for color degradations were defined. Specifically, the reference point clouds are encoded using octree tree-depth (OD) of 8, 9 and 10, to account for low, medium, and high geometry quality levels. Moreover, to reflect different levels of color fidelity, the JPEG quality parameter (QP) was set to 10, 50 and 90, respectively. The rest of the encoding options were identically set to the default configurations provided with the software release⁹. The point clouds were compressed using all possible combinations of geometry and color quality levels, leading to a total of 9 degradations per content. The output of this preparation step essentially produces the distorted testing material. In Table 4.3, the bits per input point (bpp) for geometry and color information are provided along with the corresponding percentage of discarded points, to grasp the sparsity level per stimulus. As expected, for the same OD and QP values, the distribution of both geometry and color bpp varies per content.

4.2.2 Methodology

Test method

The simultaneous DSIS protocol with 5-grading scale (5: *Imperceptible*, 4: *Perceptible, but not annoying*, 3: *Slightly annoying*, 2: *Annoying*, 1: *Very annoying*) was selected for its high accuracy and consistency in subjective quality assessment of point clouds, for both experiments. The reference and the distorted stimuli were visualized side-by-side by subjects, while being clearly annotated. An interactive extension of the protocol was enabled, by allowing users to inspect the models under evaluation at the selected viewpoints through zooming, rotation, and translation, without any restrictions in terms of time duration; thus, participants were able to spend as much time as needed for every individual assessment, before making their judgement. The subjects were required to submit a quality score through a GUI based on the level of impairment of the distorted model with respect to the reference counterpart. A free viewing protocol was followed, meaning that the users were allowed to adjust their position with respect to the screen.

Rendering

Splat-based rendering: The splat-based visualizer used in this study makes use of the rendering software described in annex D.2, and is implemented in the VTK library. The stimuli are presented side-by-side and subjects are able to interact with them in sync by rotation, translation and zooming. The visualizer covers the entire resolution of the screen. A screen-shot of the evaluation testbed as configured for the experiment is presented in Figure 4.11.

In this study, we opt for cubic splats of adaptive size, which provide a good compromise

⁹<https://github.com/cwi-dis/cwi-pcl-codec>



Figure 4.11 – Splat-based subjective evaluation testbed.

between computational overhead during rendering and visual quality. In particular, the splats are adjusted to the sparsity level of each local neighborhood that is defined around each point sample considering $k = 10$ nearest neighbors. An additional scaling factor is employed, amplifying the obtained splat sizes by a factor of 1.25. This value was empirically defined after expert viewing to avoid the perception of hollow regions with this data set. Note that the same splat scaling was applied for every stimulus.

The preparation of the stimuli was realized in an off-line mode, as described in annex D.2, and during the evaluations the prepared material was loaded into the renderer. This implementation allows fast responsiveness in user's interactions and low waiting times in between stimuli inspection. The background of the scene is set to (127, 127, 127) in RGB colorspace, to account for a non-distraction mid-grey color. The default lighting conditions in VTK were enabled without introducing any shading model. Finally, a perspective projection was enabled to better simulate realistic visual perception.

Voxel-based rendering: The voxel-based rendering software used in this experiment is described in annex D.1. It denotes an ad-hoc implementation developed in C++, handling both rendering and interactivity in real-time. The stimuli are presented side-by-side in a GUI developed in QT library that allows subjects to interact with them by rotation, translation and zooming through the mouse. A set of rating scores that users can choose from is also part of the GUI. A screen-shot of this evaluation testbed is presented in Figure 4.12.

In this study case, an image grid of resolution of 1024×1024 in selected to project the point cloud models. The choice of this resolution is made due to the octree depth of 10 that is set for



Figure 4.12 – Voxel-based subjective evaluation testbed.

all the reference models of our data set; thus leading to a 1-1 mapping between voxels and pixels at the initial pre-determined distance between the camera and the model. Upon user's actions (e.g., zooming in or out), the neighborhood of pixel over which a voxel is projected is correspondingly adjusted by a zoom factor.

The background color of the scene is set to (127, 127, 127) in RGB colorspace, to account for a non-distraction mid-grey color. An orthographic projection is used. It should be noted that, due to computational overhead from operations that are performed in real-time, the responsiveness of the system is moderate with respect to the splat-based rendering; yet, it was considered acceptable from the users without limiting the scope of the experiment.

Testing environment

The experiments were conducted in the MMSPG laboratory, which fulfils the Recommendation (ITU-R BT.500-13, 2012) for subjective evaluation of visual data representations. Specifically, the room is equipped with neon lamps of 6500 K color temperature, while the color of the walls and the curtains is mid gray. For both experiments, identical testing conditions were enabled. In particular, a typical desktop set-up involving an Apple Cinema Display of 27-inches and 2560x1440 resolution (Model A1316) was installed in the room. The brightness of the screen was always set to 120 cd/m^2 with a D65 white point profile, while the lighting conditions were adjusted for ambient light of 15 lux measured next to the screen, for both experiments.

Experimental design

Both experiments were split in a training and a testing stage. In the training, the subjects got familiarized with this type of visual data representation and the types of artifacts that would be assessed during the actual test. Additionally, the training served the purpose of letting participants adapt with the interaction part of the corresponding subjective evaluation framework. For this purpose, the *statue_klimt* content was selected; thus, it was excluded from the corresponding testing stages in both experiments. During training specific instructions and descriptions were delivered to the participants by the operator, instructing them to explicitly rate the visual quality of the degraded stimuli with respect to the reference, in terms of how annoying is for them the level of impairment. For both tests, at the beginning of each evaluation, a default frontal view of each content was displayed to every subject, which were then free to interact with the displayed models. In order to remove contextual effects from the quality scores, the position of the reference was randomly selected for every subject, and remained fixed across an entire testing session. Thus, for half of the subjects, the position of the reference was set at the right side of the screen with the distorted model placed at the left, and vice versa for the other half. Furthermore, the presentation order of the stimuli was randomly picked for every session. To reduce temporal references, we intentionally avoided showing the same content consecutively.

In each session, 6 contents and 9 degradations were assessed along with a hidden reference for sanity check, leading to 60 stimuli. A total of 20 subjects participated in the experiment using the voxel-based renderer, comprised of 6 females and 14 males, with an average age of 28 years old. In the experiment using the splat-based renderer, 20 subjects were recruited comprised of 10 males and 10 females, with an average of 26.7 years of age.

Data processing

The MOS and CIs are employed to characterize the visual quality per stimulus. To compare the two rendering schemes, we make use of the performance indexes that are described in annex A.2.

4.2.3 Results

Subjective results

The outlier detection algorithm defined in the ITU-R Recommendation BT.500-13 (ITU-R BT.500-13, 2012) was separately issued on the collected subjective scores from each experiment, in order to exclude subjects whose ratings deviated drastically from the rest of the scores. No outliers were identified, leading to a total of 20 out of 20 quality scores per stimulus, in each experiment.

The subjective results of the 6 contents that were involved in both tests are shown in Figure 4.13,

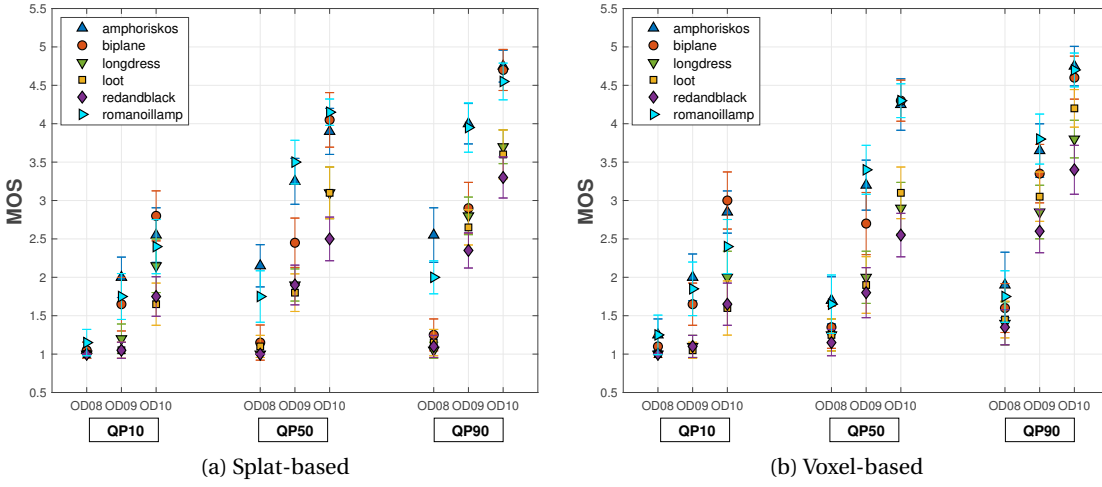


Figure 4.13 – Subjective scores against degradation levels using both rendering solutions.

with the caption of each sub-figure indicating the experiment from which they were collected. Specifically, the MOS along with the CI for every stimulus are presented against the geometry and color degradation levels. The naming convention is {ODXX, QPYY}, where OD and QP stand for octree-depth and the JPEG quality parameter, respectively, with $XX \in \{08, 09, 10\}$ and $YY \in \{10, 50, 90\}$ denoting the geometry and color quality levels.

Based on our results, it is evident that the quality scores from both experiments follow a similar trend. In particular, the subjective ratings vary per type of degradation for the same content. It is noted that for the sparsest versions (i.e., $OD = 08$), the mean score is increasing slowly as the color quality level is getting better, independently of the model. Higher rates of increase are observed as the geometric resolution becomes higher. This outcome essentially indicates that, when the geometric resolution of a content remains low, the overall perceptual quality is severely affected, regardless of color improvements. This rating behavior can be partially explained by the usage of the octree structure as basis for point cloud compression. In particular, by reducing the geometric resolution of an octree, an increasing number of points that belongs to the original model naturally falls within the leaf nodes. Thus, considering that the color of an output voxel is given by blending the colors of the input points in the same leaf node, color degradations in the form of blurriness are amplified at lower tree-depths.

It is important to note that, in the voxel-based renderer, the absence of geometric details is expressed by the presence of missing pixels, whereas in the splat-based renderer larger primitives are displayed. Specifically, in the former case, a single voxel is projected in a limited neighborhood of pixels, as a function of the zooming applied by user interaction. As such, missing information is mainly observed in contents encoded at an $OD = 8$, while for $OD = 9$, such artifacts become visible only when a viewer inspects the object from very close virtual distances after zooming. In the splat-based counterpart, the primitive sizes are adjusted to the sparser local neighborhoods, resulting in rougher representations of the underlying

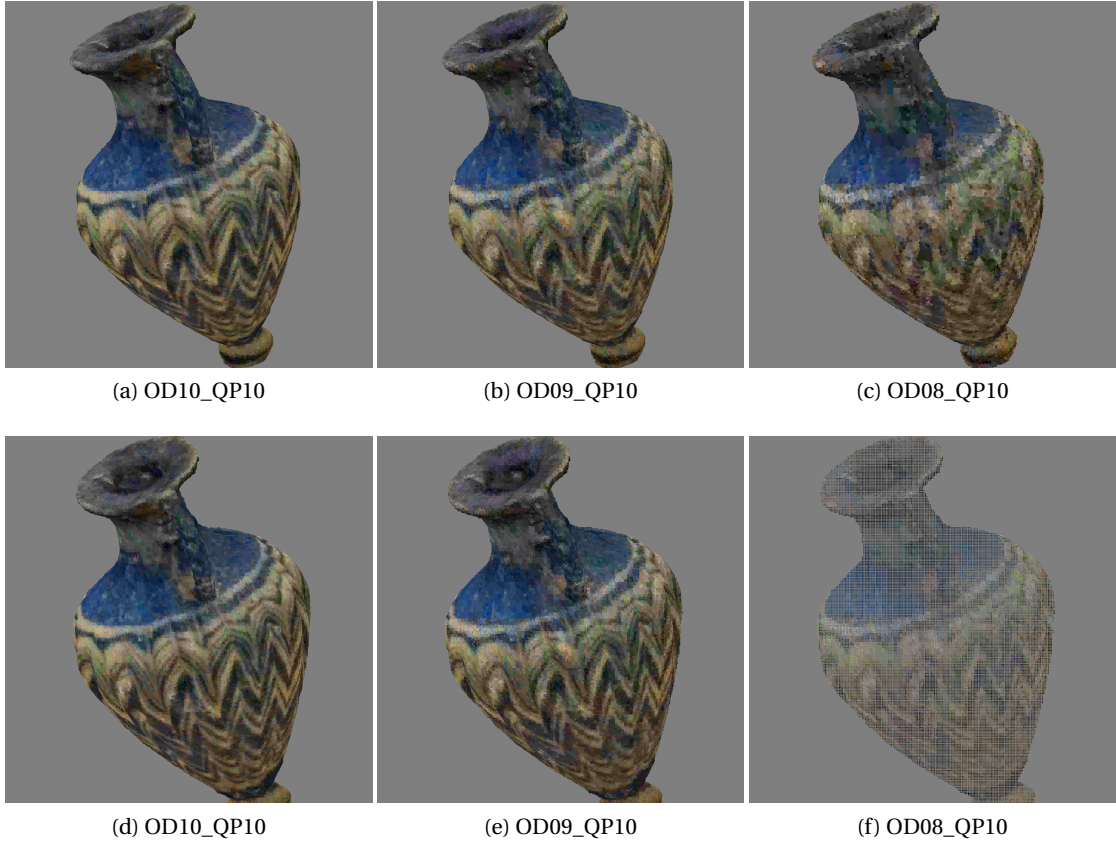


Figure 4.14 – Screen-shots of the frontal view of *amphoriskos*, encoded at the lowest color quality and every geometric degradation level. Top row: models displayed in the splat-based renderer. Bottom row: models displayed in the voxel-based renderer.

surfaces. Similarly, these artifacts are more evident at lower octree-depths. In Figures 4.14 and 4.15, various encoded versions of the *amphoriskos* and the *longdress* models are illustrated as displayed in both rendering schemes, in order to provide visual examples of representative distortion artifacts.

Another conclusion that can be drawn based on Figure 4.13 is that, for a specific degradation level, the perceptual quality notably differs depending on the type of content. In fact, subjects seem to be more critical with point clouds that represent humans, when compared to point clouds that represent inanimate objects. Smaller rating deviations are observed between contents that belong to the same type (i.e., humans, or objects), indicating that similar scoring distributions can be observed within the groups. A Wilcoxon signed-rank conducted on the scores reveals that there is a significant effect of content type on the distribution of the scores for both visualization methods, with large effect sizes (splat based: $Z = 16.242$, $p = 0.000$, $r = 0.494$; voxel based: $Z = 15.194$, $p = 0.000$, $r = 0.462$). This can be explained considering that (a) our perception is more sensitive to degradations on visual information that represents humans, and (b) the same acquisition means were employed to capture the

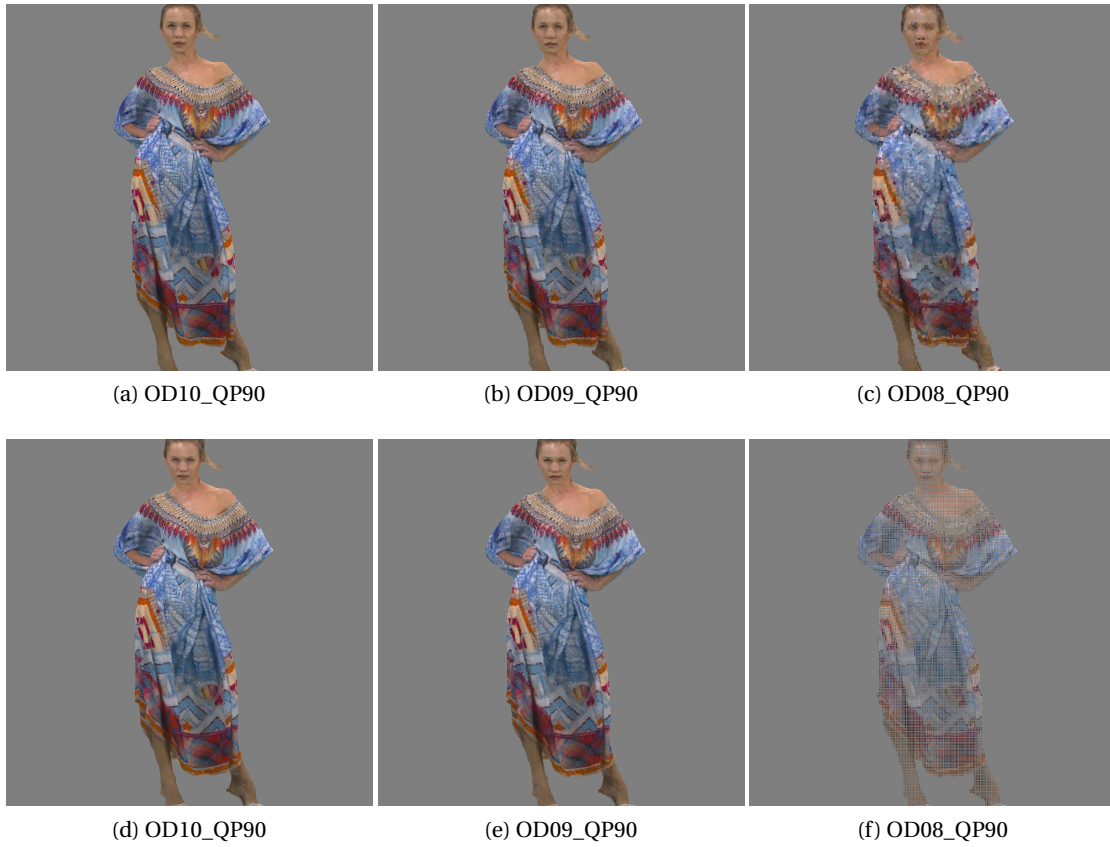


Figure 4.15 – Screen-shots of the frontal view of *longdress*, encoded at the highest color quality and every geometric degradation level. Top row: models displayed in the splat-based renderer. Bottom row: models displayed in the voxel-based renderer.

human models, which implies that they are subject to the same acquisition error. Thus, compression algorithms might result in similar distortion patterns. On the other hand, the objects were captured using different technologies, which leads to a wider range of acquisition and compression distortions.

Furthermore, by inspecting the total bit-rates of the encoded contents, as reported in Table 4.3, we conclude that higher bit-rates do not necessarily lead to better visual quality. For instance, for every stimulus, subjects from both experiments showed their clear preference in the combination of best color quality (i.e., QP = 90) with medium geometry (i.e., OD = 09), when compared to best geometry quality level (i.e., OD = 10) with the worst color quality (i.e., QP = 10); the latter combination requires higher bit-rates for every model. Although the obtained bpp values are codec dependent, such observations suggest that savings may be achieved by appropriate allocation of bits between geometry and color.

In Figures 4.16, we consolidate the results and present the MOS against the total bit-rates (geometry-plus-color), for both experiments. Different colors correspond to the testing con-

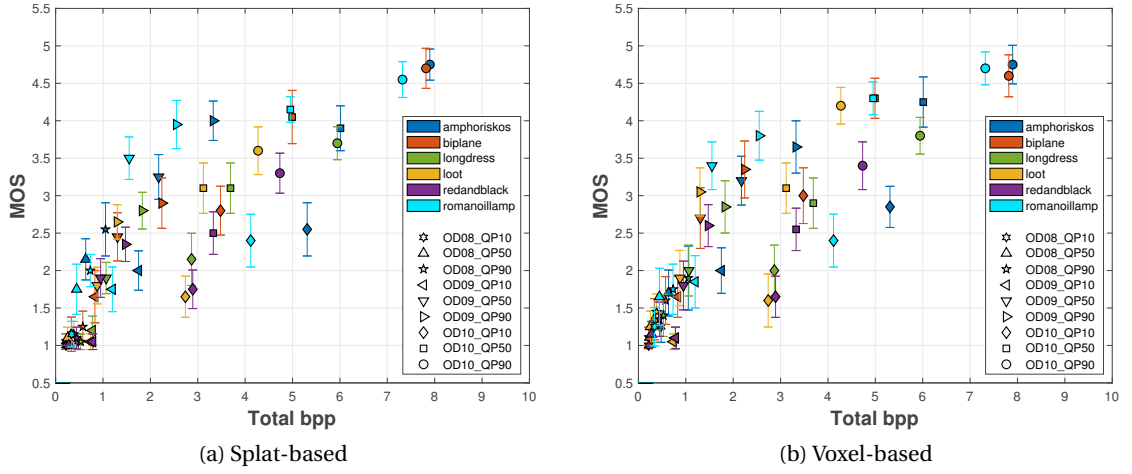


Figure 4.16 – Subjective scores against total bit-rates using both rendering solutions.

tents, whereas different markers correspond to the encoding configurations. In principle, the scatter plots using quality scores from both tests are rather similar. In both cases, the trends of preferring certain combinations of geometry and color distortions that lead to smaller bit-rate requirements over more expensive configurations are clear.

To further examine the aforementioned observations, a one-tail t-test at a 5% significance level was issued on the collected data, separately, per experiment. The null hypothesis assumes that the subjective mean of a stimulus at a particular color and geometry level is the same with the average score of another encoded version of the same content, which is subject to a different combination of degradations. This procedure is repeated for all contents. Aggregated numbers of preference are presented and color-coded in Figure 4.17, to reveal how often a particular combination that is shown in a row was preferred over a combination that is depicted in a corresponding column.

Based on Figure 4.17, it can be observed that for both rendering schemes, the combinations OD09_QP50 and OD09_QP90 are preferred 2 and 5 times in a total of 6 contents against the combination OD10_QP10. In a similar comparison at a lower geometry quality level, the combinations OD08_QP50 and OD08_QP90 were preferred 1 and 3 times against OD09_QP10 using the voxel-based renderer, and only once OD08_QP90 was favored over OD09_QP10 when using the splat-based rendering scheme. These results summarize that, for this codec, it is preferable to increase color quality when the geometric resolution is adequate, rather than further improving the topology at the expense of color. Moreover, under heavily distorted geometry, the priority should be to improve the topological information, since color improvements do not bring any visual benefits.

Regarding the different trends of subjective preferences between the two renderers, they are related to the different nature of visual artifacts, as reflected to the selected contents. Using the voxel-based renderer, color improvements at the lowest geometry quality level (i.e., OD

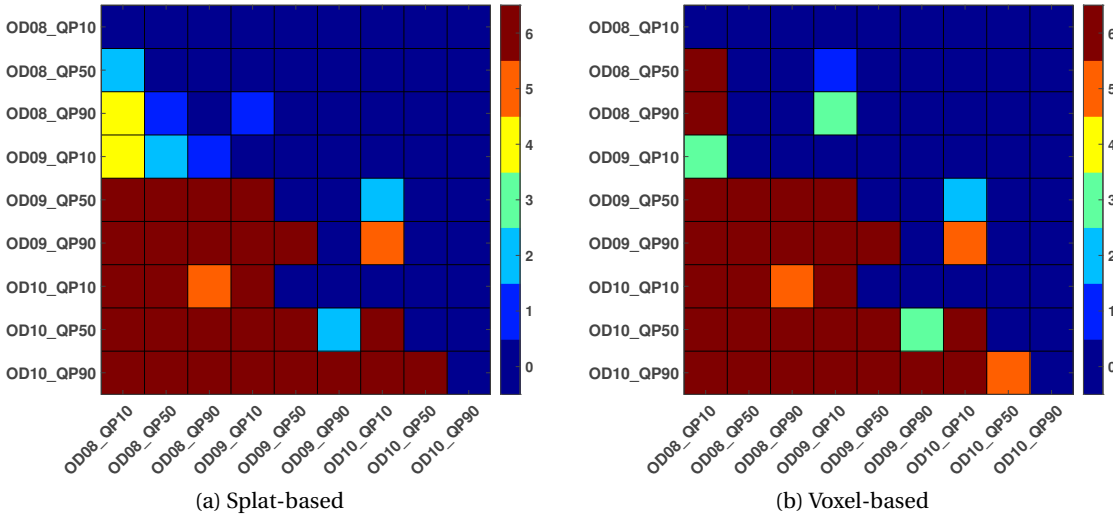


Figure 4.17 – Significance difference matrices at a 5% level per experiment, indicating the preference of subjects for a particular degradation against all others. Note that 0 and 6 denote the minimum and maximum numbers of preference, respectively, given a total of 6 contents.

= 08) are rated statistically higher, whereas using the splat-based renderer, these trends are not as consistent. In particular, OD08_QP50 and OD08_QP90 were preferred 6 times each against OD08_QP10 in the former case, whereas in the latter case they were preferred 2 and 4 times, respectively. According to Figure 4.13b, the corresponding MOS are marginally, but steadily increasing with larger QPs at OD08 under voxel-based rendering. On the contrary, based on Figure 4.13a, using the splat-based counterpart leads to a substantial upgrade of the quality scores for *amphoriskos* and *romanoillamp* with color improvements at OD08, while subjective ratings for human figures remain very low. This results suggests that, the level of visual artifacts that are introduced with very sparse models from the splat-based renderer, are model-dependent.

Comparison of rendering schemes

In Table 4.4 and Figure 4.18, the performance indexes and scatter plots comparing the MOS obtained from the splat-based against the MOS from the voxel-based experiments are provided. In both cases, no-fitting, linear and cubic fitting functions are enabled. Note that in the scatter plots, the horizontal and vertical bars indicate CIs as computed by the scores obtained from the experiment depicted in the corresponding label.

It is noteworthy that the CIs obtained from the voxel-based experiment are 19.25% larger with respect to the splat-based counterpart, indicating a higher uncertainty regarding the quality score of the testing stimuli. Yet, our results show a strong correlation between the ratings collected from both experiments. In particular, the linear fitting function achieves an angle of 44.3° in Figure 4.18a and 44.7° in Figure 4.18b, with corresponding intercepts of 0.97 and 1.08,

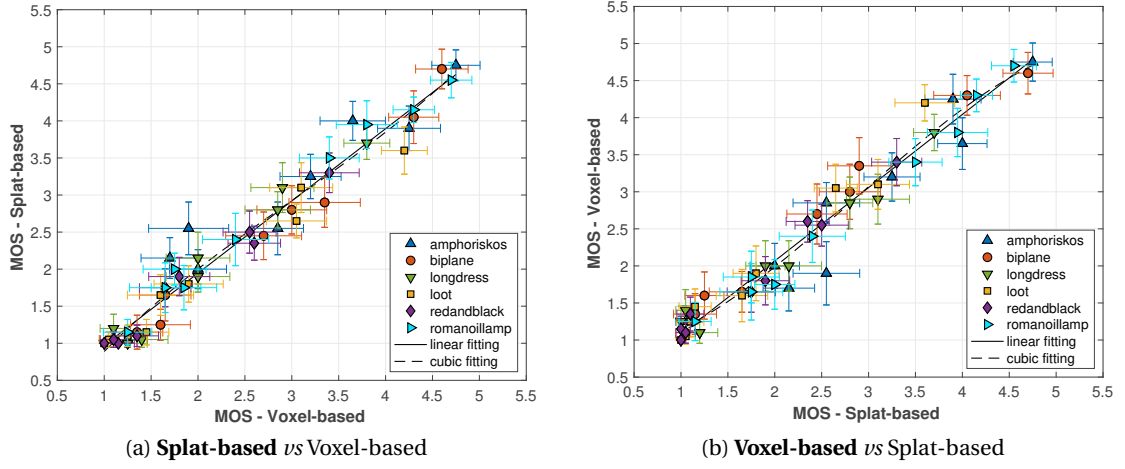


Figure 4.18 – Comparison of subjective scores obtained under both rendering solutions (Bold text represents the ground truth).

Table 4.4 – Performance indexes to compare the rendering solutions (Bold text represents the ground truth).

Splat-based vs Voxel-based											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.981	0.979	0.231	0.315	100%	0%	0%	87.49%	0.00%	5.38%	7.13%
Linear fitting	0.981	0.979	0.220	0.315	100%	0%	0%	87.35%	0.00%	4.47%	8.18%
Cubic fitting	0.982	0.979	0.216	0.315	100%	0%	0%	87.84%	0.00%	5.31%	6.85%
Voxel-based vs Splat-based											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.981	0.979	0.231	0.241	100%	0%	0%	87.49%	0%	7.13%	5.38%
Linear fitting	0.981	0.979	0.221	0.185	100%	0%	0%	87.49%	0%	7.13%	5.38%
Cubic fitting	0.983	0.979	0.213	0.222	100%	0%	0%	87.91%	0%	5.38%	6.71%

thus, confirming the high linear relationship between the score distributions observed in the scatter plots. The high PLCC values under any fitting function provide further evidence on the matter. The relatively low RMSE and OR indicate good accuracy and consistency, while the high SROCC index shows that both experiments agree on the ranking of the testing stimuli in terms of visual quality. Furthermore, a CE of 100% indicates no statistically significant difference between the MOS obtained from the two experiments. The FD and FT percentages indicate that there are cases where one experiment might differentiate the quality level of a particular stimulus over another, while the other would not. However, the 0% for the FR index indicates no false ranking, which is the most offensive type of error.

Significance differences matrices computed in the rating population of each experiment, show statistical differences on how color improvements are rated under the lowest geometric quality between the two rendering solutions. In principle, though, based on the results of the above

analysis, we may conclude that the two experiments are well-correlated, leading to similar conclusions regarding the quality of the stimuli under assessment, despite the different nature of visual artifacts occurring from the two rendering schemes. A Mann-Whitney U-test was computed on the scores to evaluate the difference in the perception of splat-based rendering with respect to the voxel-based counterpart, to validate our findings. No significant effect was found ($Z = -0.9935$, $p = 0.3205$, $r = 0.0214$).

4.3 Conclusions

In this chapter we explore and compare different point-based rendering schemes that were developed for displaying colored point cloud contents. In a first experiment, we compare different 2D and 3D geometric objects, namely disks, cubes and spheres, as point rendering primitives, under various point clouds of diverse characteristics. The obtained content representations were shown side-by-side to human subjects, which were asked to submit their preference. Our results suggest that the 3D variants were in principle preferred as rendering primitive elements in our testbed, with spheric splats outperforming the alternatives. This can be explained by the better preservation of details, and the elimination of flickering artifacts from splat size fluctuations as the camera is rotating, due to the shape of this object. Moreover, the orientation of 3D splats doesn't depend on normal vectors, thus, potential hollow regions due to mis-orientations are avoided. The cube splats were found to be the second best option over all contents, according to human opinions. However, further analysis on the matter shows that the type of represented model, its intrinsic geometric characteristics and color details, might affect this ranking. In particular, it was found that in case of contents of type "Human", cubes were found to be preferred over disks, whereas for models of type "Objects", disks were rated higher. On the contrary, the sparsity level, which accounts for the second influencing factor under examination, showed no impact in the opinion of subjects, with cubes always denoting the second preferred choice.

In a second experiment, we employ and compare two point-based rendering implementations for purposes of subjective quality evaluation. The so-called splat-based renderer is based on representing points via cubic primitive elements of adaptive size across a model, and denotes a compromised solution between quality and complexity according to the results of the first experiment. The second alternative, namely voxel-based renderer, relies on real-time voxelization of the displayed models, which are orthographically projected onto a 2D image grid. The types of artifacts that occur from these two rendering approaches are different. In particular, larger splat sizes are obtained as the topology of a point cloud becomes sparser in the former, whereas in the latter, missing information in the form of pixels is perceived. To examine whether the two renderers lead to the same conclusions regarding the visual quality of the same point clouds, two subjective quality experiments were conducted with identical design and equipment. To this aim, a state-of-the-art codec was employed and several geometric and color quality levels were combined to account for a wide range of degradation levels. Despite statistical differences in the subjective scores that were bounded

to the lowest-geometric quality models, our analysis reveals high correlation between the two rendering approaches. Our results show that the adopted point-based schemes, although simple, they are consistent in perceptual quality evaluation of the compressed point clouds. Yet, our study comes with a limitation, which stems from the usage of a single codec, thus, implying the need for further experimentation in order to draw generalizable conclusions regarding the statistical equivalence of the two rendering methods. Pertaining the performance of the encoder, our results show that higher bit-rate doesn't necessarily grant better visual quality. Specifically, it was shown that a combination of mid-range geometry and high color quality levels is statistically preferred over high geometry with low color quality levels, with the latter consistently demanding higher bit-rates. Moreover, subjective opinions indicate that low geometric resolutions govern the visual appearance, with marginal improvements brought by improving the quality of color.

5 Exploring immersive technologies

The remarkable advances of extended/cross technologies in recent years have resulted in a greater demand for richer imaging modalities that better approximate real-world sceneries. VR systems, in particular, aim at providing immersive experiences that stimulate human senses and increase the engagement of the user with the displayed imagery. In relevant applications, high-quality content is essential for enhancing the realism and the sense of presence. When real-time communication with 3D imaging is targeted (e.g., tele-presence), additional requirements are imposed related to the efficiency and flexibility for capturing, compressing and displaying of the visual data. In this context, point clouds have emerged as an attractive option by providing the possibility of adjusting the visual quality of a model per point, eliminating any dependency imposed by connectivity information from acquisition to rendering. Yet, there is a limited number of studies in the literature addressing challenges that are related to the perception of quality for such content representations in immersive environments, despite being considered as one of the main use-cases.

In this chapter we explore the potential of VR technology as the means to consume point cloud contents in 6DoF immersive inspection scenarios. Our initial efforts are focused on subjective quality evaluation experiments that are conducted in a virtual scene, which is carefully designed for this purpose. Evaluation protocols that are extended to account for interactivity are proposed and adopted in order to enable consistent comparison between a reference and the queried stimulus. High-quality models are selected and shown to human subjects, following a point-based rendering scheme that is configured for visually pleasing results. Moreover, interactivity patterns are extracted from the recorded camera positions in order to provide further insights regarding the user behavior. To better understand how people consume the models and what regions draw their attention, we proceed by conducting an eye-tracking experiment by integrating the necessary equipment in the same set-up. As previously, we aim at promoting interactivity between the user and the content, thus, the same non-distracting scene is employed. Moreover, a task is assigned to every participant to further motivate their engagement, and allowing us to receive feedback regarding properties that are preferred and considered as important for their experience. The recorded gaze and

head cues are processed to obtain importance weights that are associated to the points of each model, essentially, reflecting fixation density maps. To this aim, an heuristic algorithm is developed in order to exploit high-quality gaze measurements, and an ad-hoc scheme is proposed to determine areas of fixations in point cloud representations. Finally, we design and develop an application paradigm for behavioral recording and analysis in a virtual world that better simulates real-life experiences. In particular, a virtual museum is constructed, which is not restricted to a single scene, rather it is extended to several rooms with cultural heritage models exposition that can be visited by the user at will. In this application, both head- and gaze-related data streams can be recorded from the corresponding hardware, while auxiliary modules are integrated in order to enable better synchronization between the two streams and provide the means to benchmark the eye-tracking device.

This chapter is based on material that has been published in (Alexiou et al., 2019b, 2020b).

5.1 Subjective quality evaluation in virtual reality

In previous chapters, we have proposed and analysed several subjective quality evaluation methodologies for point clouds, under a wide range of display devices and rendering approaches, for both colorless and colored models. The experimentation was carried out mainly in desktop set-ups, while also an AR inspection scenario was deployed for geometry-only contents. The results have led to useful insights for best practices and conclusions regarding the influence of several factors that might affect scoring distributions, providing a foundation for subjective quality assessment for this visual data representation.

The objective of this study is to extend previous efforts by proposing the use of a 6DoF immersive experience in a VR environment to rate the visual quality of point cloud contents. VR applications not only enable interactivity and immersiveness, but they also allow the establishment of identical viewing conditions in fully-controlled environments, and facilitate reproducible research. For the purposes of this experiment, a set of high-quality textured point clouds was generated, forming the so-called *PointXR dataset* (Alexiou and Ebrahimi, 2020). Models from this data set are recruited to serve as the reference contents for our evaluation study. Color attributes are encoded using both color encoding modules that are integrated in the the state-of-the-art MPEG G-PCC (MPEG 3DG, 2019) test model, choosing encoding configurations that permit a fair comparison. To display the models, a point-based rendering approach with adaptive splat size and shader interpolation is enabled in order to eliminate surface discontinuities. Moreover, several environmental conditions are adjusted to avoid distractions and enhance realism, while an intuitive controlling system is deployed to improve interactivity means between the user and the content. Anticipating the particularities of 3D modelling display in such environments, we adopt suitable protocols that are adjusted to our needs. For instance, in a virtual scene, human figures would be displayed at the real-world size, in order to enhance the realism. In this example, if a simultaneous double-stimulus comparison was selected, close inspection would have been problematic due to the fact that the



Figure 5.1 – Frontal view of the models.

user cannot obtain a side-by-side view of the same angle for both models at his/her viewport. This can only be attained from distant inspection, where details might not be perceived. Thus, to evaluate the quality of the compressed stimuli in our set-up, we adopt the “sequential DSIS”, and a proposed variant suitable for evaluation of near-lossless compression, namely, “alternating DSIS”, both adjusted to accommodate interactivity. Finally, we analyse the behavior of the users based on interactivity data that was recorded during the subjective evaluations.

5.1.1 Data set

Content selection

A set of five high-quality point clouds was recruited from the *PointXR dataset* that represent cultural heritage models (Alexiou et al., 2020b).

Content preparation

The models were initially voxelized at 10-bit depth (see annex B.2 for implementation details). In particular, their coordinates were quantized, with the output geometry ranging in $[0, 1023]^3$.

The color value of each output voxel was obtained after averaging the color values of the input points that fall in the same voxel. A frontal view of each voxelized model is illustrated in Figure 5.1, indicating the naming that is adopted in the paper and the number of points after voxelization in parenthesis.

Degradation type

The contents were encoded using G-PCC version 8.0 test model. The Octree encoding module was selected to compress the geometry, while both Lifting and RAHT (de Queiroz and Chou, 2016) codecs were used for color compression. To define the configuration parameters for the Octree-plus-Lifting combination, the MPEG Common Test Conditions (MPEG 3DG, 2017) were followed. The degradation levels R02, R03, R04 and R06, annotated as D01-D04 in this study, were selected after expert viewing to represent a range of visual quality levels that spans from very low to very high. For the Octree-plus-RAHT combination, the encoding configurations for geometry remained unaltered. However, the quantization parameter for RAHT (QP) was appropriately adjusted in order to achieve the same bit-rate as Lifting, per model, to secure a fair comparison. This is because although identical QP values were originally used in the CTC for both color codecs, it is evident that Lifting requires more bits than RAHT at the same degradation level. In Table 5.1, the QP values that were used for RAHT (R-QP) and Lifting (L-QP), along with the positionQuantizationScale (PQS) for the Octree encoding module are reported, per degradation level.

Table 5.1 – Encoding configurations per model.

Degradation level	PQS All	L-QP All	R-QP				
			<i>guanyin</i>	<i>muse</i>	<i>roy</i>	<i>shield</i>	<i>tiki</i>
D01 (R02)	0.25	46	41	41	41	41	41
D02 (R03)	0.5	40	36	35	35	36	36
D03 (R04)	0.75	34	31	30	30	31	30
D04 (R06)	0.9375	22	20	19	19	20	19

5.1.2 Methodology

Test methods

In this study, two evaluation protocols were employed, both based on DSIS with a 9-grading scale (9: *Imperceptible*, 7: *Perceptible, but not annoying*, 5: *Slightly annoying*, 3: *Annoying*, 1: *Very annoying*). The first protocol is the sequential DSIS, where the reference model is initially presented to the subjects, followed by the distorted model. The second protocol is the alternating DSIS, where the subjects are allowed to toggle between the reference and the distorted model at will. In the first variant, the reference visual quality of a model is presented to the users, which are subsequently asked to provide a score for the distorted

5.1. Subjective quality evaluation in virtual reality

version displayed next. Hence, temporal masking naturally takes place. In the second variant, the users can visit the reference model at any point they decide. Thus, the scores might capture more accurately relative differences between the models.

In both experiments, the users were able to interact with the stimuli under evaluation with 6DoF in the VR environment. In particular, the subjects were able to navigate both physically in the real world and by teleporting to the position of their preference in the virtual room (i.e., locomotion) using VR controllers. To avoid additional test parameters, no manipulation of the models (e.g., drag, re-size) was allowed. Finally, no time limitations were imposed.

Rendering

The reference models were loaded in the *Rendering* scene of the *PointXR toolbox* described in annex D.4 in order to adjust visualization-related parameters. After experimentation, it was decided to render the models using adaptive point size based on 3 nearest neighbors, and enable the shader interpolation mode. Both quad and disk shaders were evaluated, with the latter bringing no visual enhancements under the aforementioned configuration. On the contrary, the rendering performance was improving in terms of frame rate (i.e., fps), when using the quad shader. Thus, the latter option was selected. A global point scaling value was adjusted per model after expert viewing in order to eliminate hollow regions while achieving the highest possible fidelity. Finally, the models were scaled at a natural size. For smaller objects, a stage of proportional dimensions was placed in the room, and the models were arranged on top for comfort viewing. In Table 5.2, the rendering configurations are summarized per model. Notice that the same settings were employed for the corresponding encoded versions.

Table 5.2 – Rendering configurations per model.

	<i>guanyin</i>	<i>muse</i>	<i>roy</i>	<i>shield</i>	<i>tiki</i>
shader	Quad	Quad	Quad	Quad	Quad
shaderInterpolation	Yes	Yes	Yes	Yes	Yes
adaptivePoint	Yes	Yes	Yes	Yes	Yes
pointScalingFactor	0.6	0.65	0.7	0.65	0.6
modelScalingFactor	0.002	0.0018	0.002	0.0014	0.002

Testing environment

The test was conducted in a controlled physical room of size 3×3 meters. The HTC VIVE Pro headset was used to consume the models in VR with a resolution of 2880×1600 pixels, a field of view of 110° , and a frame rate of 90 Hz. The VIVE base stations were installed in the room to track the position of the user and reflect the corresponding position in the virtual environment.



Figure 5.2 – Virtual environment.

The virtual environment designed for the test consisted of a non-distracting room with parallelepiped shape of dimensions $9 \times 9 \times 5$ (virtual unit meters) and mid-grey walls of low reflectivity. Each model was positioned in the center of the room, and a point light source was placed right above at a height of 3. The pre-computed real-time global illumination option was selected, and shadows of the model were visible on the floor of the room to enhance the realism and the sense of presence. A sign, clearly indicating whether a reference or a distorted model was inspected at every time instance, was placed on the floor in front of the model. An example of the virtual environment is depicted in Figure 5.2. Moreover, in Figure 5.3, instances of a subject interacting with the virtual world are illustrated.

Experimental design

The sequential DSIS experiment chronologically preceded the alternating DSIS counterpart. Both experiments were split in a training and testing stage. In the former, the subjects were able to get acquainted with the virtual environment, the navigation controls, and the evaluation protocol, as well as with representative types of visual distortions they would assess. In the latter, the queried stimuli were evaluated.

The stimuli were displayed in a random order per subject and protocol, while avoiding consecutive evaluations of the same content. At the beginning of each evaluation, the position of the subject was randomly selected in the room to motivate interactivity and inspection from various viewpoints. The loading of the models in the virtual room was performed at the beginning of each evaluation step. Then, depending on the users actions, one model would be visible and the other hidden. Provided that prefabricated objects were employed in run-time,

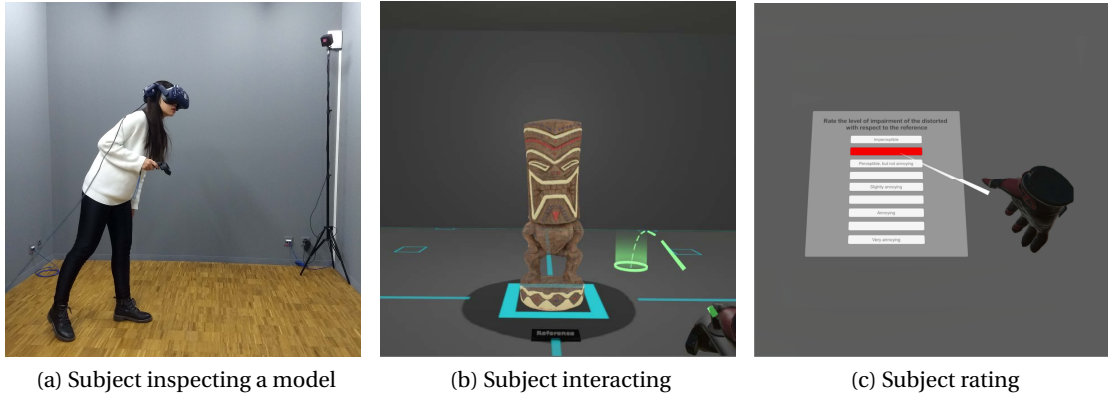


Figure 5.3 – Experimental set-up.

as exported from the *Rendering* scene of the *PointXR toolbox*, no delays were perceived when moving to the next evaluation step. Moreover, the alternation between the models could be performed instantaneously. In order to avoid flickering that could be sensed even in cases of negligible differences, a delay in the presentation of the next model was intentionally imposed. In particular, a delay of 1 sec was used for the sequential DSIS, and a delay of 0.25 sec was used for the alternating DSIS. For the second case, we allow a faster response to avoid making the delay a factor that prevents subjects from switching between models.

For each of the 5 point clouds, there were 4 degradation levels obtained from 2 color codecs, leading to a total of 40 stimuli that were assessed. For each user, an extra evaluation step was added at the beginning of each experiment, in order to ensure that the subject was familiar with the task at hand. The obtained scores from this step were later discarded. A total of 24 subjects participated in the experiments, with 20 subjects evaluating the models in each protocol. For individuals who participated in both sessions, a 2-days rest period was imposed in between to avoid temporal bias. The subjects population consisted of 15 males and 9 females, with an average age of 26.4 (min 19, max 33).

Data processing

The MOS and CIs were computed to characterize the quality level and uncertainty of a particular stimulus. Moreover, to compare the test methods, performance indexes described in annex A.2 were employed.

5.1.3 Results

Subjective results

Subjective quality scores for the point cloud models were obtained from both experiments. Outlier detection based on the ITU-R Recommendation BT.500-13 (ITU-R BT.500-13, 2012)

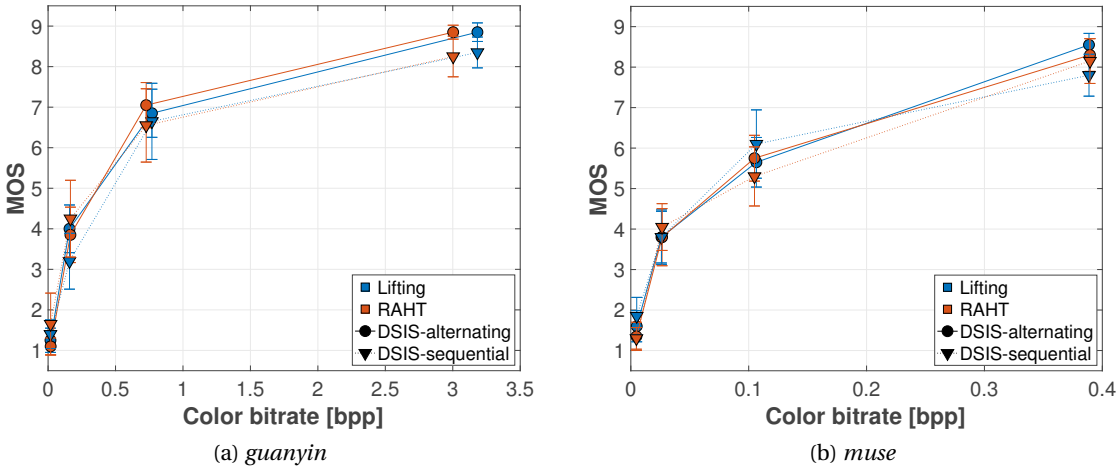


Figure 5.4 – Subjective scores against color bit-rates from both codecs, using both test methods.

was applied on each test, separately. No subject with deviating scoring behavior was identified and, thus, each individual rating was accounted to compute the MOS and the 95% confidence intervals assuming a Student's t-distribution. Moreover, the position and viewing angle of the users were recorded in real-time at the rendering frequency of 90 Hz, in order to analyse their behavior in VR.

In Figure 5.4, we present graphs for two of the models used in the experiments, indicating the MOS as a function of the color bit-rates, for both codecs and evaluation protocols. The bit-rate is presented in bits-per-(input)-point, which denotes the ratio of the total number of bits divided by the number of input points. It can be seen that the MOS is improving as the bit-rate is increasing, while the levels of visual quality for each model spans the entire scoring space. As expected, the range of color bpp varies per model; that is, models with narrow color distribution, such as *muse*, require bit-rates as low as 0.4 bpp to achieve transparency, whereas models with high color variability, such as *guanyin*, need higher bit-rates. Very similar scoring trends are obtained for the rest of the models.

Based on Figure 5.4, the performance of the color encoders is equivalent. To validate this observation, we compare the two codecs across all contents under both test methods. In particular, in Figure 5.5 and Table 5.3, we provide the scatter plot and the performance indexes after comparing the subjective quality scores for the same degradation levels of RAHT against Lifting, which is set as the ground truth, using both protocols. We observe that when using the sequential variant the correlation slightly worsens. However, in both cases it remains very high, indicating that there is no preference of the subjects for one codec over the other. A non-parametric one-way ANOVA applied on the scores obtained from both evaluation protocols separately, per color encoder, results in p -values of 0.6927 and 0.809 for the alternating and the sequential DSIS, respectively, which confirms that the color codecs are statistically equivalent in both cases.

5.1. Subjective quality evaluation in virtual reality

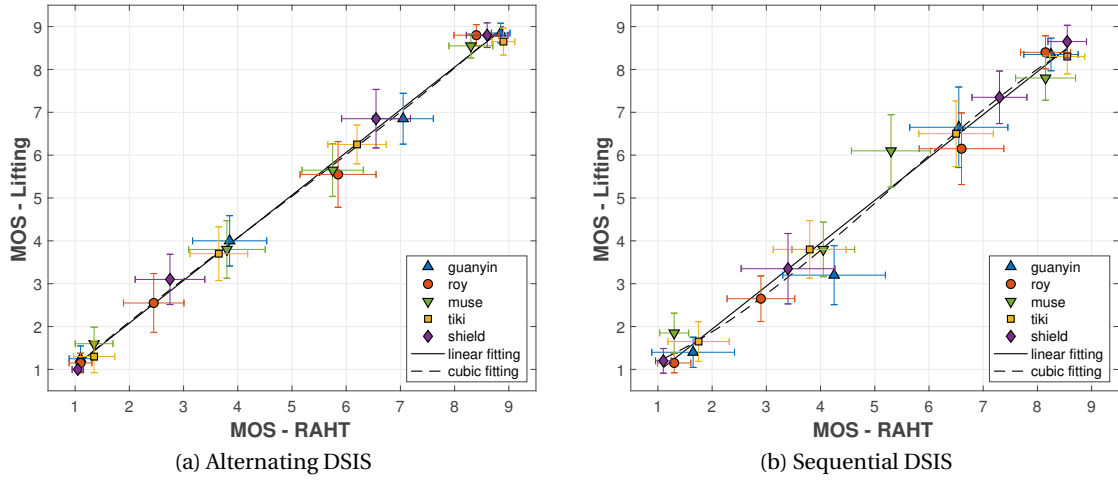


Figure 5.5 – Comparison of subjective scores between RAHT and Lifting (ground truth) color codec.

Table 5.3 – Performance indexes to compare RAHT against Lifting (ground truth) color codec.

Alternating DSIS											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.998	0.988	0.206	0.100	100%	0%	0%	97.37%	0%	1.05%	1.58%
Linear fitting	0.998	0.988	0.194	0.100	100%	0%	0%	97.37%	0%	1.05%	1.58%
Cubic fitting	0.998	0.988	0.190	0.150	100%	0%	0%	97.89%	0%	1.05%	1.05%
Sequential DSIS											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.991	0.968	0.381	0.100	100%	0%	0%	92.63%	0%	4.74%	2.63%
Linear fitting	0.991	0.968	0.377	0.150	100%	0%	0%	92.63%	0%	4.74%	2.63%
Cubic fitting	0.992	0.968	0.358	0.150	100%	0%	0%	93.16%	0%	3.16%	3.68%

Comparison of test methods

Regarding the comparison of the two DSIS protocols that were employed in our experiments, a scatter plot and the performance indexes that indicate the correlation between the corresponding score distributions are depicted in Figure 5.6 and Table 5.4. In principle, we observe strong correlation between the two test methods, confirmed by all performance indexes that were computed, independently of the employed regression model. As the data was not normally distributed according to the Shapiro-Wilk normality test ($W = 0.88$, $p < .001$), we test statistical significance between the evaluation protocols through a non-parametric Wilcoxon rank-sum test, and no significance was found ($Z = -0.40$, $p = 0.689$, $r = 0.01$). The linear fitting function achieves an angle of 46.5° and 42.99° with an intercept of -0.33 and 0.39 in Figures 5.6a and 5.6b, respectively, indicating a general tendency to rate slightly higher the low-quality model and lower the high-quality models in the alternating counterpart. This

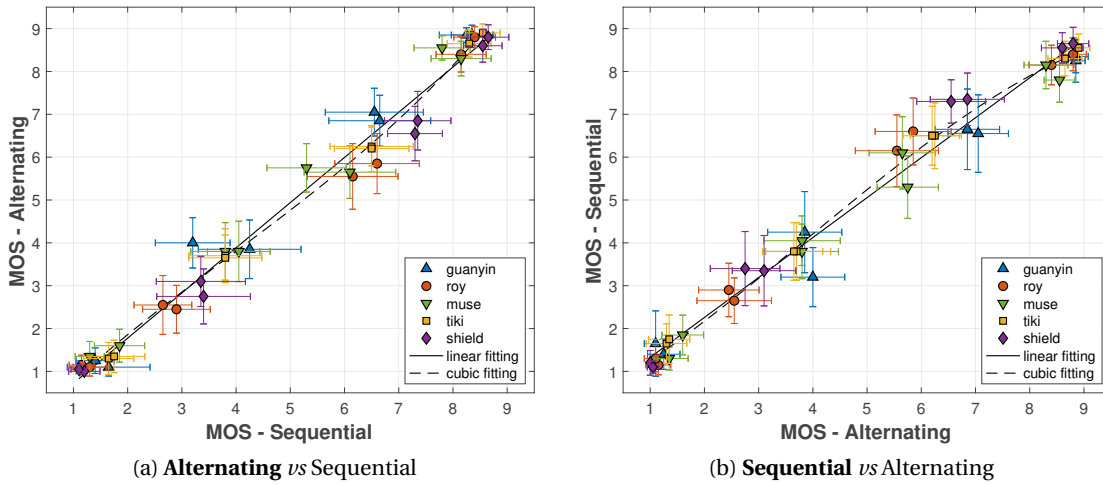


Figure 5.6 – Comparison of subjective scores obtained under both DSIS variants (Bold text represents the ground truth).

Table 5.4 – Performance indexes to compare the DSIS variants (Bold text represents the ground truth).

Alternating <i>vs</i> Sequential											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.991	0.979	0.412	0.325	100%	0%	0%	93.72%	0%	2.05%	4.23%
Linear fitting	0.991	0.979	0.379	0.300	100%	0%	0%	93.72%	0%	2.56%	3.72%
Cubic fitting	0.993	0.979	0.353	0.225	100%	0%	0%	93.59%	0%	2.56%	3.85%
Sequential <i>vs</i> Alternating											
	PLCC	SROCC	RMSE	OR	CE	UE	OE	CD	FR	FD	FT
No fitting	0.991	0.979	0.412	0.175	100%	0%	0%	93.72%	0%	4.23%	2.05%
Linear fitting	0.991	0.979	0.357	0.150	100%	0%	0%	93.72%	0%	3.72%	2.56%
Cubic fitting	0.993	0.979	0.321	0.100	100%	0%	0%	93.46%	0%	2.56%	3.97%

can be explained by the fact that subjects were having access to the distorted model upon demand, thus, it was easier to spot and penalize small quality deviations. Another notable outcome is the fact that the confidence intervals in the sequential protocol were found to be by 27% larger with respect to the alternating, showing that the latter approach leads to smaller rating variability and higher consistency. However, the learning effect on the stimuli should be accounted in this result, considering that several subjects participated in both sessions.

A post-questionnaire that was filled by the subjects participating in both sessions, shows that the alternating DSIS variant is universally preferred. The most common key-words that were provided to justify their choice were: “precise”, “no memorization”, and “faster”. The observers, in principle, agreed that the alternating protocol is more effective with high-quality models, while a participant noted that potential biases might be introduced in the sequential protocol,

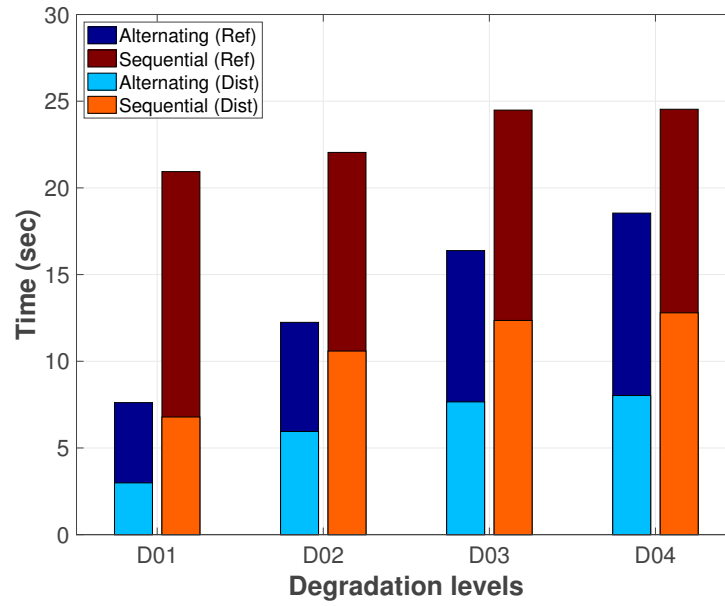


Figure 5.7 – Average time of interaction.

as the subjects might grade non-existing distortions. This can especially be problematic in the case of point clouds, where the quality of the non-compressed content might not excel already. Although such trends are not visible in our results, future studies might benefit from such remarks.

Regarding the characteristics of the population of the test, the majority of the users were naive. Specifically, 6 out of 24 subjects were using a headset for the *first time*, and 15 had tried *some times*. The participants were evidently satisfied with the level of immersion, the visual quality of the contents and the total quality of experience, with an average score of 4.4, 4.3 and 4.3 out of 5, respectively (5: *Excellent*, 4: *Good*, 3: *Fair*, 2: *Poor*, 1: *Bad*). A total of 5 subjects felt mild discomfort, whereas the rest reported that they didn't face any symptom. Finally, 3 subjects suggested that a headset without cables would have enhanced their experience.

The characterization “faster” given by the subjects for the alternating DSIS, is confirmed by the average time the users spent per stimulus. In particular, an analysis of the logged interactivity information reveals that, for the alternating case, a subject needed on average 13.7 ± 7.3 sec (6.2 on Distorted and 7.5 on Reference), whereas 23 ± 15.2 sec (12.4 on Reference and 10.6 on Distorted) were spent in the sequential counterpart. No particular trends are identified when the average interaction time is clustered per model or per codec. However, clear trends are observed for different degradation levels, as can be shown in Figure 5.7. In particular, it is obvious that as the quality level is increasing, more time is allocated from the subjects on the distorted models for both scenarios. Moreover, for the alternating variant, the total time of interaction decreases as the degradation increases. This is due to the fact that during the sequential protocol, the user is unaware of the quality level of the distorted model. Thus, the amount of time that will be spent in the reference model is independent from the distorted

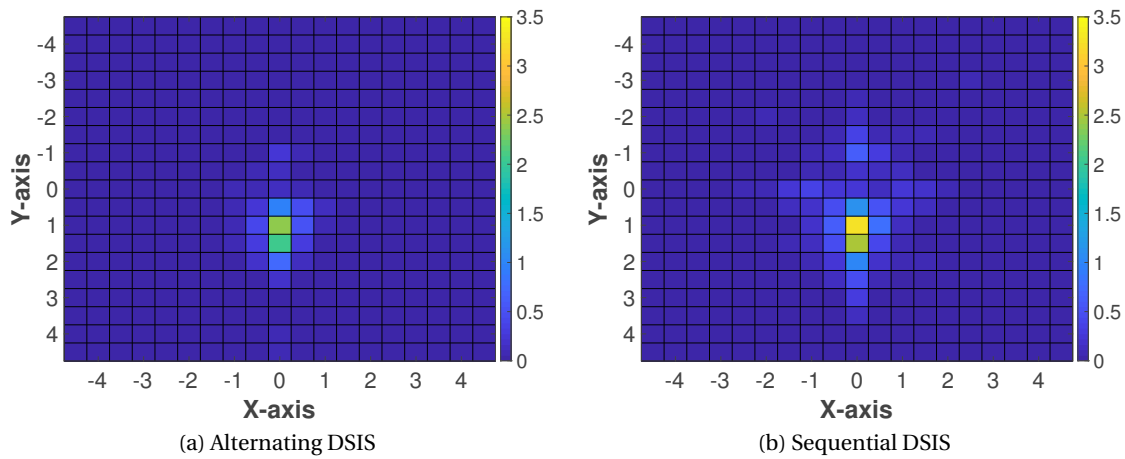


Figure 5.8 – Interactivity patterns per evaluation protocol.

version. However, in the case of the alternating protocol, the user can switch between the two models at will. In such a case, if the distorted model is of low quality, a low score will be given without further inspection. On the contrary, in case of a high-quality distorted model, more time will be needed before providing a score.

This explanation can be confirmed by the number of times the subjects inspected the reference model before providing a score during the DSIS alternating experiment. In particular, by grouping the number of re-visiting the reference model per quality level, an average of 1.4, 2.5, 3.9 and 4.5 is observed for D01, D02, D03 and D04, respectively. Note that the results show no particular tendencies when we compute the average per content (ranges between 2.9 and 3.2), or per codec (average of 3.1 for both Lifting and RAHT).

In Figure 5.8, illustrations of interactivity patterns are provided for each experiment. The virtual test room is represented by a grid, with each square corresponding to a surface of 0.5×0.5 virtual unit meters. The time spent at each position of the room, averaged across stimuli and subjects, is color coded, with brighter values indicating larger time intervals. The colorbar, on the right of each figure, indicates the range and is measured in sec. From these interactivity patterns, it can be seen that the users were more static in the alternating variant. In both cases, the participants spent most of their time in close frontal views, despite the fact that the starting position was randomized. However, in the case of the sequential protocol, more perspectives and distances were tested.

5.2 Visual attention in virtual reality

Visual saliency is a fundamental topic that studies the identification of the regions of a scene that draw the attention of observers, by exploring the mechanisms of human visual system. Models that predict human visual attention are attractive in computer vision and signal

processing communities, and have been proposed for radically different types of imaging. For 3D visual information, saliency detection is an active and largely open area of research the last decade. In particular, several algorithms are reported for predicting salient regions, based on mesh and point cloud data. Well-known mesh-based saliency schemes depend on center-surround filters with Gaussian-weighted curvatures (Lee et al., 2005), shape matching algorithms (Shilane and Funkhouser, 2007), per vertex distinctness with shape extremities and patch association (Leifman et al., 2012), local contrast and global rarity (Wu et al., 2013), spectral attributes (Song et al., 2014), and relations to unsalient regions of the content (Tao et al., 2015). Point cloud-based models rely on regional contrast using the Fast Point Feature Histogram as local shape descriptors to identify distinctness per point (Shtrom et al., 2013) or cluster (Tasse et al., 2015), comparison of local surface properties at different scales and distances (Akman and Jonker, 2010), or covariance descriptors (Guo et al., 2018). Interested readers can refer to (Liu et al., 2016) for a detailed survey.

Visual saliency models are typically validated using as ground-truth fixation density maps that are collected from eye-tracking experiments. In the case of 3D imaging, a limited number of studies has been recorded. Howlett et al. (Howlett et al., 2005) conducted an eye-tracking experiment on mesh simplification algorithms. The subjects were able to examine the degraded models from different viewports through rotation using key arrows. In (Kim et al., 2010), the performance of (Lee et al., 2005) was assessed using gaze data that were obtained after inspection of projected images from meshes. Wang et al. (Wang et al., 2016) performed an experiment with 3D printed figures. This work was recently extended to account for different viewing positions and model construction materials (Wang et al., 2018b). The collected eye-tracking data were mapped onto the 3D meshes to form fixation maps. Lavoué et al. (Lavoué et al., 2018) carried an eye-tracking campaign with animated videos of 3D meshes. Several influencing factors were considered, such as model shape, camera position, material, and illumination. It is noteworthy that the two most popular data sets that serve as ground truth, are obtained from experiments where subjects were asked to manually select points that are “interesting, or defining” (Dutagaci et al., 2012), or “likely to be selected by other people” (Chen et al., 2012).

Although the experimental settings that are typically employed in the aforementioned studies provide accurate gaze measurements in highly controlled set-ups, they lead to rather unnatural ways of consumption, with limited or non-existent user engagement. Furthermore, despite the current availability of dedicated VR platforms, the influence of visualizing 3D models in immersive experiences hasn’t been explored yet. Visual saliency of VR contents has been investigated in the form of omnidirectional image and video sequences using head-mounted displays. Specifically, several models (Bogdanova et al., 2008; Maugey et al., 2017), testbeds (Upénik et al., 2016; Abreu et al., 2017) and data sets (Corbillon et al., 2017; Rai et al., 2017; David et al., 2018; Knorr et al., 2018) have been proposed for gaze- and/or head-tracking data. In (Sitzmann et al., 2018), the authors performed a thorough analysis on gaze and head data collected from extensive experimentation using static omnidirectional panoramas on several testing set-ups. The collected gaze and head orientations from the participants were



Figure 5.9 – Selected contents for the experiment.

thoroughly analysed, showing that head trajectories are sufficient to predict saliency in a VR setting.

In this study we extend the state-of-the-art by tracking the visual attention of observers in an immersive VR experience with 6DoF. We design a virtual reality scene that offers high levels of realism with limited distractions. A wide range of static point cloud models is inspected by human subjects, while their gaze is captured in real-time. To motivate user exploration, a task-dependent protocol is adopted. The recorded visual attention information is used to extract fixation density maps. To obtain high quality fixation points, an heuristic algorithm is developed that utilizes every recorded gaze measurement from the two eye-cameras that are installed in our set-up. The fixation density maps can be interpreted as importance weights that are associated to the points of the model. Using a simple color mapping technique the latter are transformed to heat maps in order to visually display regions of higher interest for the models of our data set.

Table 5.5 – Point cloud contents characterization.

Content	Type	Voxelized	Voxel depth	Scaling	Points
<i>amphoriskos</i>	Object	✗	10	$9.78 \cdot 10^{-4}$	814,474
<i>biplane</i>	Object	✗	10	$39.10 \cdot 10^{-4}$	1,181,016
<i>egyptian_mask</i>	Object	✓	12	$2.44 \cdot 10^{-4}$	272,684
<i>longdress</i>	Human	✓	10	$18.71 \cdot 10^{-4}$	857,966
<i>loot</i>	Human	✓	10	$18.86 \cdot 10^{-4}$	805,285
<i>redandblack</i>	Human	✓	10	$19.15 \cdot 10^{-4}$	757,691
<i>romanoillamp</i>	Object	✗	10	$9.78 \cdot 10^{-4}$	636,127
<i>shiva</i>	Object	✓	12	$2.43 \cdot 10^{-4}$	1,009,132
<i>soldier</i>	Human	✓	10	$19.01 \cdot 10^{-4}$	1,089,091
<i>statue_klimt</i>	Object	✓	12	$4.88 \cdot 10^{-4}$	499,660
<i>the20smaria</i>	Human	✗	10	$18.01 \cdot 10^{-4}$	1,553,937
<i>ulliwegner</i>	Human	✗	12	$4.52 \cdot 10^{-4}$	811,019

5.2.1 Data set

In this study, 12 static point clouds that are naturally consumed outer-wise were selected (6 objects and 6 human figures). The majority of models have been recruited from the JPEG¹ and the MPEG² repositories, which were released for purposes of point cloud compression related activities, with *amphoriskos* retrieved from Sketchfab³. The acquisition technique for each model varies, thus leading to different types of artifacts on their structure and texture. Moreover, some of the models were voxelized by default, whilst for some others, the geometry was spanning in an arbitrary range. Thus, to minimize the impact of geometrical irregularities, the non-voxelized contents were also voxelized (see annex B.2 for implementation details). The voxel depth was selected per model after ensuring high system responsiveness to avoid discomfort of the subjects in the virtual environment. In Figure 5.9, the pristine models are illustrated, while in Table 5.5, additional information regarding the (optional) pre-processing, and the final geometric intrinsic resolutions, are detailed.

5.2.2 Methodology

Apparatus

The virtual environment was designed in Unity, which provides an open source platform with high flexibility for the creation of virtual games and modular structure to facilitate integration of external plug-ins. An HTC VIVE Pro headset was used as a viewport to the virtual world with a resolution of 2880x1600 pixels (1400x1600 per eye, 615 ppi), a field of view of 110°, and a

¹<https://jpeg.org/plenodb/>, last accessed 12/2020

²<http://mpegfs.int-evry.fr/MPEG/PCC/DataSets/pointCloud/CfP/>, last accessed 01/2020

³<https://bit.ly/3nekULm>, last accessed 12/2020

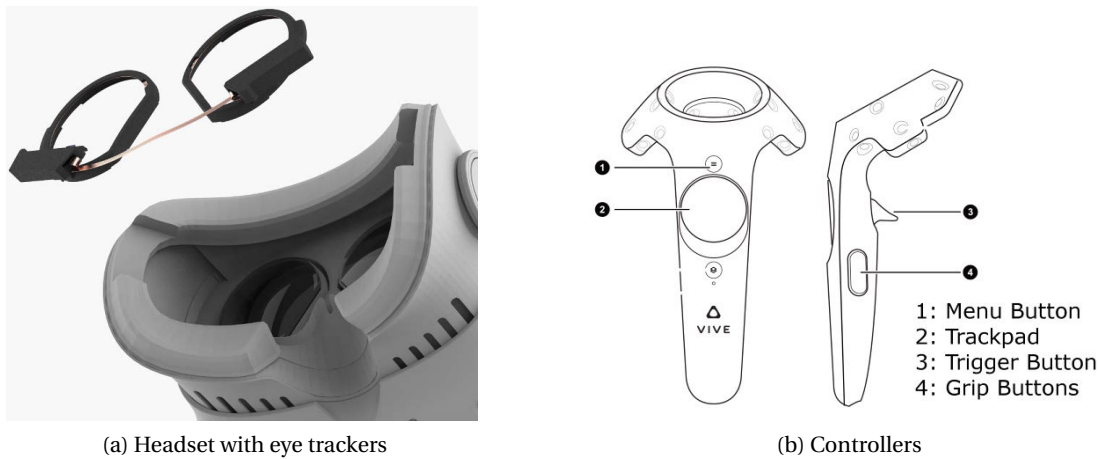


Figure 5.10 – Apparatus of the experiment.

frame rate of 90 Hz. To enable interaction between the user and the virtual world, the VIVE controllers were employed and configured using the Steam VR plug-in. In particular, following the naming of Figure 5.10b, the trackpad button was selected for tele-porting around the virtual world, the grip buttons of the left and right controllers were used to rotate the camera by 45° left-wise and right-wise, respectively, and the trigger button was employed to proceed to the next stage of the evaluation. Moreover, users were able to interact with the virtual world through physical movements; yet, with the above configuration, we ensured that people could limit their physical actions in case they would feel uncomfortable.

To capture gaze data, the Pupil Labs eye tracker (Kassner et al., 2014) was attached to the headset, as shown in Figure 5.10a. This hardware device consists of two eye-cameras that track both eyes independently at a frequency of 120 Hz with an accuracy of 0.60° under ideal conditions. In particular, the eye-cameras record image sequences, and at each frame the pupil positions are detected. The pupil positions are then mapped to gaze points in viewport space. The gaze points together with associated quality values indicating the detection accuracy of the corresponding pupil positions are delivered to Unity, denoting the eye-data stream. The headset position and rotation was tracked by VIVE base stations installed in the physical room, forming the head-data stream. Both eye- and head-data streams were synchronized with the rendering frame rate in Unity, before being exported. Thus, the recorded information corresponds to the effective frames that were visualized by a user.

The detection of pupil positions in the recorded image cues from the eye-cameras is performed by open-source software (i.e., Pupil Capture, Pupil Service) that is coming with the hardware. Relevant scripts provide implementations of gaze mapping, video recording, data streaming and events broadcast, while also additional plug-ins such as blink and fixation detection can be enabled. The Pupil Capture is a higher-level software providing an interface and more features with respect to the Pupil Service counterpart, which is designed for the implementation of lower-level operations. Thus, in our experimentation, the former is employed. The

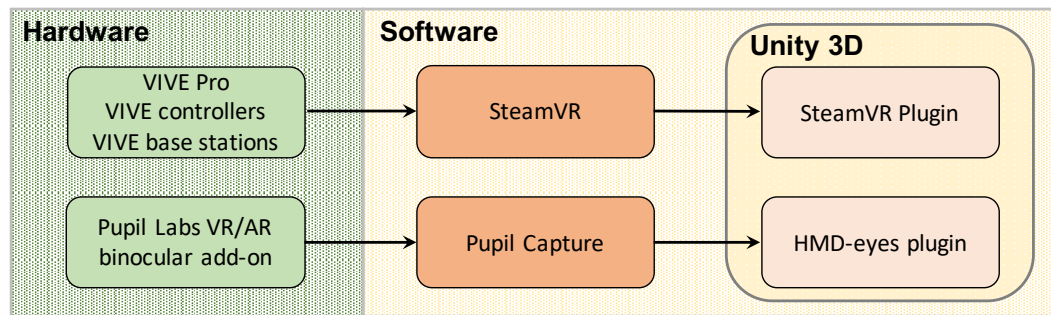
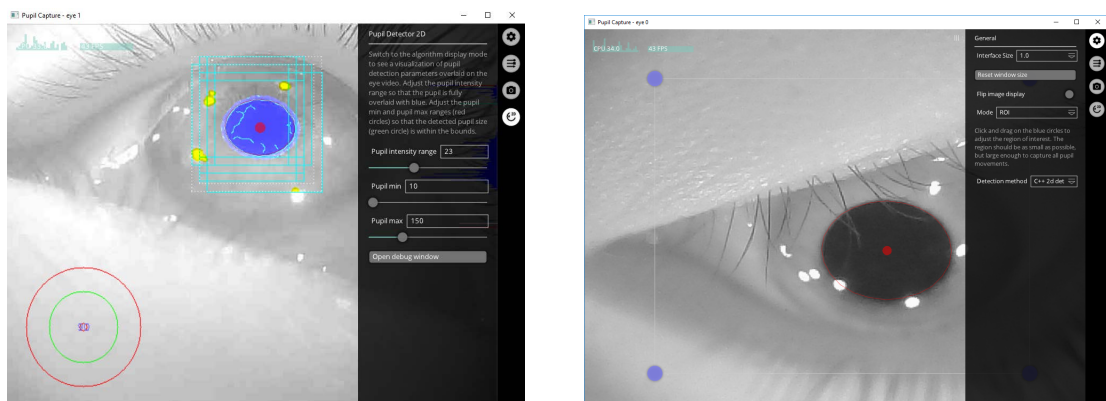


Figure 5.11 – Schematic diagram with the hardware and software modules together with their inter-dependencies.



(a) Algorithm view: the squares show the initial estimations of pupil region, while the red dot in the middle shows the exact pupil position (the higher the opacity, the higher the confidence)

(b) Region of Interest: The square box indicates the region over which the pupil position will be searched

Figure 5.12 – Eye camera window from Pupil Capture software.

communication with Unity is established through the HMD-eyes plug-in, which is developed by Pupil Labs specifically for supporting the integration of the hardware in HTC VIVE and Microsoft Hololens. The HMD-eyes is also responsible for the calibration of the eye-tracker, so that pupil positions can be mapped to gaze points in virtual coordinates inside Unity. A high-level diagram indicating the hardware/software dependencies is provided in Figure 5.11 .

The accuracy of the eye tracker mainly depends on two operations: (a) the detection of pupil positions in recorded images, and (b) the mapping of pupil positions to gaze points in the viewport domain. For the former, there are several configuration parameters that are offered through the Pupil Capture API and can be adjusted individually per eye-camera, such as the focal length, the region of interest (i.e., the part of the image over which the position of the pupil will be searched, as shown in Figure 5.12), the absolute exposure time (which leads to brightness changes in the recorded images), and the resolution of the captured images (the higher the resolution the more data are processed, thus, potentially leading to lower frame rates), among others. Note also that there were two models implemented, namely, 2D

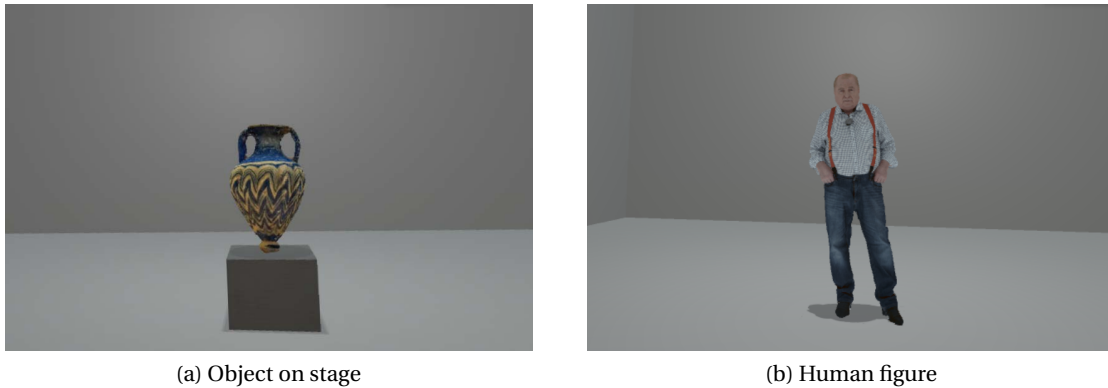


Figure 5.13 – Virtual reality scene. The environment and the illumination are not distracting, while the shadows underneath the models enhance the sense of realism.

and 3D detection mode, with the first being more stable at the time of the experiment. The second operation, which maps the pupil position, requires calibration of the eye-tracking device, which is handled from the HMD-eyes project as mentioned earlier. Several parameters can be configured, such as the number and the positions of the markers, as well as the time duration of their appearance. Despite the fact that the marker color changes during calibration to reflect the quality of the pupil positions detection, which indicates the reliability of the mapping, there is no way to validate the accuracy of the gaze measurements in the virtual world. Moreover, in the case of virtual reality using an HMD, the relative position between the eye and the camera could change during the experiment due to HMD slippages. If this happens, the mapping between pupil and gaze positions will be inaccurate. For these reasons, as will be explained later, an *error profiling* is issued after each evaluation session.

Rendering

Every model was loaded in Unity using the Pcx importer⁴, which converts a point cloud into a mesh-based object. The default renderer provides the options to display a content as a set of raw points, or disks of fixed size. In this experiment, an earlier version of the renderer described in annex D.4 was employed. In particular, the quad shader with adaptive point size based on 5 nearest neighbors was selected, and the shader interpolation (Schütz and Wimmer, 2015) option was enabled, simulating an adaptive screen faced paraboloid that resulted in water-tight surfaces.

Testing environment

The virtual world consisted of a non-distracting room in the shape of a parallelepiped ($10 \times 10 \times 5$ virtual units), with mid-grey walls. The users were able to interact through the VIVE

⁴<https://github.com/keijiro/Pcx>

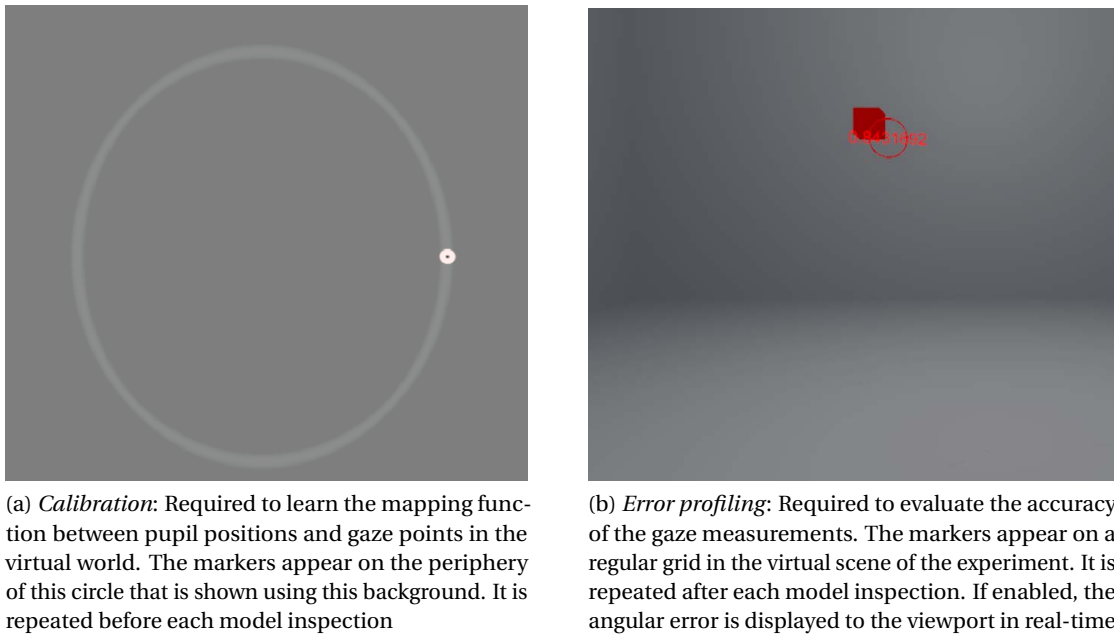


Figure 5.14 – Gaze points calibration and evaluation of measurements.

controllers, or physically navigate in the real world space (3.5×3.5 meters) while also rotate their body and orient their head to capture their preferred view. To promote interaction, the initial position of the users entering in the virtual scene for *inspection* was selected randomly, yet, it was intentionally avoided to appear far away from the models. The models were placed in the middle of the room and were scaled appropriately to simulate realistic sizes. For instance, the height of human figures was set to 1.85. Smaller objects (i.e., *amphoriskos*, *egyptian_mask*, *romanoillamp*, *shiva*) were placed on top of a stage to allow natural viewing. To enhance realism, real-time lighting was applied to the scene using a point light source, while shadows were enabled through a custom script developed by the authors. In particular, by projecting vectors defined from the position of the light and every point of the content, a shadow texture was computed and applied on a quad primitive object, simulating a first order light reflection. Examples of the VR scene with two different contents are illustrated in Figure 5.13.

Experimental design

Before the experiment, visual acuity and color vision of every subject was tested using Snellen and Ishihara charts. Then, each subject was familiarized with the controllers and the naming of each button in order to be able to communicate easier. The inter-pupillary distance was measured and the headset was adjusted by the operator accordingly. A second operator ensured that high quality gaze data were obtained by adjusting configurations in the Pupil Capture API, when needed.

A training phase preceded the actual test. The test was split in two rounds (15-20 minutes

each), with a mandatory 5-minute break in between. During training, the subjects acquainted with the virtual environment and the navigation means on a dummy content. After feeling comfortable with the set-up, they were informed about the task that was assigned to them: “We ask you to examine a set of models; after visualization, we will ask you to order them based on your preference. We will also ask what is the criterion of your preference”. Moreover, they were instructed that it was not necessary to remember any model, as access to corresponding images would be given at the end of a round. To facilitate their task and to identify potential divergence in the criteria of preference, in the first round the set of 6 objects was visualized, while in the second, the 6 human figures were inspected in random order. No time limitations were applied for the training or the actual test.

For every model and each subject, a session was split into three consecutive steps:

1. *Calibration* to (re-)map the pupil positions to gaze points in viewport coordinates. For this purpose, the HMD-eyes software was used with 7 markers appeared on a circle and fixed depth in a 2D calibration mode, as illustrated in Figure 5.14a. The mapping function is estimated using a bi-variate regression model.
2. *Inspection of models* is the step where the participants are consuming the 3D model, while their viewing behavior is recorded.
3. *Error profiling* is issued at the end of each session in order to estimate the accuracy of the gaze measurements due to calibration inaccuracies, or HMD displacements. Assuming a worst-case scenario for HMD slippage at the end of a session, a regular grid of 9 markers at pre-defined positions in the virtual scene are presented to the users, which are asked to stare at them for a certain period of time. In Figure 5.14b, an example of the presentation of one marker is indicatively depicted. The center-top, center-bottom, left and right markers were positioned at $\pm 18.25^\circ$ in the vertical and horizontal axes, respectively, while the visual angle between the middle and corner markers was 25° at a distance of 1 virtual units, with respect to the camera position and orientation. Based on the recorded gaze measurements, the average angular error is computed per marker in an off-line post-processing step. A threshold of 7.5° was used to discard unintentional gazing, and a minimum of 100 samples was required; in case of fewer samples, a marker was classified as *invalid*. This procedure allows us to decide whether the gaze data obtained from a certain session is accurate or not.

A total of 21 subjects (12 males and 9 females) was recruited for this study in a volunteering basis with a min of 20.8, a max of 38.9 and an average of 26.7 years of age.

Data processing

The recorded data consist of left and right gaze positions, estimated after mapping the pupil positions in viewport space, which is normalized and relative to the camera. The middle gaze

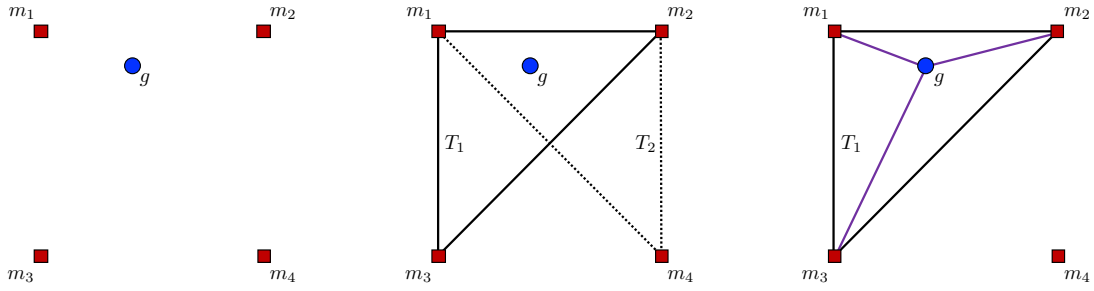


Figure 5.15 – Estimation of angular error for gaze points. On the *left* figure, we present the gaze point g and its four adjacent markers m_i , $i = \{1, 2, 3, 4\}$. We assume that the corresponding angular errors are *valid*. In the *middle* figure, the two candidate surrounding triangles are depicted, indicated as T_1 and T_2 . Given that g is closer to the vertices of T_1 , the barycentric coordinates of g in T_1 are computed and, then, interpolated in order to estimate the angular error of the gaze point g .

position is also obtained as a simple average of the above. Thus, provided the camera position, for every human gaze sample, we get three distinct measurements from left, right and middle gaze positions (*gaze types*) that approximate the actual point of gaze. Instead of selecting only one out of these three measurements, considering that physical movements may affect the accuracy of pupil detection on different regions of the screen due to HMD displacements, we devise an heuristic method to keep gaze positions of lower angular error.

Initially, for every human gaze sample, the quality value assigned to the right and left gaze position is individually assessed. A gaze position is discarded if the quality value is lower than 0.5. If at least one is discarded, the middle gaze position cannot be used. In case both values are 0, the sample is classified as blink. Moreover, a gaze position is discarded if it is outside the range determined by the markers' positions.

After removing low-confidence and out-of-range gaze positions, the angular error of each remaining gaze position is estimated. For this purpose, the data collected from the *error profiling* established after each session are used, where the average angular error at each marker is estimated, for every *gaze type*. For each gaze position, the 4 markers surrounding it are selected as displayed in Figure 5.15. There are two triangles enclosing a point that is lying between four equally spaced vertices. We start from the triangle whose vertices are closest to the gaze position. A barycentric interpolation with weights equal to the corresponding angular errors obtained from the profiling is applied. If there is an *invalid* marker in the first triangle, we proceed to the second. If there is an *invalid* marker in the second triangle too, the gaze position is discarded. Finally, among the remaining gaze positions, the gaze type with the smallest angular error is kept. This is repeated for every human gaze sample to maintain high quality estimations while avoiding discarding useful data.

To identify fixation points, the dispersion-based I-DT algorithm (Salvucci and Goldberg, 2000) is employed with 150 ms minimum duration and 1° maximum dispersion. The window length is adjusted to avoid duplicated fixations. An additional constraint that a fixation can only

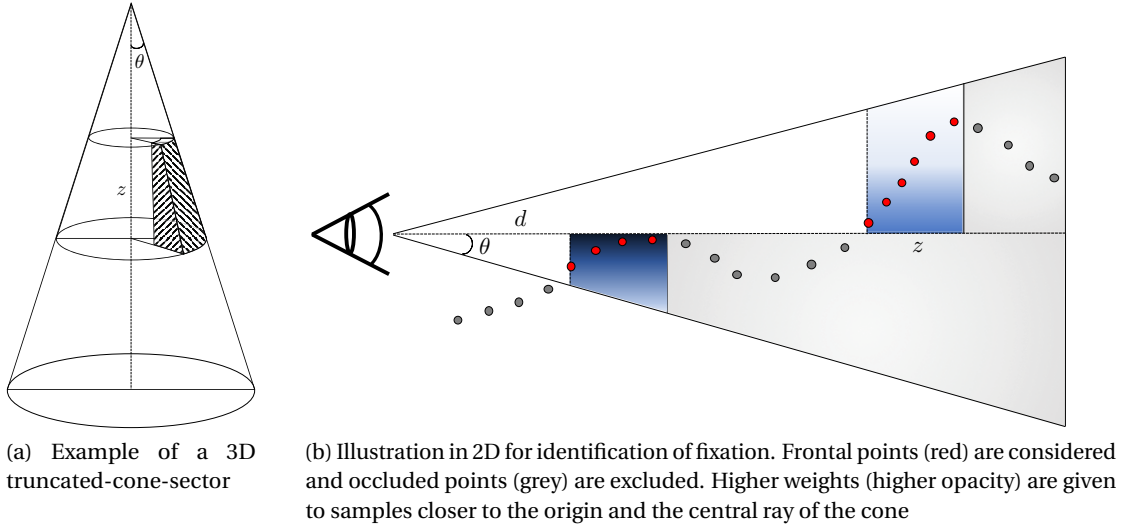


Figure 5.16 – Visual interpretations for key-components employed in the proposed heuristic methodology to identify fixations.

be obtained from consecutive measurements of the same *gaze type* is set, while a minimum number of 4 samples is required for a period of 150 ms.

After a fixation is detected, the average gaze position is estimated over the duration of the fixation. The corresponding average angular error is computed based on barycentric interpolation, similarly to what has been done for individual gaze positions. If the fixation point is out-of-range, or there is no triangle with valid markers, the fixation is discarded. Otherwise, the direction of the fixation is computed as the vector between the average camera and average gaze position in world coordinates, over the duration of the fixation.

A cone is cast towards this direction, and the points of the model that fall inside the cone are collected. Since no colliding can be achieved with points, a set of angles and distances to split the cone into a non-overlapping set of volumes is defined, which we call cone-sectors. Moreover, a threshold of acceptable depth, z , for the points that lie in a cone-sector is determined, leading to a truncated-cone-sector, as shown in Figure 5.16a. The latter is defined by the current cone-sector, the enclosed point that is closest to the origin, and the acceptable depth along the direction of the cone. Once a truncated-cone-sector is identified, the remaining points lying in the same cone-sector are not considered. Thus, frontal points of the model are selected, while points that correspond to occluded regions are discarded. This procedure is repeated for every cone-sector, and the resultant set of points constitutes the fixation.

Finally, the points determined from the procedure above are weighted as follows. Let f be a fixation with angular error θ and duration t . Let x be a point of the fixation, p its distance from the central ray of the cone, and d the distance between the origin and the projection of x

onto the ray. With $\sigma_f = d \cdot \tan(\theta)$, the importance weight of the point x is given as:

$$w(x) = \frac{t}{\sqrt{2\pi\sigma_f^2}} e^{-\frac{r^2}{2\sigma_f^2}}. \quad (5.1)$$

An illustrative example in 2D is provided in Figure 5.16b.

Sessions where a high percentage of fixations come from low-quality, or out-of-range gaze measurements should be avoided, as they would not be representative of the entire viewing experience of one user. Thus, sessions with good tracking accuracy and high percentage of in-range fixations are determined; the former is based on the ratio of the total number of low-confidence gaze positions excluding blinks, divided by the total number of gaze positions, whereas the latter is based on the ratio of in-range divided by the total fixations. A threshold of 17.5% is set for low-confidence and 75% for in-range fixations. If both conditions are satisfied, a session is qualified as valid. The fixation points from valid sessions are aggregated across the subjects for each model, forming a fixation density map.

Head-tracking: In this part of the analysis we use only the head-trajectories in order to compute importance weights for each point of a model. The employed approach has been proposed in previous work on omnidirectional imaging (Upénik et al., 2016), however, it can be straightforwardly applied in our case study. In particular, we decide that a user is not on a transitional head-movement based on a threshold of 20 degrees/sec that we set for the head velocity. For each sample that is falling below this threshold, a cone of angle equal to 10° is emitted based on the recorded head position and rotation. The frontal points of the model under inspection are identified, as explained earlier, and a Gaussian weighting is applied as a function of (only) the distance from the central ray of the cone.

5.2.3 Results

Following the proposed method, 72.22% of the sessions were used to form fixation density maps (15.17 ± 2.48 subjects per model), with an average of 9.92% low-confidence gaze samples and 92.24% in-range fixations. The average number of valid fixations per model is 44.29 ± 7.17 , with a duration of 259.16 ± 30.42 ms, and an angular error of $1.90^\circ \pm 0.84^\circ$. This corresponds to a reduction of 50.90%, 46.33%, and 53.20% with respect to the angular error estimated during *error profiling* for the right, left, and middle gaze, respectively, on the set of valid sessions.

Based on the recorded gaze data obtained from the subjects, in Figure 5.17, we present the probabilities of gaze and fixation position at different viewing angles as measured in the virtual world. These results suggest that the sight of the users tends to deviate from the horizon following a Laplacian distribution with a peak between 5° and 10° .

In Figure 5.18, importance weights from fixation density maps are illustrated using gaze

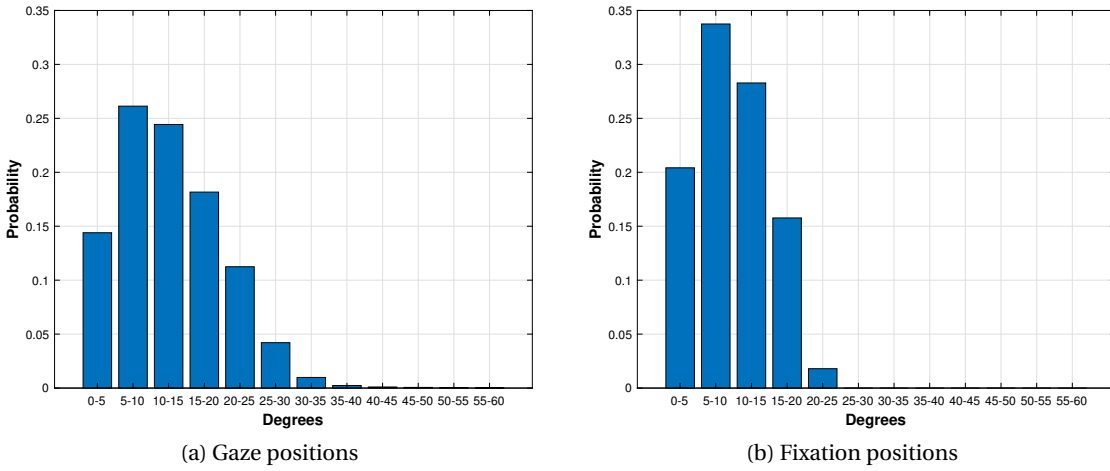


Figure 5.17 – Histograms indicating the probability of gaze and fixation positions as a function of the viewing angle with respect to the head direction.

and head cues. It can be seen that visual attention is attracted by low-level features such as edges and contrast, and high-level features such as faces. Attention is also drawn on text and signs (e.g., *biplane*), on unexpected objects (e.g., chest of *ulli wegner*), and degradations (e.g., surface discontinuities in *egyptian_mask*). These observations are in alignment with trends observed in visual attention experiments using other types of imaging modalities in different environments.

In Figure 5.19, the importance weights obtained using only the head data-streams are presented. We observe that models with regions that attract the interest of people and do not deviate much from the horizon (in the same height, or slightly up or down), the estimations are good. For instance, the faces of human figures, of the upper parts of *amporiskos* and *shiva*. However, when staring up or down, it is natural for the gaze to further extend corresponding head movement towards the direction of interest. In such cases, the gaze predictions, which are obtained from the head directions are poor. This can be seen in the case of *biplane* and *egyptian_mask*, which are inspected from above and, especially in the latter case, where the user's interest was intended to be in the bottom part of the content, as Figure 5.18 suggests.

The average time of interaction found to be similar for both objects and human figures data sets (60.9 ± 10.7 against 56.4 ± 4.6 sec.). A tendency of subjects spending more time on bigger and more complicated objects (e.g., *biplane*) was naturally observed. The models were mostly inspected from mid- to close-range distances. For example, the 76% of the recorded gaze samples in the human figures data set (height of 1.85 virtual units) are collected from distances of inspection that lie inside a circle of radius 2.5 virtual units. In Figure 5.20, the behavior of the users in terms of aggregated time of inspection across all models is indicated. For illustration purposes, a random point cloud is placed in the middle. The virtual room is represented by a grid on a 2D histogram, with each square representing an area of 0.5×0.5 virtual units. Brighter color values indicate larger aggregated time of inspection across all models and users.

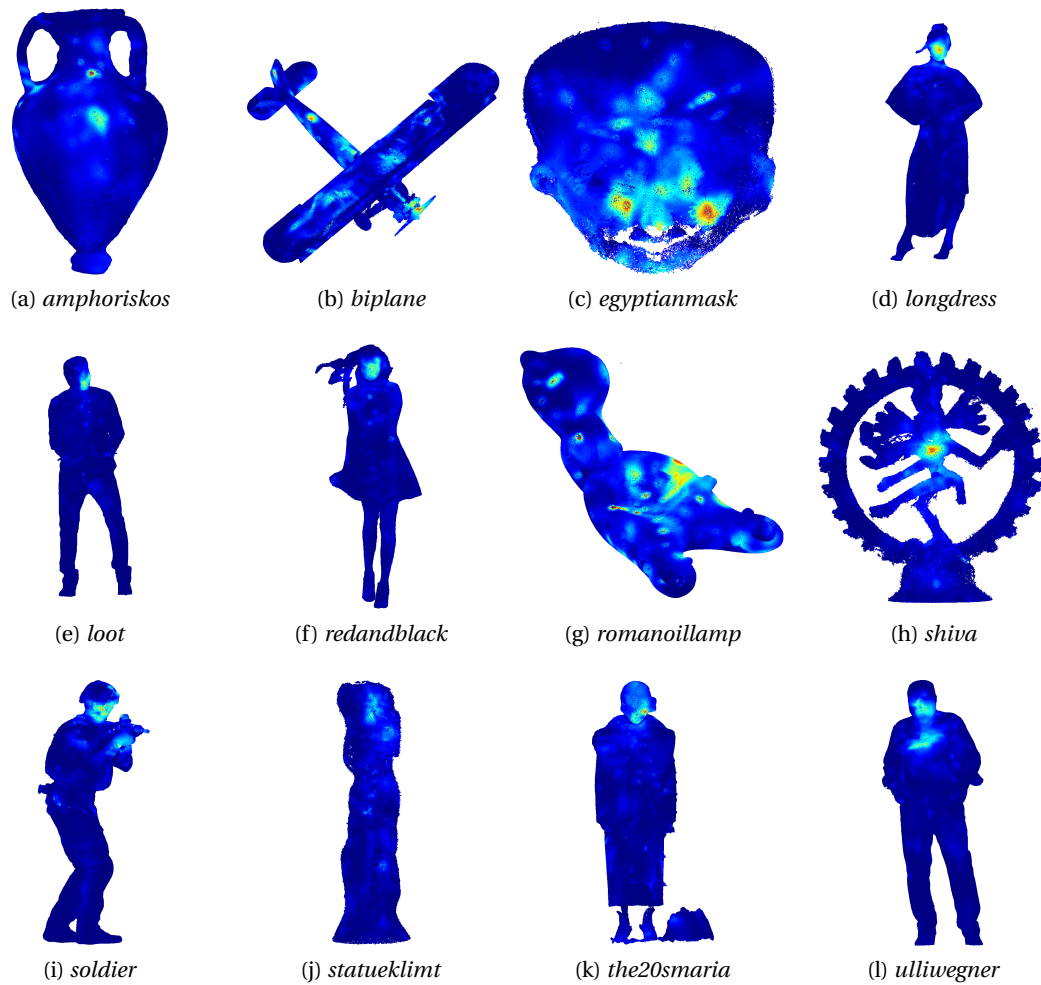


Figure 5.18 – Importance weights from fixation density maps using gaze information.

It is evident that users prefer the frontal and rear views of the models.

Based on post-questionnaires, the majority of the participants were naive users of VR. The immersion level and the total quality of experience was reported to be high, with grades of 4.15 and 4.35 out of 5, respectively. The visual quality of the contents under inspection was graded as 3.7. For the above questions a 5-grading scale was used (5: *Excellent*, 4: *Good*, 3: *Fair*, 2: *Poor*, 1: *Bad*). The discomfort levels were rated low, with 1.15 out of 3 (1: *No*, 2: *Mild*, 3: *Strong*). Finally, regarding the criteria of preference, “realistic” (6), “details (e.g., hair)” (6), “friendliness” (3), and “color” (3) were the most common keywords for human figures, while the most popular for objects were “realistic” (5), “smoothness” (3), “color” (3), and “aesthetic” (2). In parenthesis, the number of keyword occurrence is indicated, in a total of 21 subjects.

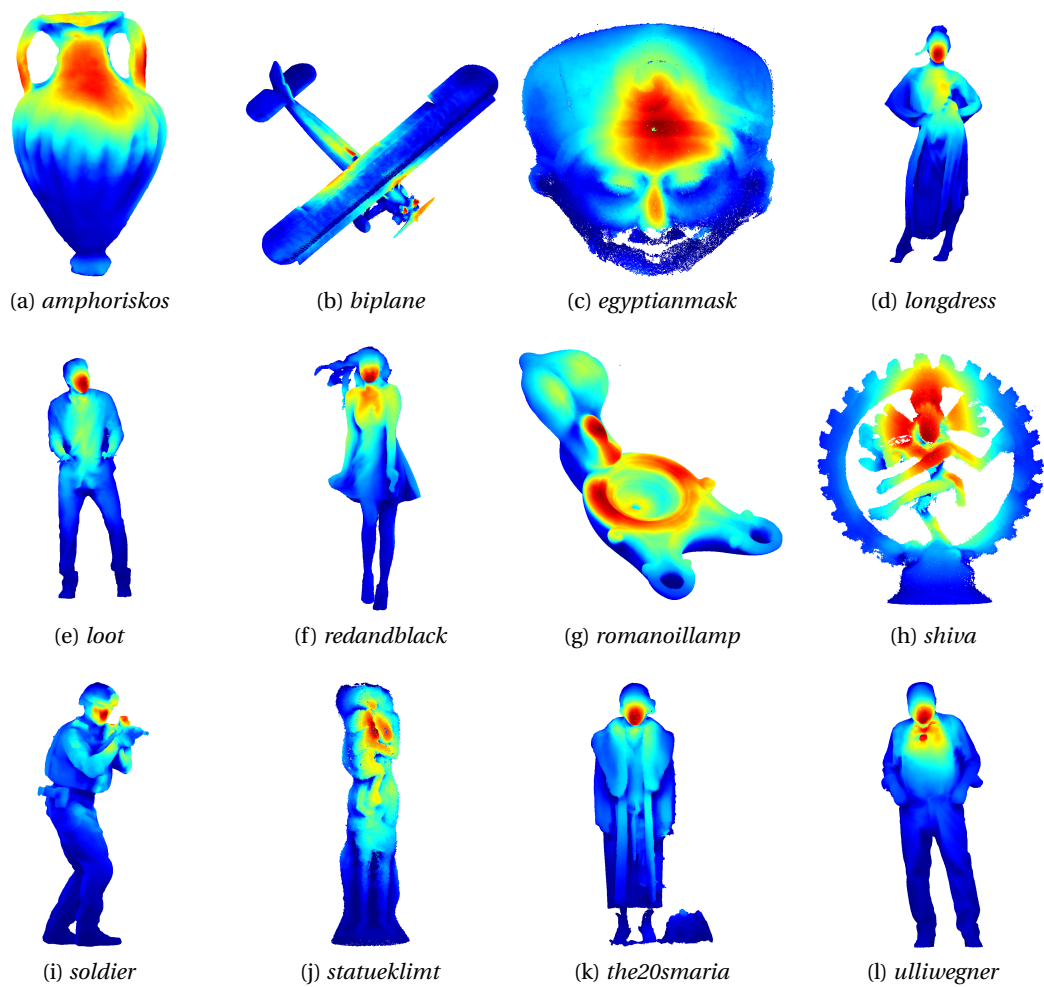


Figure 5.19 – Importance weights from using head-trajectories.

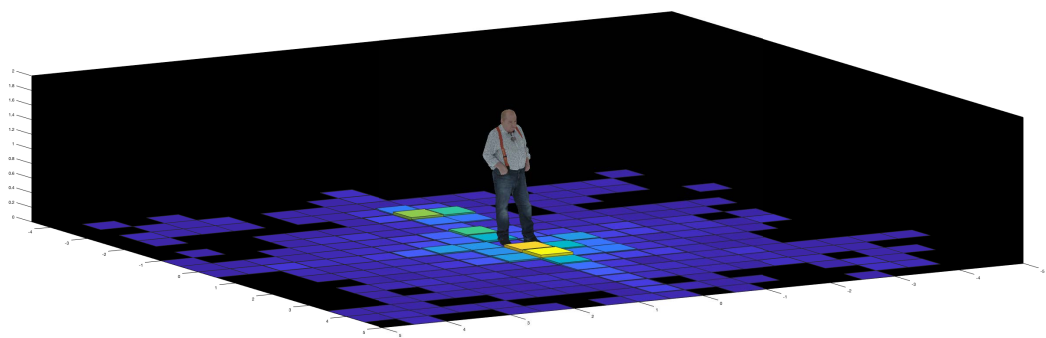


Figure 5.20 – Time of inspection across models and users.

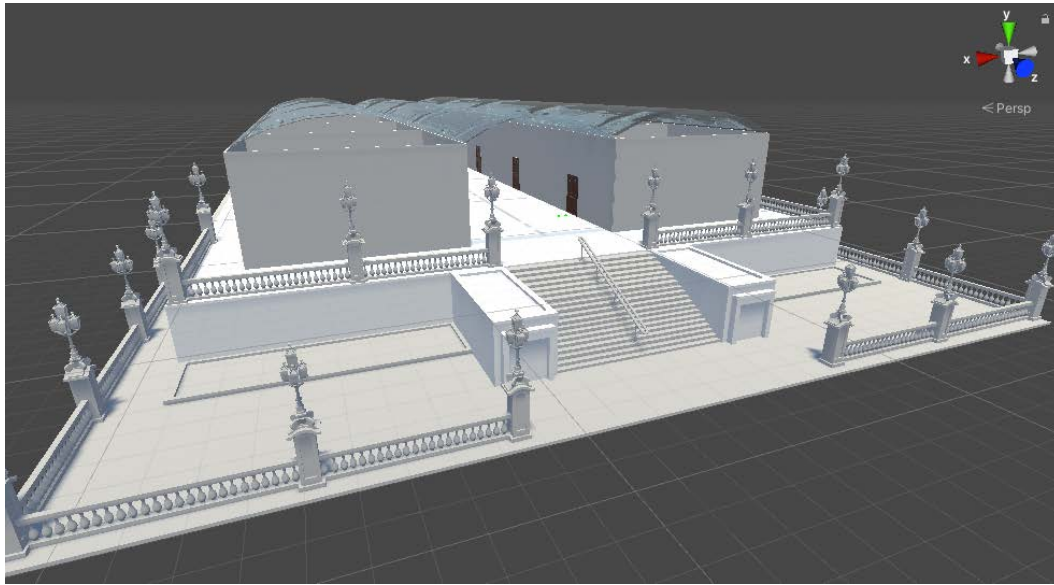


Figure 5.21 – Overview of the virtual museum.

5.3 Towards visual attention in virtual museums

The advances in 3D technologies over the past decades have resulted in integration of AR and VR applications in our nowadays lives. VR technologies, in particular, offer immersive experiences with high quality content, in fully controlled and reproducible sceneries. These denote attractive features that can reduce the parameter space and adjust influencing factors of a viewing session, which is typically challenging in the real world. When VR applications are combined with eye-tracking, the exact visual experience of a user can be recorded, accessed and reproduced with high accuracy. Such information permits studying and analysing the user behavior, which is critical in a wide range of research domains spanning from marketing and gaming, to psychology and neuroscience.

In our previous efforts, user movements in 6DoF VR experiences were tracked in the context of subjective quality evaluation, while eye-tracking was additionally enabled to study visual attention on point cloud contents. For these studies, a generic virtual scene was designed that ensured no distractions, while the evaluation methodology that was adopted promoted exploration of the user. In this work, we design a virtual environment that serves a more specific application, while enhancing the realism of the experience and the naturalness of users interactions with the virtual world. In other terms, in the first case a virtual scene was created to display models, whereas in the second case, a virtual world is constructed where the models are part of it. Specifically, the virtual environment represents a museum comprised of a main corridor and exposition rooms. In each room, a cultural heritage model in the form of point cloud is presented. The user is free to choose the navigation path and in which room he/she will enter, at will. This application aims at simulating a realistic experience of a user visiting a museum, while offering the possibility of tracking both head and gaze cues for

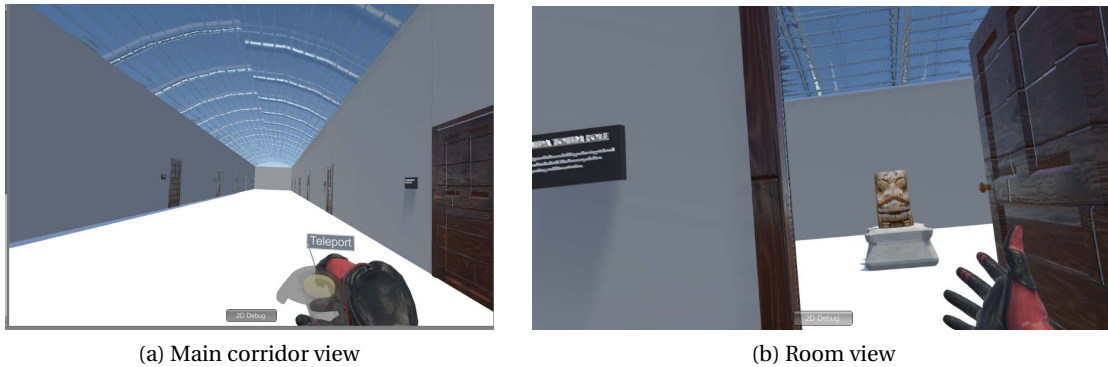


Figure 5.22 – Exemplary views of the virtual environment from the user's perspective.

analysis purposes, provided the corresponding hardware. Moreover, auxiliary tools that are device-dependant are developed as modules that can be enabled or disabled, allowing: (a) better synchronization between gaze- and head-related data streams, and (b) benchmarking of the eye-tracking device.

5.3.1 Scene design

The virtual museum is designed using the Unity 3D development engine and is consumed by means of an HTC VIVE Pro headset. The users are able to navigate in the virtual world with 6DoF through physical movements, while also they can teleport by using the trackpad button of the VIVE controllers (see Figure 5.10b). Optionally, the head position and rotation (i.e., virtual camera) can be recorded in Unity at the system frame rate, while the user's gaze can be tracked by the Pupil Labs hardware that must be integrated in the headset.

An overview of the museum is illustrated in Figure 5.21. As can be seen, there is an external space, which serves as the entrance and is created to better simulate real-life designs. A main corridor is constructed in the middle of the layout, with a number of exposition rooms placed in a symmetric way on the left and the right side. In Figure 5.22a, the view of the user while walking in the main corridor is indicated. The entrance in each exposition room is granted through a door, which opens with a simple hand hover over the door handle using the controller. Upon entrance, a point cloud model appears in the center the room and is oriented so that the frontal view is faced, as depicted in Figure 5.22b. A configuration file carries information regarding the arrangement of models inside the rooms. Each room is identical, with dimensions of $10 \times 10 \times 5$ virtual units. To represent the models, the renderer described in annex D.4 is employed. Thus, prefabricated models of adjustable appearance are loaded in run-time. The size of the models is adjustable, and can be configured in the same configuration file. In our experimentation, we set realistic sizes to facilitate inspection. In case of smaller objects, a stone stage was additionally employed, as presented in Figure 5.22b, to better simulate realistic conditions and to ease user inspection.

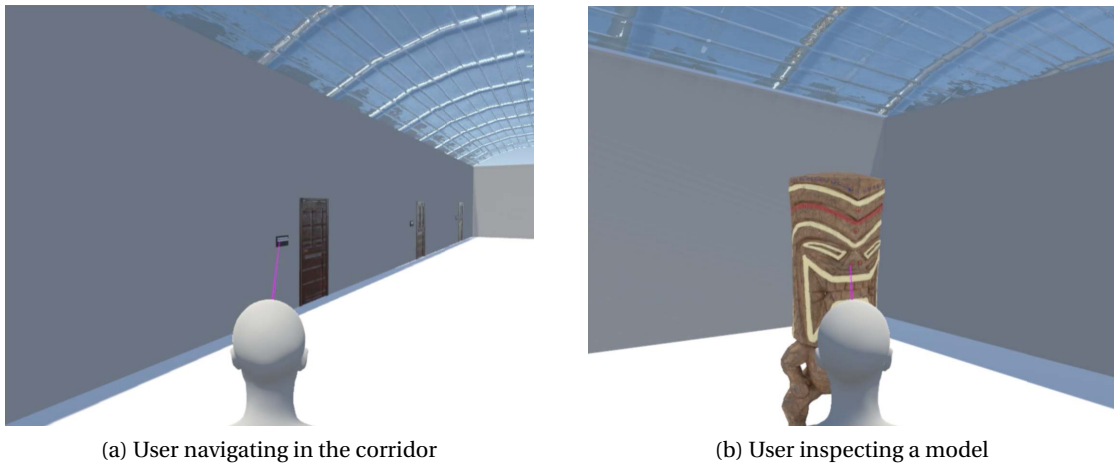


Figure 5.23 – Demonstration of user interactions that is played-back using the recorded data. The head position is indicated by the white avatar and the gaze direction by the purple line.

Users can freely navigate in the virtual space; however, certain limitations are enabled to avoid confusions and enhance realism. For instance, users cannot walk on the stairs placed in the entrance of the museum, since it is a rather unrealistic experience when not accompanied with external stimuli that simulate the feeling of changing ground level. Moreover, users cannot enter a room through the walls, rather, they should use the door handle. The latter also permits to efficiently display and hide models upon user's entrance in a room, in order to reduce the rendering costs and increase system responsiveness (i.e., assets that are not currently inspected, are not loaded).

5.3.2 Supplementary tools

Time synchronisation between Unity and Pupil Labs

In eye-tracking experiments in 6DoF virtual environments, both head and gaze information is required in order to extract the position of the user and the point of interest at any time instance. In our experimental set-up, the gaze cues are recorded by the Pupil Labs software, whereas the head cues are recorded in Unity, which has access to the virtual camera parameters (i.e., position and orientation) in world coordinates. Provided that each application has its own clock, the two time series (i.e., data streams) need to be synchronised.

In our previous efforts, the adopted methodology relied on subscription of Unity to the gaze topic implemented by Pupil Labs in order to receive the respective data. The received packages were extracted and, at every frame refresh, the latest available gaze-relevant information was recorded together with the current head data in an external file. This way, head and gaze information was synchronized per effective frame. In the current implementation, we synchronize the gaze and head data streams using the Pupil Labs `TimeSync.cs` script,

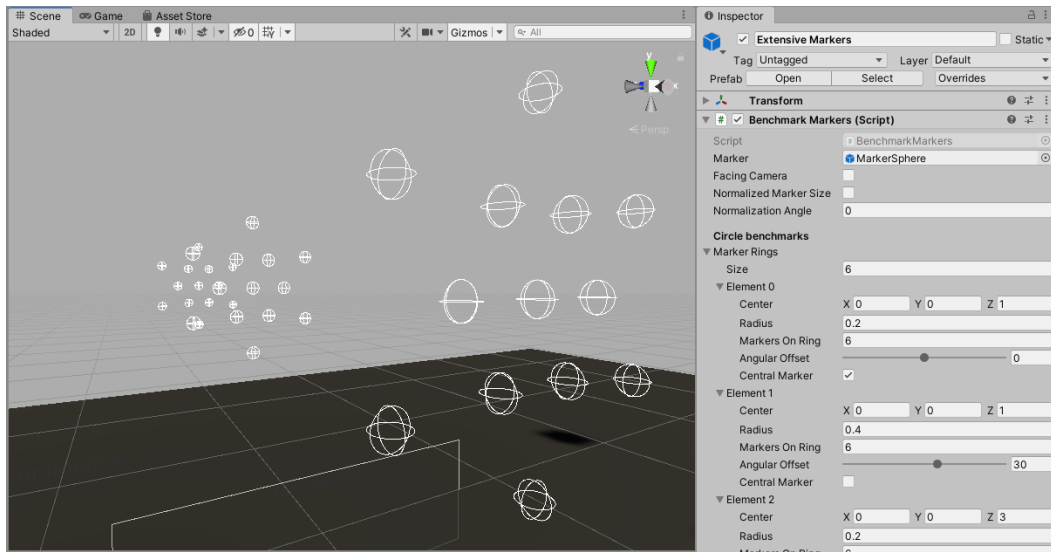


Figure 5.24 – Menu for configuration of markers.

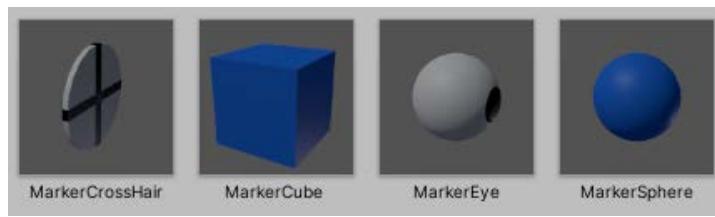


Figure 5.25 – Built-in markers integrated in the module.

which calculates the time offset between the two clocks. In particular, we call the function `TimeSync.GetPupilTimestamp()` from Unity, in order to obtain the Pupil Labs time. We store as t_S the Unity time the function is called and as t_R the Unity time we receive the response. The Unity time that corresponds to the Pupil Labs time returned from the function is set as $0.5 \times (t_S + t_R)$, in order to account for the round-trip communication delay.

In Figure 5.23, a snapshot from the play-back of a user's experience based on the recorded gaze and head-data streams is demonstrated. The white avatar indicates the head position, and the purple line the gaze direction.

Module for benchmarking of eye-tracker

A benchmarking module was developed to quantify the accuracy of the eye-tracking device, by computing the angular difference between the estimated and the target gaze point. This way, the quality of the recorded material can be evaluated and conclusions can be drawn regarding their validity. Moreover, such an application may allow investigation of the device performance under different conditions, such as the position of the headset, or the subject's physiology (i.e., eyes color and shape, wearing contact lenses) among others, which might

affect the accuracy of the measurements.

The benchmarking scene is simple and non-distracting. A sequence of markers with known position and random presentation order appear consecutively, for a pre-determined time duration. The subjects are asked to stare at the markers as soon as they appear, which is very similar to a typical calibration procedure. Provided the marker position, the accuracy of the measurements to estimate the gaze position can be computed.

Regarding the markers, we consider four aspects that can be adjusted through a menu that is provided in the Unity editor, illustrated in Figure 5.24.

Appearance: Different geometrical shapes and textures can be used as markers. In our package the spherical, cubic, eye and cross-hair marker indicated in Figure 5.25 are provided.

Position: The marker positions should allow for a variety of gaze angles in order to compute the device performance under a wide range of eye movements. Moreover, the markers should be placed at different depths in order to account for different eye vergence. In our implementation, the markers positions are set by defining a set of rings. The shape of each ring is defined by its “Radius” and the “Center position”, as shown in Figure 5.24. Moreover, the “Number of markers” is set individually for each ring and they are placed at equally spaced positions. The “Angle offset” option sets the angle of the markers around the ring, allowing a wide range of constellations. Note that the “Central Marker” option adds a marker at the center of the ring.

Orientation: The orientation of the markers can be adjusted to face the camera (i.e., the subject) through the “Face camera” option. For the spherical marker, this option is obviously superfluous. However, for the cross-hair marker, for example, it allows the subject to target at the markers head-on, and not from the side.

Size: Finally, the “Normalize marker” option sets the size of each marker so that it takes up a certain angle in the subject’s field of view. This angle is set with the “Normalization angle” setting, which can be useful to prevent close markers from being too large.

In this section we have describe our recent efforts in creating virtual worlds, as well as our considerations and development of tools that allow to accurately capture head and gaze information of users interacting with it. Preliminary tests have been concluded with success (see Figure 5.23) and are inspiring for further experimentation that will be carried in the future.

5.4 Conclusions

In this chapter we exploit VR technology to conduct subjective experiments with point cloud contents. Virtual environments are designed to serve our purposes and rigorously described to allow reproducibility. In particular, a simple, non-distracting scene is developed to perform subjective quality assessment with 6DoF of point clouds under color distortions occurring from the two color encoding modules of the state-of-the-art MPEG G-PCC encoding engine. A well-established test method is adjusted to the interactive nature of the inspection protocol that was adopted, and a new variant is proposed. The results indicate statistical equivalence of the two color codecs that were tested, based on the subjective scores obtained from both test methods. Moreover, the proposed alternating DSIS protocol was found to result in lower uncertainty for the perceived quality of the displayed stimuli, and it was generally preferred by the participants. Analysis of the interaction patterns extracted from the recorded navigation of the subjects during evaluation showed a preference for close-range, frontal view examination.

The same scene was then employed in an eye-tracking experiment that was conducted to identify regions of interest for popular point cloud models. A task-dependent viewing scenario was adopted and head-plus-gaze information was recorded in real-time. The experiment was split in two sessions, and subjects were asked to visualize the set of models that belongs to the same content type (i.e., objects and human). At the end of the experiment, the participants were requested to set a criterion of preference and order the models accordingly. Based on the received feedback, the “realism” was the most common criterion for both types of contents, which coincides with our prior expectations for this experimental set-up. After processing the recorded material, fixation density maps were extracted in the form of importance weights. To improve the accuracy of our measurements and to compensate limitations of the eye-tracking hardware due to headset slippage, a method to exploit the highest-quality recorded gaze data was introduced based on a per-session error profiling, reducing remarkably the average angular error. Moreover, a scheme to determine areas of fixations in a point cloud was proposed, dealing with the particularity of this type of content representation; that is provided that points have no dimensions, common techniques that make use of colliders cannot be exploited.

Finally, we develop and describe functionalities that have been integrated in a proof-of-concept application of a virtual environment that allows more realistic human interactions. This virtual world essentially represents a museum, which is a common VR use-case. Associated modules that improve the synchronization between gaze and head data streams are implemented and described, while also a scene to benchmark the gaze measurements that are obtained from an eye-tracking hardware is detailed, providing insights that can be useful in future attempts.

For the purposes of our experimentation, a high-quality point cloud data set was generated, so-called *PointXR dataset*, which was publicly released. Moreover, the application that was employed in our VR experiments and additional rendering tools, namely *PointXR toolbox*, have

been also freely distributed. Finally, the subjective scores collected from the experiment that was conducted to assess the G-PCC color encoding module, form the *PointXR experimental data* which was also made publicly available. Information on how to retrieve and where to refer for additional information are provided in annex E.

Modelling perceptual quality Part II

6 Point-based objective quality metrics

Objective quality evaluation is a research area involved in the design of algorithms that predict visual quality of contents, typically as they would be perceived by human end-users. This research field is impactful on several tasks that are related to information and communication systems. For instance, having access to accurate predictions of visual quality for contents after encoding or transmission can greatly assist in improving user experience, by updating corresponding configurations of the underlying systems to reduce perceptual impairments. Moreover, the benchmarking of new solutions can be facilitated by carrying out performance evaluation analysis using objective scores from well-performing predictors instead of human opinions. The latter are collected from subjective experiments, which are assumed to reveal ground-truth visual quality ratings; yet, they are costly and cumbersome, as well as limited in terms of ad-hoc implementation and large scale realization.

The development of predictors that accurately decide the level of visual distortions from realistic types of degradations (e.g., noise, compression) for different imaging modalities, has been at the center of attention of the research community with a relevant interest for many years. The initial focus was naturally drawn on conventional images, where it was early understood that naive implementations of error quantification in a pixel-by-pixel basis, e.g., MSE, did not correlate well with human judgements. As a consequence, efforts were concentrated on approaches that consider characteristics of the human visual system. These, in principle, can be categorized as bottom-up, and top-down. The former denote theoretical approaches that aim at measuring perceived errors in a content, whereas the latter signify engineering solutions that aim at capturing properties of human visual perception. Objective quality metrics can also be clustered based on the availability of the original version of the content at run-time as full-reference, reduced-reference and no-reference metrics.

In the field of 3-D imaging, top-down full-reference approaches are the most common. They have been largely explored in the case of polygonal meshes, and more recently extended to point cloud data. In fact, a substantial amount of work has been lately carried out on the latter type of content representation, which has led to numerous new objective methods for perceptual quality prediction. Current point cloud metrics can be classified as (a) point-based,

and (b) image-based. The former class operates on the point cloud domain, thus, requiring as inputs point cloud contents, whereas the latter predictors function on the image domain, making use of algorithms that are applied on projected views of the models.

In this chapter, we describe and validate our contributions in point cloud objective quality assessment that relies on the primal 3-D data. The proposed solutions exploit explicit and/or implicit information that is carried in a point cloud format, hence, falling in the point-based class.

We initiate by defining the point cloud angular similarity metric, hereafter often referred to as plane-to-plane. This predictor makes use of normal vectors in order to compute the angular similarity of tangent planes between an original and a degraded model, capturing geometry-only degradations. Its performance clearly depends on the quality of the relevant attribute data, and the approximations the latter provide for the underlying model surfaces. To shed light on the matter, the plane-to-plane metric is benchmarked under different normal estimation algorithms and configurations, using several subjectively annotated data sets. This performance analysis allows us to calibrate the metric, and obtain insights regarding the relationship between surface approximations and prediction accuracy.

We then proceed to the definition of the point cloud structural similarity metric, also cited as PointSSIM. This method relies on statistical dispersion measurements that characterize distributions of point cloud topology and/or color properties in local neighborhoods. In particular, features that capture local variations in the selected attribute domains (i.e., location, normal, curvature, and color) are extracted and compared between an original and a degraded model, resembling the operation of the well-known SSIM (Wang et al., 2004). To compute relevant statistics, a series of dispersion estimators is recruited. Moreover, a voxelization step is proposed and applied prior to feature extraction, in order to eliminate intrinsic resolution differences across stimuli and mimic distant inspection. The performance of the metric is benchmarked under different attributes, dispersion estimators, neighborhood sizes, and voxel resolutions, against various subjectively annotated data sets. This process permits in-depth understanding of the metric's performance, with respect to the parameter configuration.

This chapter is based on material that has been published in (Alexiou and Ebrahimi, 2018c, 2020).

6.1 Point cloud angular similarity

In the visualization process of a point cloud content, the human brain tends to interpolate the individual point samples in order to perceive the underlying model. Degraded versions of a content typically lead to a different number of point samples, and coordinates that deviate from their original position, introducing visual impairments. Such perceptual differences could be quantified by a measurement of similarity between corresponding surfaces fitted to the pristine and the impaired point samples.

To obtain a surface similarity measurement, one solution is to convert the point cloud data to mesh representations. However, this approach brings the issue of sensitivity on the selection and configuration of the surface reconstruction algorithm by introducing extra complexity and ambiguity in the obtained objective quality scores. A relevant subjective study described in section 3.4, reveals that distortions applied on the point cloud domain lead to visual artifacts of different nature after surface reconstruction, which might be rated differently. A simpler approach is to consider tangent planes to estimated fitted surfaces. In particular, the dihedral angle between tangent planes quantifies the local similarity of corresponding surfaces by measuring the orientation difference. Using the angular similarity formula instead, an equivalent measurement is obtained, additionally bringing the advantages of a distance metric.

Our algorithm aims at capturing perceptual degradations by computing the similarity of surface approximations between a pristine and an impaired model. In particular, it relies on the angular similarity between tangent planes that correspond to pairs of nearest points from the reference and the model under evaluation. The points can be interpreted as discrete samples drawn from underlying continuous surfaces, and the tangent planes as local linear approximations of the surfaces at those particular coordinates. Finally, the tangent planes are defined as perpendicular to normal vectors, which can accompany coordinate data in a point cloud format; hence, the latter are employed in practice to compute angular similarity scores.

In this section the proposed point cloud angular similarity metric is defined. Implementation details are provided and limitations of the algorithm are explicitly described. The performance of the metric is analysed under different approximations of underlying surfaces, which are reflected on the estimated normals, and against numerous subjectively annotated data sets.

6.1.1 Definition

Let us consider a point a , with its associated normal vector \vec{n}_a that belongs to the set of points A , which represents model O_A . Let us also consider another point b , with its normal vector \vec{n}_b that belongs to another set of points B , which represents another model O_B . Let us finally assume that the coordinates of a and b are identical, as shown in Figure 6.1. The difference between the normal vectors \vec{n}_a and \vec{n}_b is expressed through the angle θ , which is equal to the angle between the corresponding tangent planes perpendicular to these normals. Differently oriented tangent planes indicate that different local surfaces connect the points a and b with their corresponding neighbors in sets A and B . Thus, a larger angle θ implies a larger difference between the local surfaces of models O_A and O_B , respectively

Let us assume that point cloud A is the pristine content and point cloud B is an impaired version. By setting A as the reference, for each point b that belongs to B , the nearest point a is identified in the reference using the Euclidean distance, in order to measure the angular similarity of adjacent surfaces.

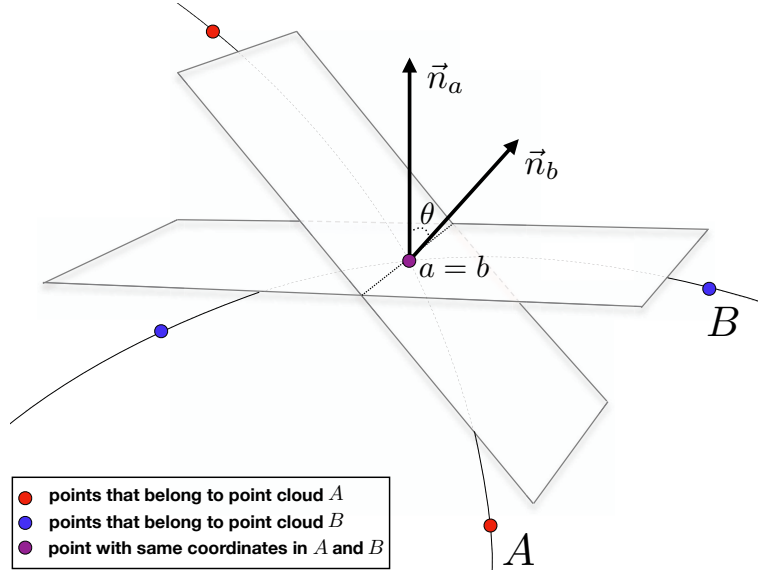


Figure 6.1 – Point cloud angular similarity metric.

The cosine similarity sim between the tangent planes of a and b is initially derived according to Equation 6.1

$$\text{sim} = \cos(\theta) = \frac{\vec{n}_a \cdot \vec{n}_b}{\|\vec{n}_a\| \|\vec{n}_b\|} \quad (6.1)$$

where θ is the angle between \vec{n}_a and \vec{n}_b and $\text{sim} \in [-1, 1]$. Then, the inverse cosine is computed and the angle $\hat{\theta} = \arccos(\text{sim})$ is estimated. Notice that different notations are used, since $\theta \in [0, 2\pi]$ while $\hat{\theta} \in [0, \pi]$, by the definition of the inverse cosine. Considering that we are only interested in the angular similarity between tangent planes, we want to keep the minimum out of the two angles that can be formed between the intersecting planes (i.e., dihedral angle); thus, we define $\tilde{\theta} = \min\{\hat{\theta}, \pi - \hat{\theta}\}$, with $\tilde{\theta} \in [0, \pi/2]$. An equivalent expression is given by Equation 6.2

$$\tilde{\theta} = \arccos(|\text{sim}|) \quad (6.2)$$

where sim is calculated using Equation 6.1. Notice that this quantity essentially reflects the angle between two unoriented normal vectors.

Lastly, the angular similarity $\text{asim}_{B,A}(b)$ between the tangent planes of a and b , bounded in the range $[0, 1]$ and using A as reference, is given by Equation 6.3.

$$\text{asim}_{B,A}(b) = 1 - \frac{2\tilde{\theta}}{\pi} \quad (6.3)$$

After computing a similarity value for each point of the point cloud under evaluation, in this case B , a pooling method $\mathcal{P}(\bullet)$ is applied in order to calculate a global score $\text{ASIM}_{B,A}$ that

characterizes the degradation of B with respect to A , as shown in Equation 6.4.

$$\text{ASIM}_{B,A} = \mathcal{P}(\text{asim}_{B,A}(b)) \quad (6.4)$$

Indicative examples of widely-used pooling algorithms for the computation of a global degradation score are given in Equations 6.5 and 6.6.

$$\mathcal{P}(\text{asim}(m)) = \frac{1}{M} \left(\sum_{m=1}^M \text{asim}(m)^q \right)^{\frac{1}{p}} \quad (6.5)$$

$$\mathcal{P}(\text{asim}(m)) = \frac{\sum_{m=1}^M w(m) \text{asim}(m)}{\sum_{m=1}^M w(m)} \quad (6.6)$$

where m indicates a point and M the cardinality of the point cloud under evaluation, $\text{asim}(m) = \{\text{asim}_{B,A}(m), \text{asim}_{A,B}(m)\}$ is the angular similarity by respectively setting A and B as the reference, $w(m)$ denotes a corresponding weight, and $q, p \in \mathbb{R}_{0+}$. More often, the mean and the MSE are employed for pooling.

In an analogous way, the global score $\text{ASIM}_{A,B}$ is derived to reflect the degradation of A with respect to B , after setting the latter as reference. The final point cloud angular similarity (or plane-to-plane) score ASIM is defined as the symmetric error, obtained after setting both models as reference and keeping the minimum out of the two global degradation scores, as depicted in Equation 6.7.

$$\text{ASIM} = \min\{\text{ASIM}_{B,A}, \text{ASIM}_{A,B}\} \quad (6.7)$$

The description of the metric is summarized in Algorithm 1.

Algorithm 1

- 1: Set as reference point cloud A
 - 2: **for all** $b \in B$ **do**
 - 3: Identify a as the nearest neighbor of b in A
 - 4: Compute angular similarity $\text{asim}_{B,A}(b)$
 - 5: Compute global degradation $\text{ASIM}_{B,A}$
 - 6: Set as reference point cloud B
 - 7: **for all** $a \in A$ **do**
 - 8: Identify b as the nearest neighbor of a in B
 - 9: Compute angular similarity $\text{asim}_{A,B}(a)$
 - 10: Compute global degradation $\text{ASIM}_{A,B}$
 - 11: Compute angular similarity (plane-to-plane) ASIM
-

Complexity

The complexity of the proposed algorithm is limited by the selection of the algorithm to identify nearest neighbors. In particular, let us set point cloud A as the reference content.

Assuming a linear search approach, the computational complexity to specify a nearest neighbor a for a point b , would be $\mathcal{O}(M)$, with M the cardinality of A . Following a k -d tree approach, a space-partitioning data structure of M points should be initially constructed, which is an operation of $\mathcal{O}(M \log M)$. Then, the search in the k -d tree to determine a nearest neighbor a for a point b , is an operation of $\mathcal{O}(\log M)$. After establishing a pair of associated points, the angular similarity is computed. Considering that the calculation is constant in regard to the number of points, a cost of $\mathcal{O}(1)$ is added on the top. The aforementioned procedure, excluding the potential k -d tree construction, is repeated $\forall b \in B$. Then, the computation of the global degradation score poses an additional complexity of $\mathcal{O}(N)$, with N the cardinality of B , since it is a function of the number of points of the point cloud under evaluation B . Analogously, we obtain the computational costs after setting point cloud B as the reference. The consolidated computational complexity of the proposed algorithm is $\max\{\mathcal{O}(M \log M), \mathcal{O}(N \log N)\}$, assuming a k -d tree, or $\mathcal{O}(NM)$ assuming a linear search approach for the identification of the nearest neighbors.

Limitations

The main limitations of the proposed metric are: (a) It captures geometry-only degradations. (b) It is a full-reference metric, indicating that both the pristine and the impaired point clouds should be available in order to compute an objective quality score. (c) The association of points between the reference and the model under evaluation to compute the angular similarity is based on nearest neighbors in the Euclidean space, which implies that the metric is vulnerable to translation and scaling. (d) It relies on normal vectors, requiring this attribute to coexist with the coordinates of both the pristine and the impaired point clouds, or to be estimated in case of absence. No specific normal estimation methodology is imposed on our side as part of the metric's implementation. Ideally, normals would be given as attributes along with the point cloud content. However, this is often not the case. Since the accuracy of this predictor depends on how the normal vectors approximate the underlying surfaces, we analyse the performance of the metric under different surface approximations and subjectively annotated data sets, showing the results in section 6.1.2.

6.1.2 Validation methodology

Data sets

A total of 4 subjectively annotated data sets is recruited in order to evaluate the performance of the point cloud angular similarity metric, which are briefly summarized below.

G-PCD: This data set has been assembled from our efforts published in (Alexiou and Ebrahimi, 2017b). It consists of 5 geometry-only static point clouds that are generated by different means, depicting simple objects. The contents are distorted by injecting Gaussian noise and after

Octree-pruning at 4 degradation levels. The subjective experiments were conducted using an interactive platform in a desktop set-up and the models were rendered as collections of points. Two test methods were adopted, namely, simultaneous DSIS and ACR. Provided the different type of visual distortions from each degradation type, a different session was held per test method and type of impairment. In our analysis, subjective scores obtained from all sessions are considered. More details regarding the generation of stimuli and the subjective experiments are provided in sections 3.1 and 3.2, respectively.

J-PCED2: The so-called J-PCED2 data set is published in (Perry et al., 2020) under the framework of activities conducted by JPEG experts that participated in the Exploratory Study 2, issued by the JPEG Pleno AhG on Point Clouds. It contains 6 colored static point clouds that represent human figures, whose geometry and color is encoded using the V-PCC and two G-PCC variants at 5 degradation levels. Regarding the latter, the Octree and TriSoup modules are enabled for geometry encoding and the Lifting module for color encoding. The contents are compressed following the MPEG Common Test Conditions document (MPEG 3DG, 2017); the exact encoding configurations can be found in the respective paper. The encoded stimuli were subjectively evaluated using points of fixed size in four independent laboratories, under a passive evaluation protocol. The MOS that serve as the ground truth in our analysis are obtained after merging the scores from the participated laboratories, since they were found to be highly correlated. Further details can be found in (Perry et al., 2020).

M-PCCD: This data set is created from our efforts that are published in (Alexiou et al., 2019b). It contains 8 colored static point clouds that represent both human figures and inanimate objects, whose geometry and color is encoded using the V-PCC and the four G-PCC variants (i.e., Octree-plus-Lifting, Octree-plus-RHAT, TriSoup-plus-Lifting and TriSoup-plus-RAHT) under the MPEG Common Test Conditions (MPEG 3DG, 2017). The compressed point clouds were rendered using adaptive point sizes in subjective experiments that were conducted in two independent laboratories, following an interactive assessment protocol. Since the results from the two experiments were strongly correlated (Alexiou et al., 2019b), the ground-truth MOS that are considered are obtained after pooling together the two subjective rating populations. We refer to details about the generation of stimuli and the subjective experiment in sections 9.1 and 9.2, respectively.

IRPC: This data set is published in (Javaheri et al., 2019). It consists of 6 static colored point clouds whose geometry only is compressed using three codecs, namely, V-PCC, G-PCC (TriSoup module) and PCL, at 3 degradation levels. Two of the models were selected from the class *inanimate objects*, two were obtained from the class *buildings and facades*, and the remaining two from the class *people* of the MPEG data set. The point clouds were subjectively evaluated in three different sessions, including and excluding color information. In this study, we make use of scores from the *rpoint* session, which was conducted using a fixed-size point-

based rendering without color information. The point clouds were evaluated passively, after sequential inspection of video sequences showing the reference and the distorted models. Further details can be found in the corresponding paper (Javaheri et al., 2019).

All data sets consist of point clouds with diverse characteristics, resulting from the different nature of the represented models and the acquisition technologies that were employed. Moreover, the wide span of degradation schemes leads to different types of artifacts, making them representative and suitable candidates for benchmarking purposes.

Computation of quality metrics

To evaluate the performance of the proposed method on G-PCD, the normals of all stimuli are estimated using the k -nearest neighbor (k -nn) plane fitting algorithm with $k = 6$, as implemented in PCL. Angular similarity scores are then computed using the estimated normal vectors between pairs of associated points from the model under evaluation and the corresponding reference. In this data set, we use both the mean (i.e., AVG) and the MSE pooling methods in order to calculate a corresponding global degradation score for each model under evaluation. For every pooling method, a plane-to-plane score is obtained using the symmetric error, as per Equation 6.6.

As described in section 6.1.1, the performance of the proposed metric depends on normal vectors and how they approximate the underlying surfaces. Therefore, in order to quantify their impact in the prediction accuracy of the metric, as part of our subsequent analysis we consider different algorithms and configurations for normal estimation. In particular, we choose three widely-used schemes, namely, (a) plane fitting using k -nn, (b) plane fitting using range search, and (c) quadric fitting using range search with radius R , also referred to as R -search, which are applied on different neighborhood sizes around every queried point. For the former algorithm, we use the MeshLab implementation, whereas for the latter two, the CloudCompare software is employed. Note that in annex C, the accuracy of these schemes is evaluated against ground-truth normal vectors, by means of angular error. Moreover, implementation details are provided and valuable insights are drawn regarding their performance on this task.

The prediction accuracy of the plane-to-plane metric subject to the aforementioned normal estimation approaches is analysed using the J-PCED2, M-PCCD and IRPC data sets. To avoid including the same location data more than once in the formulation of neighborhoods for computing the normals, duplicated coordinates are discarded, for each stimulus of every data set. After this pre-processing step, we proceed to estimate the normals. Specifically, using the k -nn with plane fitting algorithm, neighborhoods of 8, 16, 32, 64, 128, 256, 512 and 1024 points are employed, for all data sets. Using range search with plane and quadric fitting, an R of 5, 10, 20, 30, 40 and 50 is used in the former two data sets (i.e., J-PCED2 and M-PCCD). In the case of IRPC, to account for the presence of the same contents at multiple resolutions i.e., *facade*, *frog*, *house*, and *mask*, the radius is adjusted accordingly. In particular, contents of this data set with voxel resolution equal to 12 bits were down-voxelized to 10 bits before encoding with

V-PCC, in order to respect the limitations of the codec. Thus, the 10-bit content version serves as the uncompressed reference for V-PCC encodings, whereas the 12-bit version denotes the reference for G-PCC and PCL encodings. In order to estimate normals over the same region of such a content, the range search radius for stimuli with resolution of 4096 is multiplied by 4 with respect to stimuli of 1024 resolution. At the end, a radius of 5, 10, 15, 20, 25 and 30 is used for 10-bit contents, whereas a radius of 20, 40, 60, 100 and 120 is employed for 12-bit contents, respectively.

Using every selected algorithm and configuration, the normal vectors of each point cloud of a data set are determined. Based on the estimated attributes, angular similarity scores are computed for a model under evaluation with respect to a reference. The MSE pooling method is applied to obtain a global degradation value, and a plane-to-plane score for a stimulus is given by the symmetric error, as described in section 6.1.1.

For comparison purposes, well-established objective quality metrics are additionally evaluated on all data sets. In particular, the point-to-point with MSE, the point-to-plane with MSE, the corresponding geometric PSNR variants, and, for colored stimuli, the color PSNR computed on the luminance channel are employed. It should be noted that the PSNR point-to-point and point-to-plane with MSE, also referred to as PSNR D1 and PSNR D2 respectively, together with the color PSNR in Y, U and V components, were employed by the MPEG standardization committee to carry out objective quality evaluation in the recent point cloud Call for Proposals (MPEG 3DG and Requirements, 2017). In our analysis, we additionally report the prediction accuracy of the non-PSNR variants, since they were often found to outperform the alternatives in the selected data sets. Moreover, we exclude the color PSNR in U and V chromatic components, given that they were found to consistently under-perform with respect to the luminance-based predictions.

The metrics are computed using the software ver. 0.13.5 (Tian et al., 2017c). For the execution of the point-to-plane, default normals that are coming with the released contents are employed when possible, otherwise they are estimated. In particular, k -nn plane fitting using $k = 6$ as implemented in PCL is used for the G-PCD contents. Moreover, k -nn plane fitting with $k = 12$ as implemented in PCL is employed for contents *ricardo10* and *sarah9* from the J-PCED2, and the contents *amphoriskos*, *biplane*, *head*, *romanoillamp* and *the20smaria* from the M-PCCD data set. For IRPC, the normals published with this data set are recruited. In order to compute the geometric PSNR variants for the stimuli of G-PCD, the maximum nearest-neighbor distance of a pristine content is automatically set as the peak signal value, provided that the point clouds of this data set are not voxelized. For the rest of the data sets, the voxel grid resolution of the pristine models is given as input to the software, and the corresponding voxel grid diagonal is set as the peak in the numerator of the ratio. For the calculation of the luminance PSNR, the color attributes are converted from the original RGB to the YCbCr color space, following the ITU-R Recommendation BT.709-6 (ITU-R BT.709-6, 2015), as implemented in the same software. In all cases, the symmetric error is utilized, which is obtained by setting both the pristine and the impaired model as a reference, and keeping the maximum error.

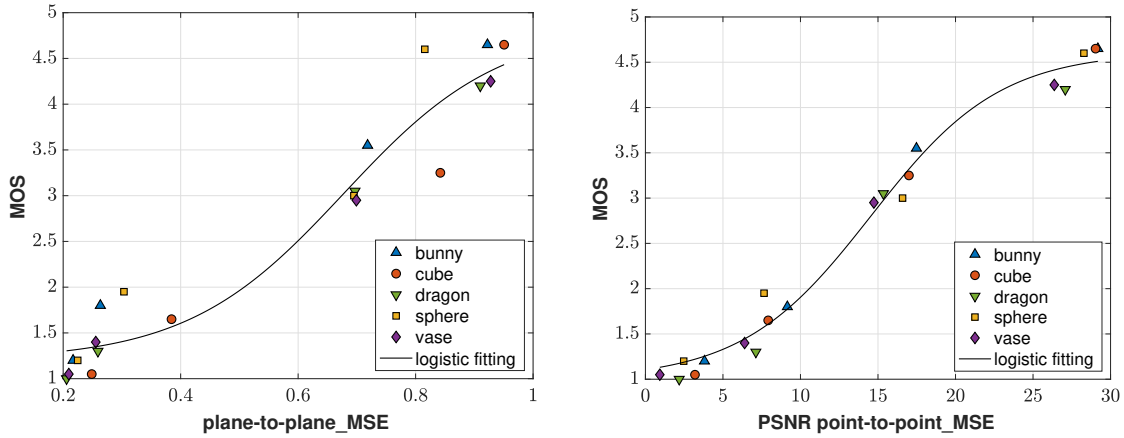


Figure 6.2 – G-PCD: Subjective against objective scores from the best-performing proposed (left) and anchor (right) quality metrics under Gaussian noise, using the ACR test method.

Table 6.1 – G-PCD: Performance indexes of objective quality metrics under Gaussian noise, for both test methods.

	ACR test method				DSIS test method			
	PLCC	SROCC	RMSE	OR	PLCC	SROCC	RMSE	OR
plane-to-plane_AVG	0.963	0.967	0.372	0.450	0.959	0.947	0.392	0.400
plane-to-plane_MSE	0.969	0.967	0.339	0.400	0.965	0.947	0.364	0.400
point-to-point_MSE	0.977	0.927	0.302	0.500	0.984	0.948	0.249	0.300
point-to-plane_MSE	0.978	0.937	0.288	0.350	0.937	0.921	0.485	0.650
PSNR point-to-point_MSE	0.993	0.985	0.162	0.150	0.993	0.971	0.168	0.100
PSNR point-to-plane_MSE	0.992	0.978	0.172	0.150	0.991	0.963	0.188	0.150

Benchmarking of quality metrics

To benchmark the objective quality metrics, we follow the methodology described in section A.3. In particular, the subjective MOS are considered as the ground truth and are compared to predicted MOS values that are obtained from the objective methods, using logistic regression. Then, the PLCC, the SROCC, the RMSE, and the OR are computed between the MOS and the predicted MOS values, to account for linearity, monotonicity, accuracy and consistency of the quality predictors, respectively.

6.1.3 Results

Performance evaluation on G-PCD

In Table 6.1, the performance indexes of the objective metrics under evaluation are provided after Gaussian noise, against the subjective scores collected under both test methods. In Figure 6.2, we demonstrate scatter plots of subjective against objective quality scores from the

6.1. Point cloud angular similarity

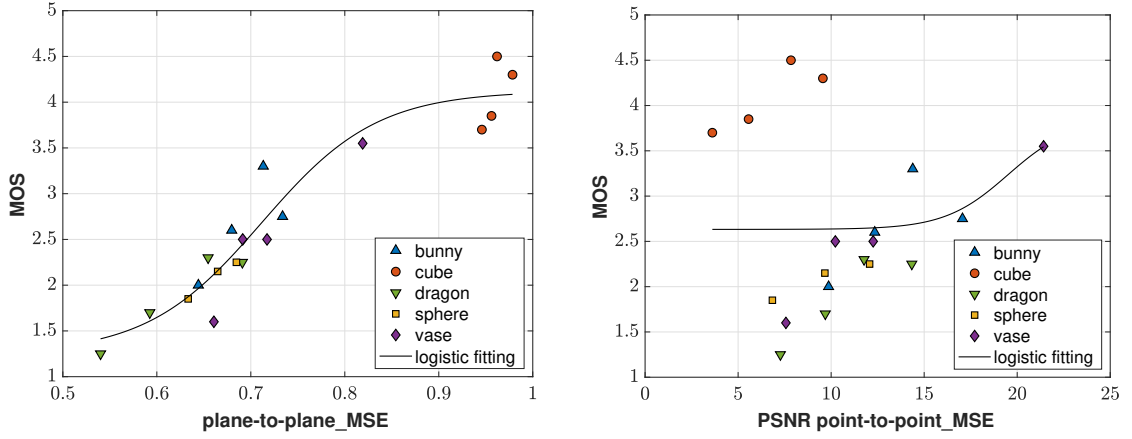


Figure 6.3 – G-PCD: Subjective against objective scores from the best-performing proposed (left) and anchor (right) quality metrics under Octree-pruning, using the ACR test method.

Table 6.2 – G-PCD: Performance indexes of objective quality metrics under Octree-pruning, for both test methods.

	ACR test method				DSIS test method			
	PLCC	SROCC	RMSE	OR	PLCC	SROCC	RMSE	OR
plane-to-plane_AVG	0.954	0.935	0.273	0.150	0.896	0.903	0.423	0.300
plane-to-plane_MSE	0.955	0.940	0.272	0.150	0.894	0.904	0.427	0.350
point-to-point_MSE	0.431	-0.024	0.826	0.550	0.193	-0.178	0.937	0.700
point-to-plane_MSE	0.618	0.029	0.720	0.500	0.386	-0.118	0.892	0.700
PSNR point-to-point_MSE	0.224	0.141	0.891	0.650	0.284	0.276	0.912	0.500
PSNR point-to-plane_MSE	0.614	0.110	0.724	0.421	0.415	-0.055	0.861	0.579

best-performing proposed and anchor algorithms, using the test method that revealed the highest correlation (i.e., ACR). Regarding the anchors, the PSNR point-to-point with MSE was found to outperform the alternatives using both the ACR and the DSIS test methods, whereas for the plane-to-plane variants, the MSE is marginally better.

Based on our analysis, strong correlation between objective and subjective scores can be observed in the presence of Gaussian noise, for both the anchor and the proposed objective quality metrics. Provided that the anchors measure geometric distances of closest points between the original and the distorted models, by increasing the standard deviation of the noise, the objective scores naturally worsen. The subjects were able to recognize such distortions and identify the amount of noise introduced by the level of points' displacement. Notably, the proposed metric achieves comparable performance, albeit the displacement of points typically leads to lower quality in the normal estimation process. It should be accounted that the selected normal estimation algorithm is generally considered as robust against noise.

In Table 6.2 and Figure 6.3, performance indexes and scatter plots are depicted after Octree-pruning. Based on our results, the correlation between subjective and objective scores is

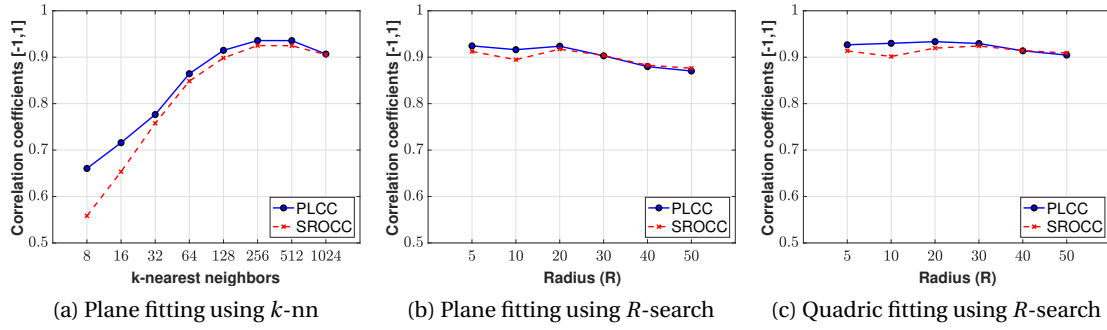


Figure 6.4 – J-PCED2: Performance indexes PLCC and SROCC of plane-to-plane metric, per normal estimation algorithm and configuration.

poor for every anchor metric. In general, this type of degradation leads to elimination of high frequency components and the perception of visual artifacts in the form of structural loss. Thus, the visual quality of point clouds with high curvature values and irregular topology is more severely impacted, whereas the structure of low curvature geometry models with regular geometry is not significantly affected, thus, leading to less perceptible visual degradations.

The anchor metrics, despite capturing position errors after pruning, do not consider local surface properties. Specifically, point-to-point metrics assign the same error value to a deviation of a point from the original position, independently of the underlying shape. Point-to-plane metrics assign different errors based on the direction of displacement of a point; that is, if a point deviates along the tangent plane perpendicular to the reference normal vector, no error occurs. However, in the corner case of a regular vertical displacement of a grid of points, high error values will be attained, while perceptual quality is not really affected by such distortions. This can explain the low prediction power in contents such as *cube*. On the contrary, plane-to-plane metrics rely on similarity between surface approximations, which qualifies them as better to capture such degradations, as proven by the higher correlation.

Performance evaluation on J-PCED2

In this data set we initiate by evaluating the performance of the plane-to-plane metric, subject to the selected normal estimation algorithms and neighborhood sizes. In particular, the PLCC and the SROCC correlation coefficients that were computed for each testing case, are depicted in Figure 6.4. It can be observed that when using plane fitting with k -nn, the performance of the metric is improving as the number of neighbors is increasing up to $k = 256$, whereas it remains almost the same for $k = 512$ and decreases for $k = 1024$. When using plane fitting with range search, the indexes are stable and high for R up to 20, above which they slowly decay. A similar trend is observed for quadric fitting, with slightly better performance and fewer deviations of the PLCC and SROCC indexes across the tested ranges.

Evidently, the prediction power of the plane-to-plane metric varies across different normal

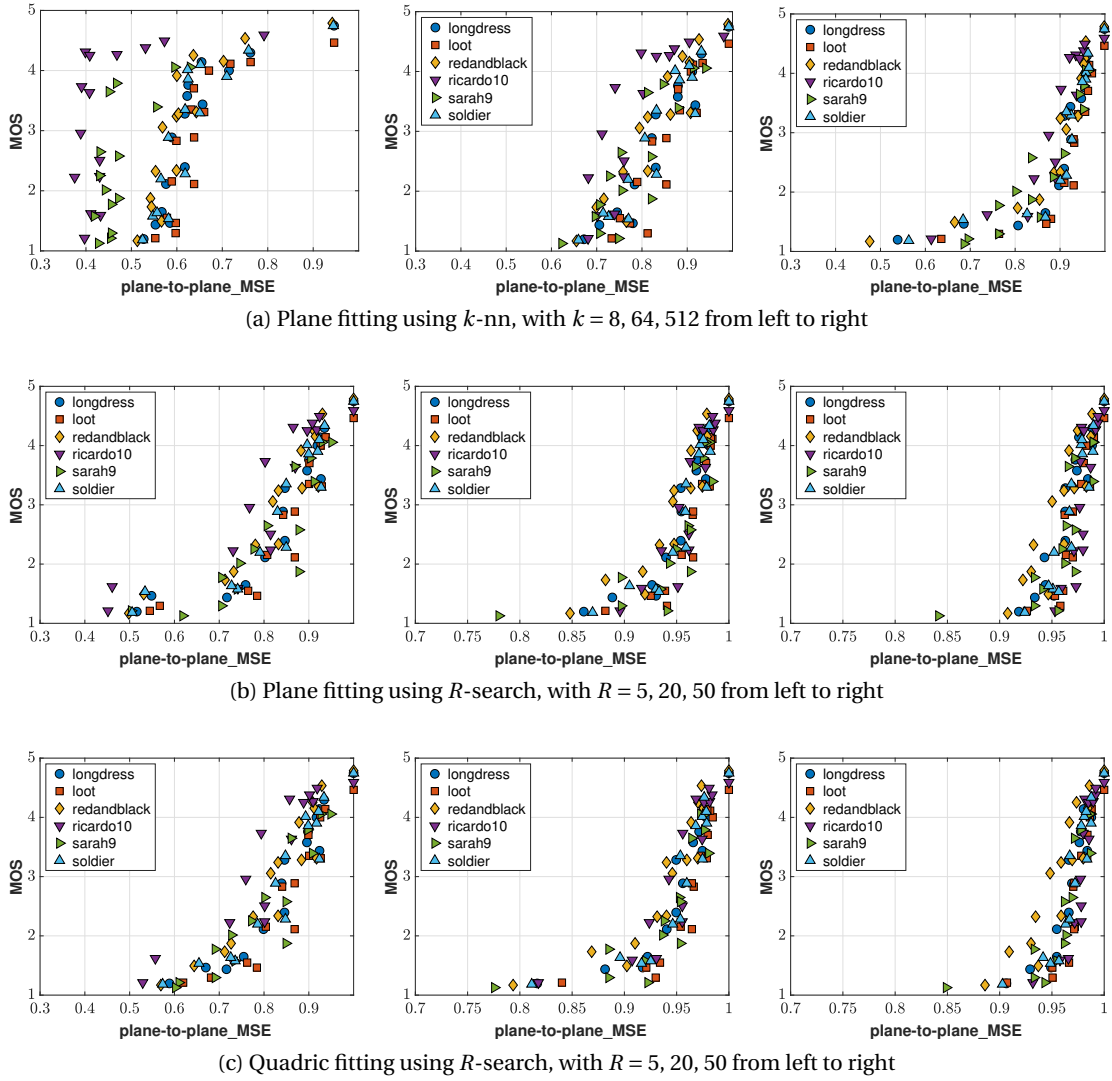


Figure 6.5 – J-PCED2: Subjective against objective scores from plane-to-plane metric, per normal estimation algorithm and configuration.

estimation algorithms and settings. This indicates the importance for a careful selection of a normal estimation methodology, prior to the computation of objective scores. Based on our results, we can conclude that the range search-based algorithms are substantially more robust to the selection of a neighborhood size, among the examined sets.

In Figure 6.5, scatter plots using 3 indicative configurations of neighborhood sizes (small, mid and large) per normal estimation algorithm are illustrated, in order to visually interpret the prediction accuracy of the proposed metric and the impact of normal estimation on the obtained quality scores. Based on the plots, it can be confirmed that plane and quadric fitting with range search lead to very similar result, as suggested earlier. Using the k -nn algorithm for neighborhood formulation, poor generalization capabilities might be observed

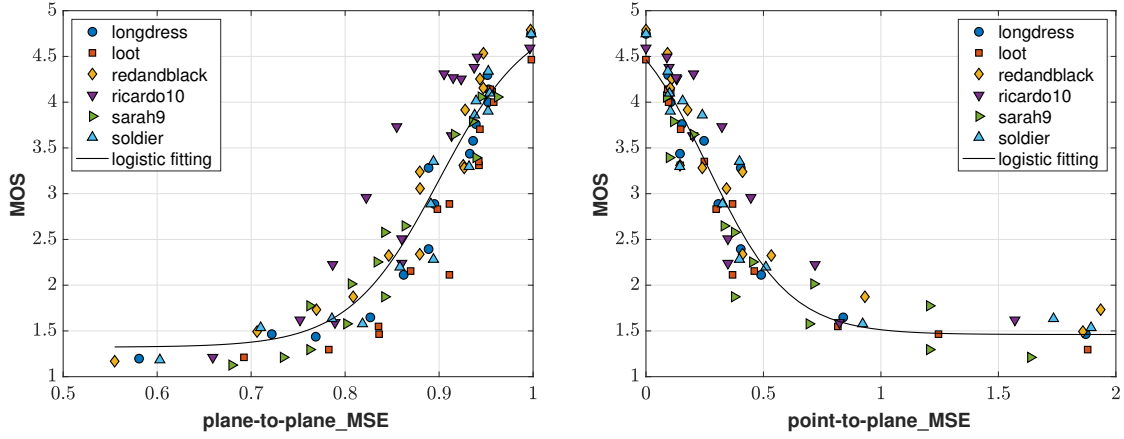


Figure 6.6 – J-PCED2: Subjective against objective scores from the best-performing configuration (i.e., plane fitting with $k = 256$) of the proposed (left) and anchor (right) quality metrics. The right plot is zoomed-in to provide a more informative view, capturing the majority of stimuli.

Table 6.3 – J-PCED2: Performance indexes of objective quality metrics. For plane-to-plane, the best-performing configuration per normal estimation algorithm is reported, using the following notation: [fitting surface, neighborhood configuration].

	PLCC	SROCC	RMSE	OR
plane-to-plane_MSE [plane, $k = 256$]	0.936	0.925	0.403	0.589
plane-to-plane_MSE [plane, $R = 20$]	0.924	0.917	0.438	0.656
plane-to-plane_MSE [quadric, $R = 30$]	0.930	0.924	0.421	0.667
point-to-point_MSE	0.947	0.935	0.369	0.667
point-to-plane_MSE	0.958	0.954	0.327	0.589
PSNR point-to-point_MSE	0.869	0.855	0.540	0.753
PSNR point-to-plane_MSE	0.911	0.915	0.449	0.612
PSNR_Y	0.888	0.893	0.526	0.689

(i.e., across different contents), especially when a small k is used. As the neighborhood size is increasing, such deviations are narrowed. Moreover, the spanning-range of the similarity scores is decreasing as the neighborhoods are enlarged using both k -nn and range search variants, which is reasonable if we consider that the estimated surfaces are getting smoother. Thus, less differences and higher similarity scores are obtained.

In Table 6.3, indexes for the best-performing configurations of the proposed metric are reported, per normal estimation algorithm. Moreover, the prediction accuracy of the anchors is indicated for comparison purposes. As can be seen, the point cloud angular similarity metric with plane fitting and $k = 256$ attains the best predictions, with marginal differences when compared to quadric fitting with $R = 30$. Under the best-performing normal estimation settings, the proposed method is found to be superior than the PSNR-based metrics, while

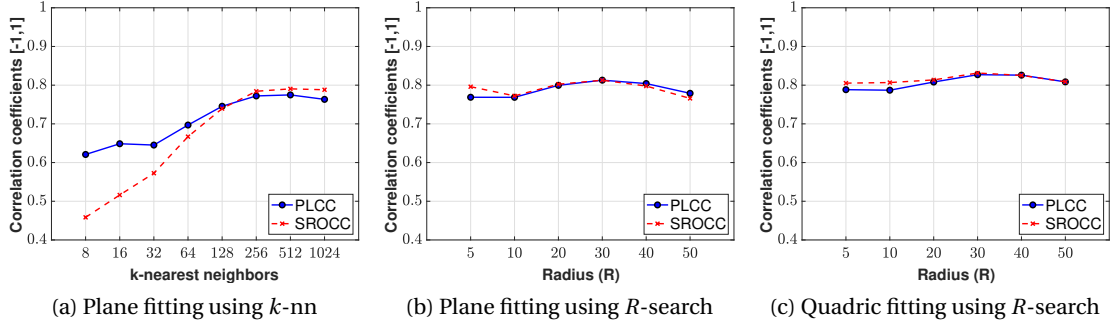


Figure 6.7 – M-PCCD: Performance indexes PLCC and SROCC of plane-to-plane metric, per normal estimation algorithm and configuration.

closely following the non-PSNR geometric predictors that achieve the highest performance. It should be remarked that the prediction power of the plane-to-plane remains competitive for the majority of tested neighborhoods, when employing the range search variants.

In Figure 6.6, scatter plots with subjective against objective quality scores are provided, as computed from the most efficient anchor metric, and the plane-to-plane algorithm using the highest-performing normal estimation configuration. As can be observed both methods attain a good linear and monotonic relationship.

Note that this data set consists of contents with very similar geometry, since the majority of them depict humans, while at higher degradation levels, both the geometry and the color attributes are simultaneously encoded at lower qualities. This explains the high accuracy that is achieved by geometry-only predictors. The lower performance of the PSNR-based geometric metrics can be explained by the presence of *sarah9*, whose voxel depth is lower (i.e., 9 bits) than the rest of the contents (i.e., 10 bits). Provided that the voxel resolution is set as the peak signal in the computation of PSNR, the corresponding objective scores are mapped to a shifted range with respect to the other contents, without accurately reflecting the respective differences in subjective opinions.

Performance evaluation on M-PCCD

The PLCC and the SROCC indexes of the plane-to-plane metric achieved for each neighborhood size and normal estimation algorithm over M-PCCD are reported in Figure 6.7. In principle, our results are very similar to the ones obtained using J-PCED2, with the only exception that the performance is lower in this case. However, this is a general tendency for all the metrics that were examined, as shown in Table 6.4. The decreased overall performance can be explained, firstly, from the more diverse type of contents (i.e., human figures and inanimate objects) and, secondly, by the larger number of stimuli and codecs that were evaluated, making this data set a more challenging benchmarking set-up. Based on Figure 6.7, it can be again remarked the lower sensitivity of the plane-to-plane metric in the selection of a neighborhood

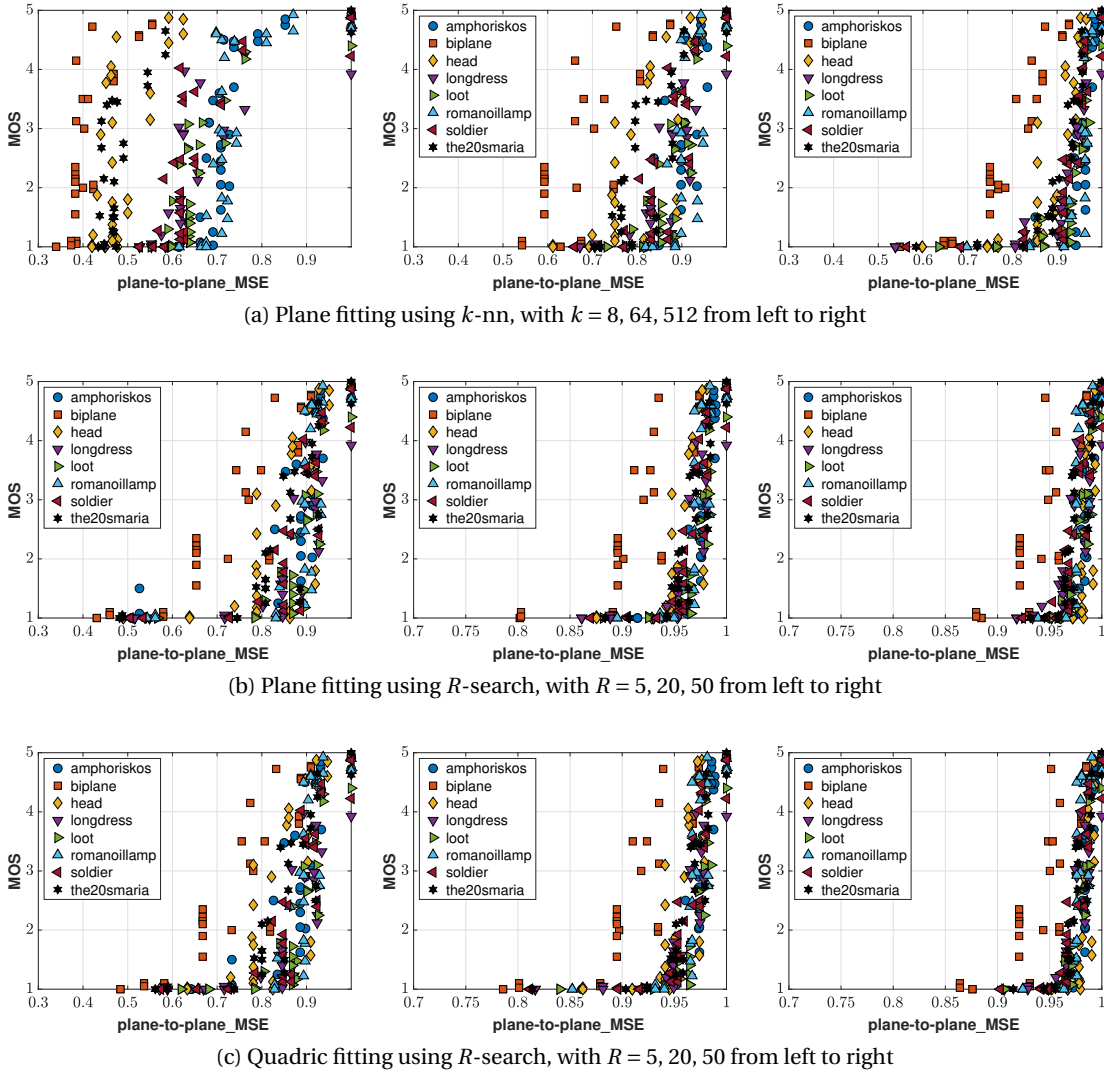


Figure 6.8 – M-PCCD: Subjective against objective scores from plane-to-plane metric, per normal estimation algorithm and configuration.

size for normal estimation under the range search variants, which perform very similarly.

In Figure 6.8, scatter plots with subjective scores against plane-to-plane predictions, under 3 representative configurations (small, mid and large neighborhood sizes) per normal estimation algorithm are depicted. In principle, our results coincide with earlier observations made on the J-PCED2 data set. We note that *head* and, especially, *biplane* contents behave as outliers, having a negative impact in the overall performance. Their irregular topology due to acquisition noise influences the normal estimation computations, and that explains the deviations that are observed in the predicted scores with respect to the rest of the contents. One potential solution would be to consider larger neighborhoods for such noisier contents, in order to enforce smoother surface approximations and reduce differences with respect to predicted scores for contents with more regular geometry.

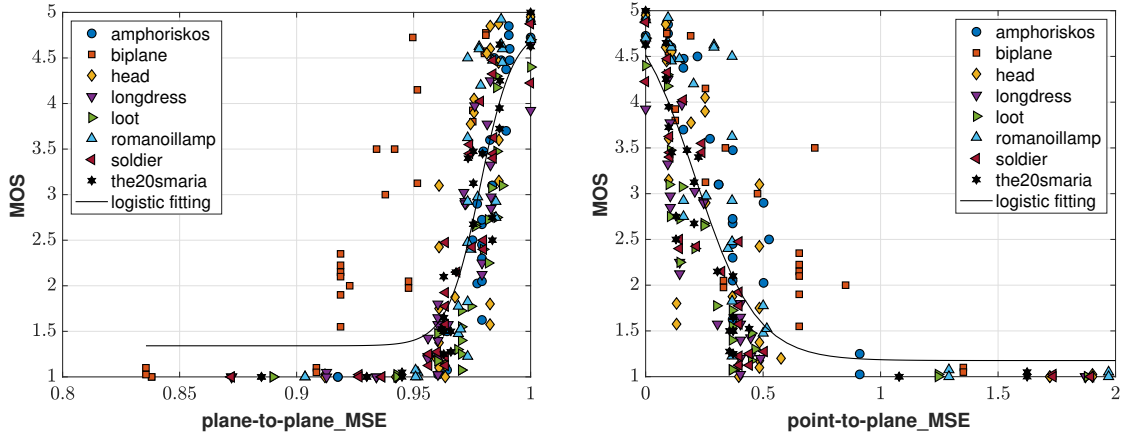


Figure 6.9 – M-PCCD: Subjective against objective scores from the best-performing configuration (i.e., quadric fitting with $R = 30$) of the proposed (left) and anchor (right) quality metrics. The right plot is zoomed-in to provide a more informative view, capturing the majority of stimuli.

Table 6.4 – M-PCCD: Performance indexes of objective quality metrics. For plane-to-plane, the best-performing configuration per normal estimation algorithm is reported, using the following notation: [fitting surface, neighborhood configuration].

	PLCC	SROCC	RMSE	OR
plane-to-plane_MSE [plane, $k = 512$]	0.775	0.791	0.862	0.802
plane-to-plane_MSE [plane, $R = 30$]	0.813	0.813	0.794	0.832
plane-to-plane_MSE [quadric, $R = 30$]	0.827	0.831	0.766	0.819
point-to-point_MSE	0.845	0.868	0.728	0.841
point-to-plane_MSE	0.858	0.884	0.700	0.832
PSNR point-to-point_MSE	0.720	0.759	0.885	0.819
PSNR point-to-plane_MSE	0.756	0.807	0.834	0.852
PSNR_Y	0.671	0.662	1.011	0.871

In Table 6.4, the coefficients of the best-performing configurations per normal estimation algorithm are reported for the proposed method, together with performance indexes for anchor metrics on this data set. As can be seen, the quadric fitting with $R = 30$ is identified as the best normal estimation setting. When compared to the anchors, the performance of the plane-to-plane using best-performing configurations is superior than the PSNR-based geometric and color predictors, and closely follows the point-to-point and point-to-plane with MSE. Analogously to the J-PCED2 data set, the range search variants for normal estimation are competitive independently of the neighborhood size, among the examined cases. Moreover, the performance of the PSNR-based geometric metrics was found to be negatively impacted by the presence of *head*, which is of lower voxel resolution with respect to the rest of the contents (i.e., 9 and 10 voxel bit-depths, respectively.) Finally, it is noteworthy that the poor results of

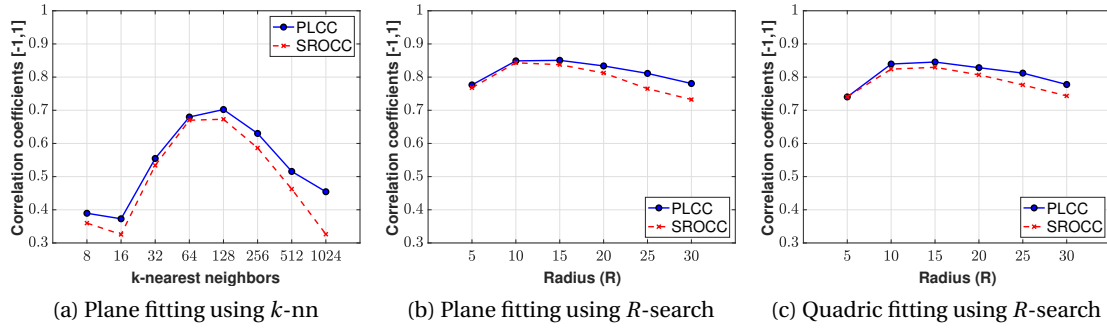


Figure 6.10 – IRPC: Performance indexes PLCC and SROCC of plane-to-plane metric, per normal estimation algorithm and configuration.

the PSNR_Y algorithm arise from its inefficiency to generalize predictions across different contents.

In Figure 6.9, scatter plots of subjective against objective scores from the best anchor method, and the plane-to-plane metric under the best-performing normal estimation settings, are illustrated. It is evident that both approaches have limitations, most apparently the inability of the plane-to-plane to adequately capture perceived distortions on the *biplane* content, as mentioned earlier.

The performance drop in this data set can be justified by the different nature of artifacts from the radically different encoding schemes, combined with the diverse topology of the contents. Moreover, it should be accounted that the models are colored, and as such they were subjectively evaluated. The fact that both metrics ignore color has an immediate negative impact on their prediction accuracy in this data set, considering that some of the stimuli have identical geometry and different color information. The latter is observed when using the same geometry and a different color encoding module that are part of the G-PCC test model.

We refer to chapter 8 for a performance evaluation study of the state-of-the-art objective quality metrics over this data set.

Performance evaluation on IRPC

The PLCC and the SROCC indexes that are computed for each neighborhood size and normal estimation algorithm using the IRPC data set, are illustrated in Figure 6.10. According to our results, the performance of the metric when using the k -nn algorithm is substantially lower with respect to the range search counterparts. In particular, despite an increase in performance as the neighborhoods are enlarging, the prediction remains at low accuracy levels. Using the range search algorithms, the coefficients are similarly improving from low- to mid-radii, where they start decaying. In general, an overall lower performance is observed, with respect to the previous data sets. The composition of contents spanning from small-scale

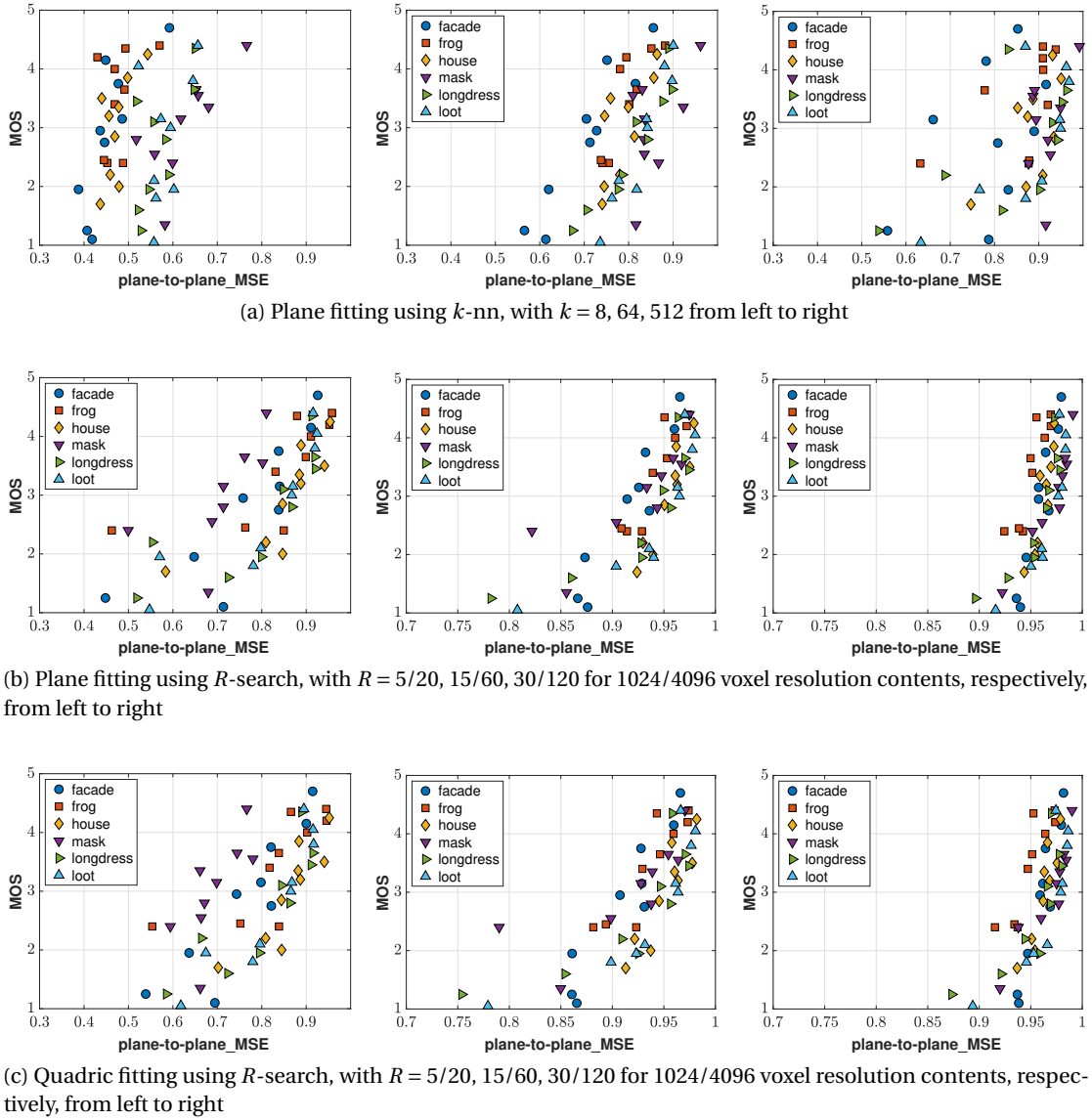


Figure 6.11 – IRPC: Subjective against objective scores from plane-to-plane metric, per normal estimation algorithm and configuration.

objects to large-scale buildings with varying levels of acquisition noise and missing regions, makes this a challenging set-up, which is reflected to the deteriorated prediction power of all metrics, as reported in Table 6.5.

Scatter plots of subjective against objective quality scores from the proposed metric are presented in Figure 6.11, under a small, mid and large neighborhood size for the selected normal estimation algorithms. It is observed that the performance of the k -nn is remarkably worse, and particularly when using small or large k 's. Yet, when using range search, similar general trends with respect to our previous analysis are noted, with plane-to-plane scores spanning at a narrower range as the neighborhood is increasing, and achieving good linear

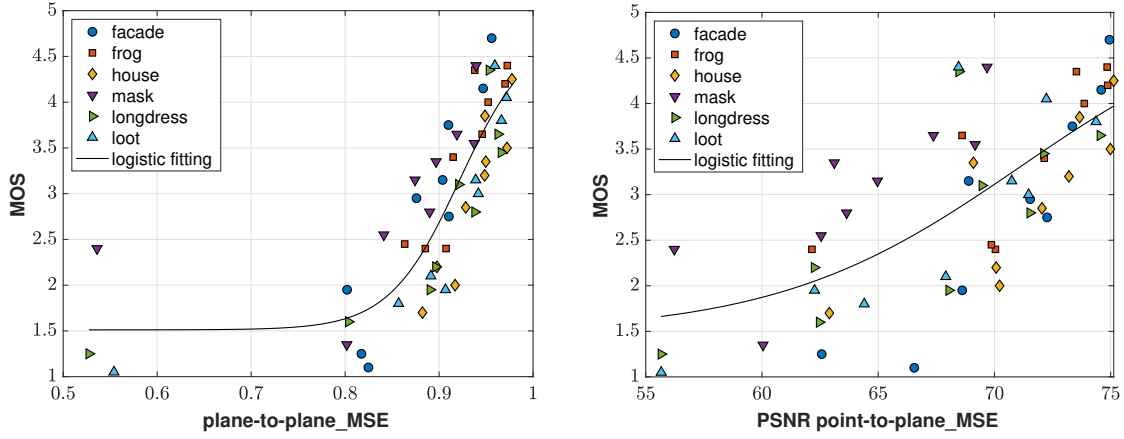


Figure 6.12 – IRPC: Subjective against objective scores from the best-performing configuration (i.e., linear fitting with $R = 10/40$ for 10-bit and 12-bit contents, respectively) of the proposed (left) and anchor (right) quality metrics.

Table 6.5 – IRPC: Performance indexes of objective quality metrics. For plane-to-plane, the best-performing configuration per normal estimation algorithm is reported, using the following notation: [fitting surface, neighborhood configuration].

	PLCC	SROCC	RMSE	OR
plane-to-plane_MSE [plane, $k = 128$]	0.702	0.673	0.711	-
plane-to-plane_MSE [plane, $R = 10/40$]	0.849	0.843	0.527	-
plane-to-plane_MSE [quadric, $R = 15/60$]	0.845	0.830	0.533	-
point-to-point_MSE	0.460	0.319	0.886	-
point-to-plane_MSE	0.537	0.428	0.842	-
PSNR point-to-point_MSE	0.668	0.647	0.743	-
PSNR point-to-plane_MSE	0.724	0.704	0.689	-

and monotonic relationships under certain configurations.

The best-performing coefficients of the anchor and the plane-to-plane metric, per normal estimation algorithm, are reported in Table 6.5. As shown, estimating the normals using range search leads to a prediction accuracy that outperforms the k -nn alternatives, under all examined neighborhood sizes. The best performance is attained using linear fitting with $R = 10/40$ for 10-bit and 12-bit contents, respectively, with marginal differences when compared to the best-performing configuration under quadric fitting. In this data set, the proposed metric is found to outperform the anchor methods with substantial differences. Recall that the subjective scores that serve as the ground-truth in our analysis, reflect the perceived quality of the stimuli as rated in the absence of color.

Finally, in Figure 6.12, scatter plots using the best normal estimation configuration for the proposed metric and the highest-performing anchor predictor are provided, showcasing the

superior performance of the former.

It is noteworthy that the best-performing metric reported in the literature is given in (Javaheri et al., 2020a), with PLCC = 0.801 and SROCC = 0.777, using the point-to-plane with Hausdorff over the 99% of the ranked distances, and min pooling to obtain a symmetric error.

6.1.4 Discussion

According to the performance analysis reported in the previous section, the plane-to-plane competes, or outperforms well-established solutions, provided a good configuration of the selected normal estimation algorithm. The neighborhood size over which the normals are estimated may act as a regularizer, which can be adjusted per content in a data set to enhance the generalization capabilities of the metric. Our observations suggest that it is often beneficial to employ larger neighborhood sizes with larger resolution point clouds, and contents with more irregular topology (i.e., acquisition noise).

Regarding the impact of the examined normal estimation algorithms on the performance of the plane-to-plane metric, the latter was found to be less sensitive to the neighborhood size selection and generally behaving better when using the range search variants with respect to k -nn. Note that the range search algorithm doesn't define the number of samples that form a neighborhood; rather, a spatial sub-space of equal volume is defined around every queried point. On the contrary, by using the k -nn alternative, the normal vector of a point is estimated across a fixed number of samples over a neighborhood that expands arbitrarily.

In the framework of the plane-to-plane computation, it should be additionally considered that the same normal estimation scheme is applied to pristine and impaired stimuli. When computing the angular similarity between a reference and a model under evaluation, the corresponding surface approximations are essentially compared. Using the same search radius for both models, leads to estimated normal vectors that reflect the same region of the content. On the contrary, using the k -nn approach, the normals will reflect surfaces that extend analogously to the local sparsity of the models. The former approach is intuitively more coherent, and its selection is further justified by the higher robustness it reveals across different neighborhood sizes.

The performance of the k -nn scheme with small k 's is rather unstable, which is further deteriorated by varying intrinsic geometric characteristics between the stimuli of a data set. As k is getting larger, smoother surfaces are obtained under any location data arrangement; thus, the angular similarity between a reference and a distorted model will be higher. This can explain why we observe lower prediction accuracy at small k values, whereas with larger k 's, the objective scores are higher and span in a narrower range.

As mentioned earlier, the range search variants were found to be robust against different neighborhood sizes, meaning, that good performance is attained for a fairly large selection of radii. The latter might seem counter-intuitive considering our findings reported in annex C,

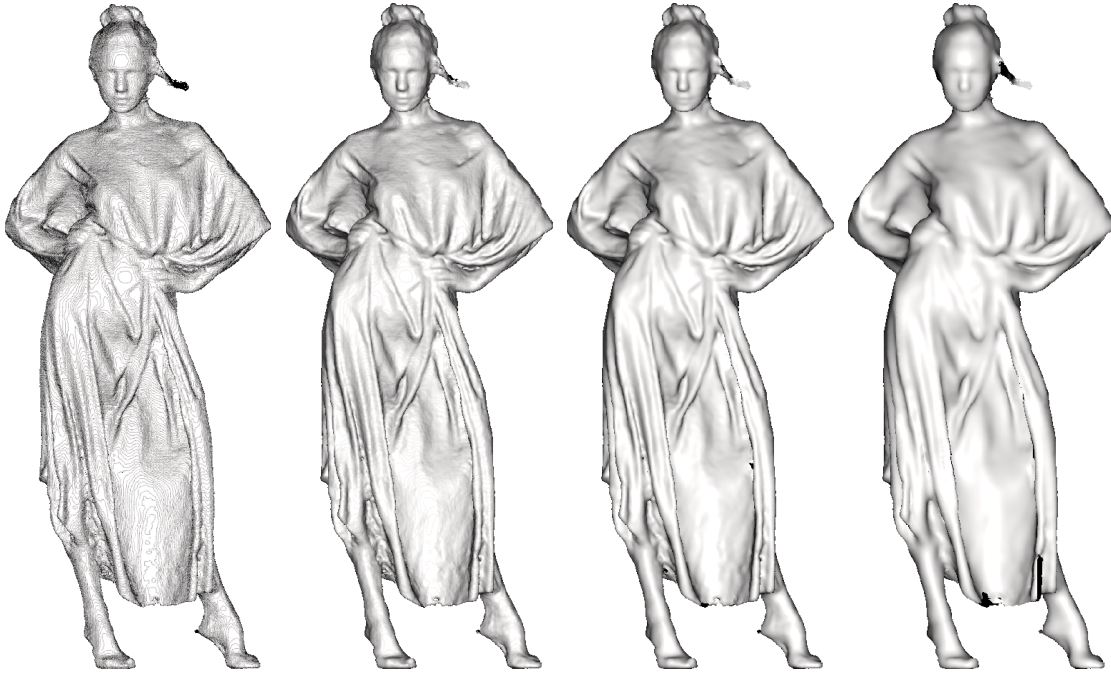


Figure 6.13 – Visualization of the reference *longdress* with point shading. The normal vectors are estimated with plane fitting and $k = 8, 32, 128$ and 512 from left to right.

where we study the accuracy of the same normal estimation algorithms. In particular, our results show that by exceeding a threshold for the searching radius, the normal estimation error is increasing rapidly. Conversely, when using larger radii for normal estimation, the prediction performance of the plane-to-plane metric is higher with respect to lower radii, where lower angular error would be expected (i.e., considering neighborhood sizes relative to the content resolution). Note that by computing the normals at larger neighborhoods, the approximated surfaces are becoming smoother, which simulates the application of a low-pass filter. Such an operation doesn't necessarily lead to more accurate results in terms of normal estimation error. Evidently, though, it may be better at capturing perceptual distortions, when properly configured. Notably, our conclusions indicate that optimizations to achieve lower normal estimation errors do not necessarily lead to better performance for the plane-to-plane metric.

A visualization example is provided in Figure 6.13, showing the *longdress* model rendered using point shading assisted by the estimated normal vectors under ambient light. In this illustration, a plane fitting with the k -nn approach is used, and results using $k = \{8, 32, 128, 512\}$ are indicatively presented. As we can see, at very small neighborhoods the estimated normal vectors wrongly reflect high-frequency geometric components, while as the size is increasing, the underlying surfaces are getting smoother. It should be noted that *longdress* is a model with approximately 800K points and mid-range geometric complexity. Thus, consulting Figure C.2a, a configuration of $k = 32$ is expected to provide relatively good results in terms of normal estimation error, which is confirmed by the illustration in Figure 6.13.

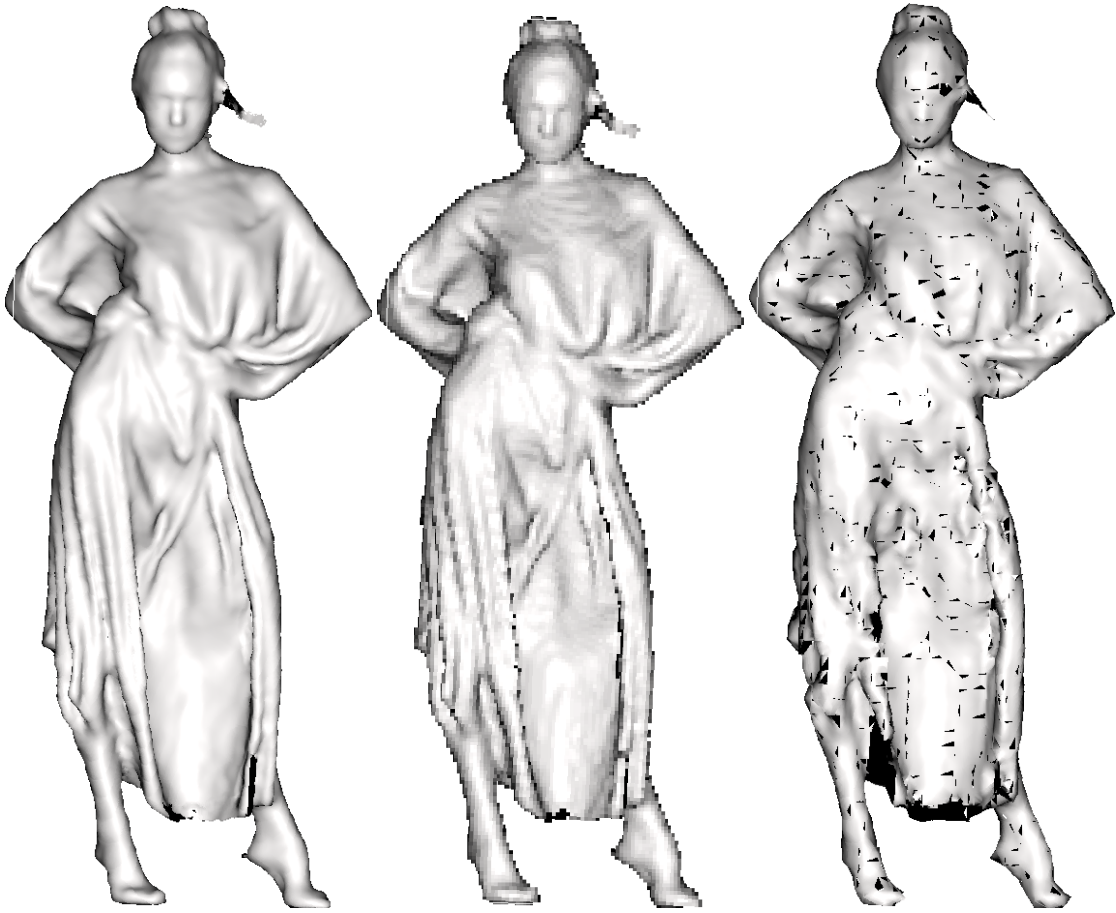


Figure 6.14 – Illustration of normal surface approximations. The geometry of the reference model *longdress*, a version after encoding with Octree at R02, and another version after encoding using TriSoup at R01 following the MPEG Common Test Conditions are displayed, from left to right. The normal vectors are estimated using plane fitting with range search and radius 10. The obtained plane-to-plane scores are 0.883 and 0.796, for the second and third stimulus, respectively.

Summarizing our findings, the plane-to-plane metric can provide good predictions of visual quality, given a good configuration of the algorithm to estimate normals. The range search variants were found to be better approaches for normal estimation in the context of the metric's computation, which is supported by both quantitative results and interpretable notions. In brief, they denote more robust solutions in regard to the neighborhood size selection, and they grant that the underlying surfaces under comparison approximate the same region of the content. In Figure 6.14, indicative degraded versions of the *longdress*, after normal estimation using range search with $R = 10$ are displayed, under point shading. The geometric distortions of the models are visible and accompanied by corresponding quality scores that are exported from the plane-to-plane metric.

6.2 Point cloud structural similarity

Point cloud geometry is fundamental for the topological definition of a 3D content. Thus, early works on the field of point cloud objective quality assessment explicitly focus on this type of information. Yet, more recent submissions aim at assessing distortions in color attributes and, often, incorporating them to a weighted sum together with predictions of geometric degradations. Textural information defines, to a large extent, the appearance of a content with the potentials of masking or enhancing underlying geometric distortions. Hence, it is of critical importance for the final judgement of an observer, regarding the perceptual quality of a content.

For both geometric and color-based quality prediction, there is a multitude of algorithms that has been proposed in the literature, ranging from simple distances to more elaborate methods that capture local deviations. So far, approaches that rely on local statistics were found to outperform the alternatives, denoting more promising directions for future developments. In particular, the pooling that is locally applied by such algorithms for the extraction of relevant features, may simulate processes that take place in the human visual system, thus, better quantifying the perceived quality of a content. This logic has been earlier exploited in 2-D imaging algorithms, laying the basis for some of the most successful objective quality metrics.

Under the same principle, our algorithm aims at capturing perceptual degradations based on the similarity of structural features that are locally extracted from a point cloud attribute (i.e., location, normal, curvature, color) of a pristine and an impaired model. The operating method shares similarities with the SSIM (Wang et al., 2004), thus, our metric is henceforth referred to as PointSSIM. In particular, relevant quantities are defined per attribute, and dispersion statistics are estimated from local populations. These measurements are obtained per point and denote our features that describe local properties of a point cloud attribute. After establishing associations based on nearest neighbors between points from the reference and the model under evaluation, the corresponding feature values are compared. A voxelization step can be optionally enabled prior to feature extraction, in order to reduce cross-content density differences and produce differently scaled versions of the models.

In this section, the proposed point cloud structural similarity metric is defined. The implementation of the metric is detailed, and the parameter space of the algorithm is reported. In regard to the prediction accuracy of the metric, (a) the efficiency of the point cloud attributes over which the algorithm is applied is explored, (b) a family of statistical dispersion estimators is evaluated, (c) the impact of the neighborhood size to compute the features is analysed, and (d) the effect of a voxelization step prior to feature computation is investigated. The performance analysis is conducted using available data sets with diverse characteristics and reveals best-performing attributes, features, and configurations.

From a different perspective, this study can also be regarded as an exploration of the applicability of the SSIM operating principle in a higher dimensional, irregular space (volumetric content), incorporating not only color, but also topological coherence among local regions.

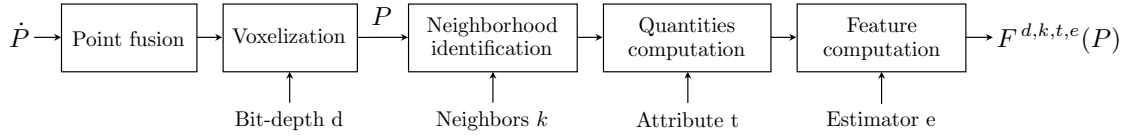


Figure 6.15 – Block diagram of the feature extraction steps. \dot{P} indicates an input point cloud, and P the point cloud at the output of the *voxelization* step. The features map $F^{d,k,t,e}(P)$ consists of structural features extracted from every point p that belongs to P , for a selected voxel bit-depth d , neighborhood size k , attribute t and dispersion estimator e .

6.2.1 Definition

The core idea behind measuring structural similarity is to capture local changes of visual information. Instead of comparing attributes directly associated with every point, we extract features from local neighborhoods, in order to capture local relationships and distribution variations among adjacent points. Specifically, we define relevant quantities that reflect properties of point cloud attributes, and for every point we estimate the statistical dispersion of their distribution in their local neighborhood. Such a statistical dispersion measurement forms a feature value for every point.

Structural features

A schematic overview of the feature extraction pipeline is illustrated in Figure 6.15, with each processing step described below.

Point fusion enables removal of duplicated coordinates and averaging of corresponding color values with identical locations across a point cloud. This step prevents enlisting points with the same position more than once during *neighborhood identification*. Moreover, redundant point correspondences between the model under evaluation and the reference for the computation of structural similarity scores (described below), are discarded.

Voxelization permits simulation of distant inspection, by down-scaling the resolution of a point cloud content. The voxelization is realized by quantizing the coordinates of a model and averaging the color between points with identical quantized positions. The voxel grid is defined by a target voxel bit-depth d , which can be manually adjusted. In our implementation, no clipping is applied on voxelized coordinates lying outside of the grid, in order to avoid introducing extra loss. Note that point reduction and color blending is applied only on models whose *intrinsic resolution* is larger than the target. *Intrinsic resolution* refers to the resolution of a content prior to a potential upscaling (i.e., mapping of a sparser point cloud to a higher resolution voxel grid). For models whose *intrinsic resolution* is smaller than the target, the color distribution remains unaltered, and the topology is upscaled without impacting the number of points.

Neighborhood identification defines the local region $N(p)$ for every point p that belongs to

a point cloud P . For this task, two are the most common approaches, namely, k -nn, and range search. In the first method, the set is extended until the specified number of points is reached, whereas in the second method, the set consists of points whose distance is smaller than the specified radius. Thus, in the former case, the range is adaptive in terms of size and the number of points is fixed, whereas in the latter case the range is fixed and the number of points can vary. In our implementation, the k nearest neighbors of every point are employed, thus, fixating the sample population of each neighborhood.

Quantities computation leads to sets that characterize local properties of a point cloud attribute. In our algorithm, we take into consideration quantities defined for point cloud locations, normal vectors, curvatures, and colors:

- A set of location-based quantities $X_p^{k,l}$ for a given point p are defined based on the Euclidean distances between this point and every point \tilde{p} that belongs to its neighborhood $N(p)$ with $|N| = k$, as per Equation 6.8. Note that $p, \tilde{p} \in \mathbb{R}^3$ by default, and $\|\bullet\|_2$ specifies the l_2 norm. The corresponding features assess the regularity of the local geometric structure.

$$X_p^{k,l} = \{\|p - \tilde{p}\|_2 \mid \tilde{p} \in N(p), \tilde{p} \neq p\} \quad (6.8)$$

- A set of normal-based quantities $X_p^{k,n}$ for given point p are defined based on the angular similarity between the normal vector of this point \vec{n}_p , and the normal vector $\vec{n}_{\tilde{p}}$ of each neighbor \tilde{p} that belongs to the neighborhood $N(p)$ with $|N| = k$, as per Equation 6.9. Note that the angular similarity formula is essentially applied on the unoriented vectors. The corresponding features evaluate the roughness of the local surface.

$$X_p^{k,n} = \left\{ \arccos \left(\frac{|\vec{n}_p \cdot \vec{n}_{\tilde{p}}|}{\|\vec{n}_p\| \|\vec{n}_{\tilde{p}}\|} \right) \mid \tilde{p} \in N(p), \tilde{p} \neq p \right\} \quad (6.9)$$

- A set of curvature-based quantities $X_p^{k,c}$ for given point p are defined by the curvature values $C(\tilde{p})$ of the points \tilde{p} that belong to the neighborhood $N(p)$ with $|N| = k$, as per Equation 6.10. The corresponding features, analogously to the normal-related quantities, evaluate the roughness of the local surface.

$$X_p^{k,c} = \{C(\tilde{p}) \mid \tilde{p} \in N(p)\} \quad (6.10)$$

- A set of color-based quantities $X_p^{k,y}$ for given point p are defined by the luminance values $Y(\tilde{p})$ of the points \tilde{p} that belong to the neighborhood $N(p)$ with $|N| = k$, as per Equation 6.11. The corresponding features estimate the local contrast, similarly to SSIM (Wang et al., 2004).

$$X_p^{k,y} = \{Y(\tilde{p}) \mid \tilde{p} \in N(p)\} \quad (6.11)$$

Feature computation enables the application of a dispersion estimator on a set $X_p^{k,t}$ for given point p and neighborhood size k , with $t \in \{l, n, c, y\}$ specifying the point cloud attribute.

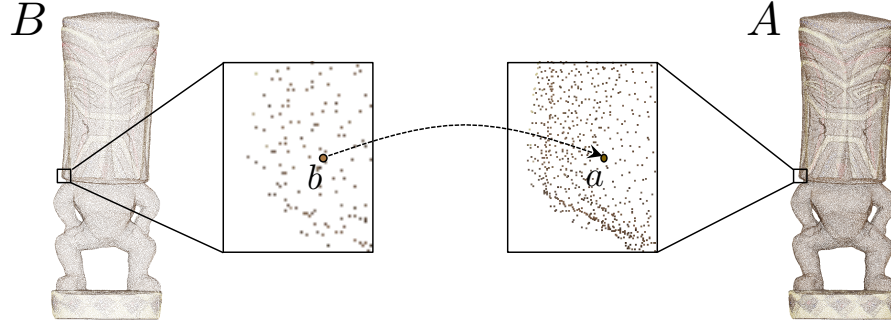


Figure 6.16 – Illustrative example of point association. The model A is set as the reference and the model B as under evaluation. The point a belongs to A and denotes the nearest neighbor of point b that belongs to B . The local neighborhoods $N(a)$ and $N(b)$ are defined around the former and the latter, respectively, in order to compute corresponding features.

Several dispersion estimators exist, and such measurements are often utilized to estimate scale parameters; that is, population parameters that indicate the spread of a distribution. In our implementation, an estimator e can be selected from the following pool: standard deviation (σ_X), variance (σ_X^2), mean absolute deviation (μAD_X), median absolute deviation (mAD_X), coefficient of variation (COV_X), and quartile coefficient of dispersion (QCD_X), using Equations 6.12-6.15 for the last four metrics, respectively

$$\mu AD_X = \mathbb{E}(X - \mu_X) \quad (6.12)$$

$$mAD_X = \mathbb{E}(X - m_X) \quad (6.13)$$

$$COV_X = \frac{\sigma_X}{\mu_X} \quad (6.14)$$

$$QCD_X = \frac{Q_X(3) - Q_X(1)}{Q_X(3) + Q_X(1)} \quad (6.15)$$

where $\mathbb{E}(\bullet)$ indicates expectation, μ_X the mean, m_X is the median, and $Q_X(i)$ denotes the i -th quartile of a set X .

Quality score

Using the aforementioned procedure, features are extracted from a reference and a model under evaluation and subsequently compared in pairs of associated points, in order to obtain a quality score. Let us assume that point cloud A is the pristine content and point cloud B is an impaired version, as obtained after the *voxelization* step, per Figure 6.15. By setting A as the reference, each point b that belongs to B is associated to the nearest reference point a

in the Euclidean space, in order to compute the structural similarity of adjacent topological regions, as presented in Figure 6.16.

An error value $E_{B,A}^{d,k,t,e}(b)$ is computed for point b as the relative difference between the corresponding feature values, using Equation 6.16

$$E_{B,A}^{d,k,t,e}(b) = \frac{|F^{d,k,t,e}(b) - F^{d,k,t,e}(a)|}{\max\{|F^{d,k,t,e}(b)|, |F^{d,k,t,e}(a)|\} + \varepsilon} \quad (6.16)$$

with ε expressing an arbitrarily small number to avoid undefined operations; in our simulations, we set ε equal to the machine rounding error for floating point numbers.

A structural similarity score $S_{B,A}^{d,k,t,e}(b)$ is obtained by taking the complement of 1 to this error value, as depicted in Equation 6.17.

$$S_{B,A}^{d,k,t,e}(b) = 1 - E_{B,A}^{d,k,t,e}(b) \quad (6.17)$$

A global degradation score $\text{PointSSIM}_{B,A}^{d,k,t,e}$ for the model under evaluation B with respect to A is estimated by applying a pooling method $\mathcal{P}(\bullet)$ on the individual structural similarity scores that are obtained for every $b \in B$, as indicated in Equation 6.18.

$$\text{PointSSIM}_{B,A}^{d,k,t,e} = \mathcal{P}(S_{B,A}^{d,k,t,e}(b)) \quad (6.18)$$

Indicative examples of pooling methods are given in Equation 6.19

$$\mathcal{P}(S_{B,A}^{d,k,t,e}(b)) = \frac{1}{|B|} \left(\sum_{b \in B} S_{B,A}^{d,k,t,e}(b)^q \right)^{\frac{1}{p}} \quad (6.19)$$

where $|B|$ indicates the cardinality of the point cloud under evaluation B , and $q, p \in \mathbb{R}_{0^+}$. More frequently, the mean or the MSE are employed for pooling.

Analogously, the global degradation score $\text{PointSSIM}_{A,B}^{d,k,t,e}$ is obtained, by setting B as the reference and A as the model under evaluation. The symmetric error $\text{PointSSIM}^{d,k,t,e}$ can be computed using Equation 6.20.

$$\text{PointSSIM}^{d,k,t,e} = \min \left\{ \text{PointSSIM}_{B,A}^{d,k,t,e}, \text{PointSSIM}_{A,B}^{d,k,t,e} \right\} \quad (6.20)$$

The description for the computation of the global degradation score using A as the reference is summarized in Algorithm 2. In an analogous way, the global degradation score using B as the reference is computed. The symmetric error can then be straightforwardly obtained.

Algorithm 2

-
- 1: For both models, fuse duplicated points
 - 2: For both models, voxelize at bit-depth d
 - 3: Set as reference A
 - 4: **for all** $a \in A$ **do**
 - 5: For neighbor size k , identify neighborhood $N(a)$
 - 6: For attribute t , compute set of quantities $X_a^{k,t}$
 - 7: For estimator e , extract feature $F^{d,k,t,e}(a)$
 - 8: **for all** $b \in B$ **do**
 - 9: For neighbor size k , identify neighborhood $N(b)$
 - 10: For attribute t , compute set of quantities $X_b^{k,t}$
 - 11: For estimator e , extract feature $F^{d,k,t,e}(b)$
 - 12: Identify a as the nearest neighbor of b in A
 - 13: Compute error $E_{B,A}^{d,k,t,e}(b)$
 - 14: Compute structural similarity $S_{B,A}^{d,k,t,e}(b)$
 - 15: Compute global degradation $\text{PointSSIM}_{B,A}^{d,k,t,e}$
-

6.2.2 Validation methodology**Data sets**

A total of 3 subjectively annotated data sets is recruited in order to evaluate the performance of the point cloud structural similarity metric. In particular, we employ the J-PCED2, M-PCCD and IRPC, which have been summarized in section 6.1.2. Regarding IRPC, we additionally use as ground truth subjective scores that were collected from the so-called *rcolor* session. In particular, the subjective evaluation of the stimuli was conducted using point-based rendering with color information that was obtained from the reference models through a re-coloring step. Further details can be found in the corresponding paper (Javaheri et al., 2019).

Computation of quality metrics

In our analysis, we explore the parameter space of the proposed metric. In particular, we let k take values from $\{6, 12, 24, 48\}$, in order to examine the impact of the neighborhood size over which a structural feature is computed. Moreover, to evaluate the effect of employing a different dispersion estimator, all the statistics described in section 6.2.1 are employed. Under every combination of neighborhood size and dispersion estimator, the performance of every attribute-based feature that is defined in section 6.2.1 is evaluated. Finally, to explore potential benefits by computing structural similarity scores on scaled models, voxelization is applied using several target voxel bit-depths on best-performing attribute-based features, per data set. Note that the same voxel bit-depth is always used for a reference and a model under evaluation. In order to compute a total degradation score for a model under evaluation, average pooling is applied. The above analysis is repeated using both the pristine and the impaired models as a

reference, and the symmetric error is additionally computed. Marginal improvements were identified when using the impaired models as a reference, thus, the corresponding objective quality scores are employed in our results.

For the computation of normal-based and curvature-based quantities at the execution of the PointSSIM metric, relevant attributes are estimated using quadric surface fitting, which is implemented following the algorithm described in (Meynet et al., 2020). In our implementation, the k nearest neighbors of each point are initially identified, with $k = 12$ being used in our simulations. A Principal Component Analysis (PCA) is issued to provide an orthonormal basis and a linear approximation of the local surface, which passes from the centroid of the neighborhood. A least-squares error quadratic fitting function is computed across the normal of the plane, after transferring the origin of the orthonormal basis from the centroid to the transformed point of focus. The normal vector in this new coordinate system is obtained by simply computing the gradient of the locally fitted quadric surface at that point. Then, the inverse transform brings the estimated normal vector back to the original coordinate system. Moreover, the mean curvature value at the point of focus is computed from the coefficients of the fitted quadric surface, as described in (Meynet et al., 2020).

For comparison purposes, the point-to-point with MSE, point-to-plane with MSE, the corresponding geometric PSNR variants, and, for colored stimuli, the color PSNR computed on the luminance channel are additionally evaluated, using the software ver. 0.13.5 (Tian et al., 2017c). Details regarding their execution are provided in section 6.1.2.

Benchmarking of quality metrics

To evaluate how well an objective metric is able to estimate perceptual quality, MOS computed from ratings of subjects that participate in an experiment are required and serve as ground truth. The objective quality scores are typically benchmarked after applying a regression model. In our case, the logistic function is used following the methodology described in section A.3. The PLCC, the SROCC, the RMSE, and the OR are computed to conclude on the linearity, monotonicity, and accuracy of the objective quality predictors, respectively.

6.2.3 Results

Performance evaluation on J-PCED2

In Figure 6.17, the performance of the PointSSIM is provided on J-PCED2 without enabling voxelization, and under every combination of neighborhood size, dispersion estimator, and attribute. In particular, each figure depicts the performance of features extracted from a particular attribute, for all estimators and neighborhood sizes. Correlation coefficients are displayed in the form of bars, with thick bars denoting the PLCC and thin bars indicating the SROCC index. They are grouped per estimator, which is indicated on the x-axis, and in each group, the four selected neighborhoods are displayed in an increasing order.

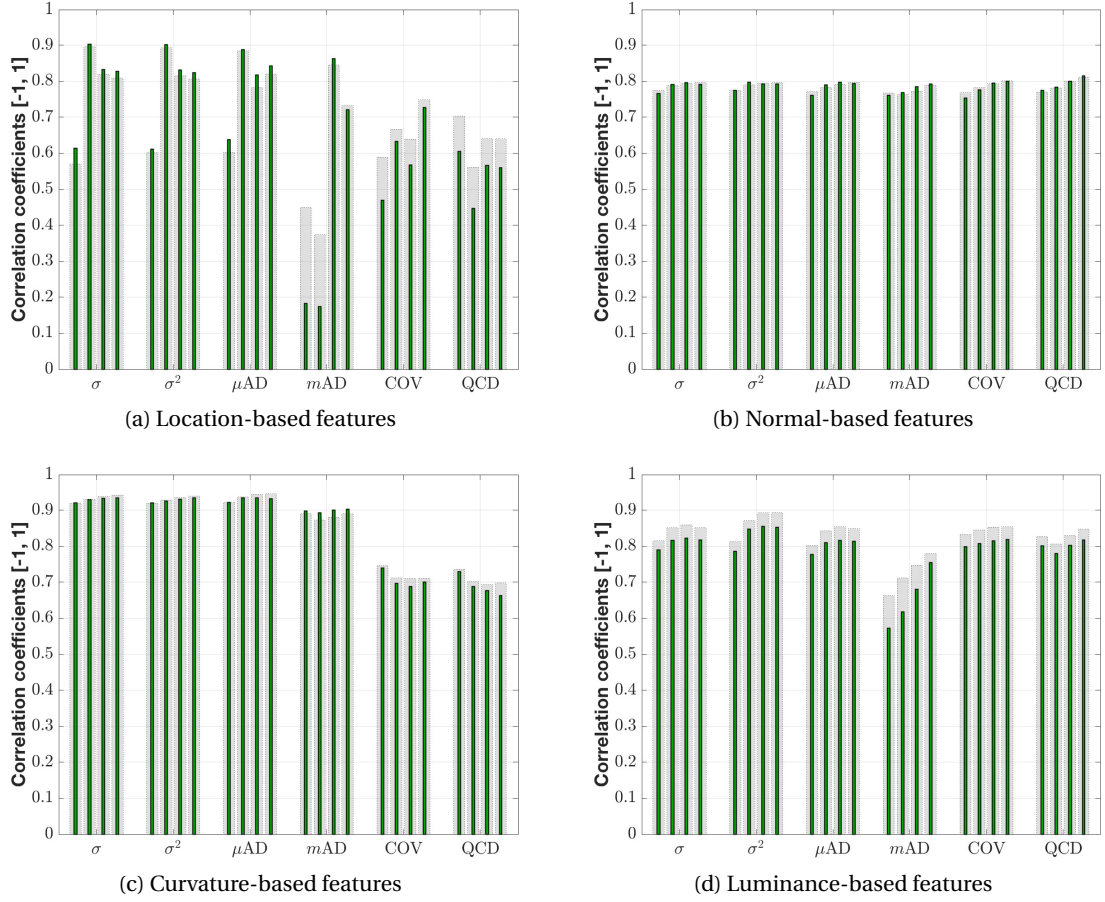


Figure 6.17 – J-PCED2: Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the neighborhood size is 6, 12, 24 and 48, from left to right.

Our results show that curvature-based features are better quality predictors in this data set, with objective scores achieving a highly linear and monotonous relationship with respect to subjective opinions. In principle, the performance of all estimators is very similar, with COV and QCD denoting sub-optimal solutions. Moreover, it is evident that the neighborhood size doesn't critically affect the predicted scores, although marginal improvements are brought with larger k 's.

Certain configurations using the location-based features are found to be the second best option in this data set. However, their instability in regard to the neighborhood size, suggests limited generalization capabilities. On the contrary, the luminance-based features, despite showing slightly lower performance, provide substantially higher consistency on this matter. Moreover, the majority of the estimators lead to good performance results, with mAD denoting the only exception with notable drops. Normal-based features also lead to fairly good results, with correlation coefficients improving as the neighborhoods are expanded, under any estimator.

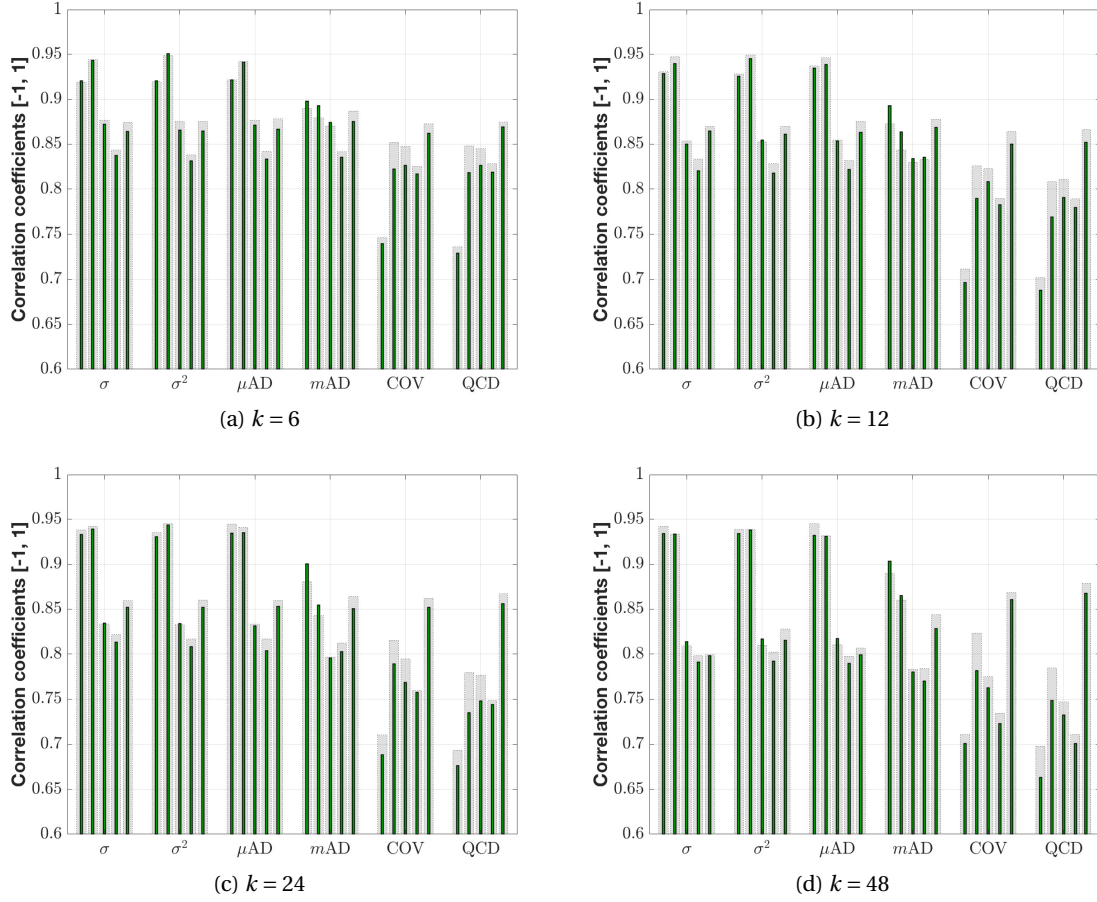


Figure 6.18 – J-PCED2: Curvature-based features. Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the corner left bar corresponds to no voxelization, whereas the rest of the bars correspond to voxel bit-depths equal to 9, 8, 7 and 6, from left to right.

In Figure 6.18, the PLCC and SROCC indexes for the curvature-based features are presented after voxelization at bit-depths d , starting at the lowest resolution present in the data set and progressively decreasing. In particular, J-PCED2 consists of stimuli with voxel resolution equal to 9 and 10 bit-depth. In our simulations, we employ $d = \{9, 8, 7, 6\}$, with $d = 9$ indicating that all stimuli are voxelized at a resolution of 512. The performance using the original stimuli is shown for comparison purposes.

Our results show that a voxel depth of 9, leads to better, or similar PLCC and SROCC values when compared to the no-voxelization case. Moreover, we observe that when voxelizing at 9 bits, as the neighborhood is increasing, the performance using these estimators is decreasing. This denotes an inverse relationship with respect to the one observed at the no-voxelization case (i.e., better performance at larger neighborhoods). This trend can be explained considering that a smaller voxel resolution implies a decrease in point count. Thus, the same

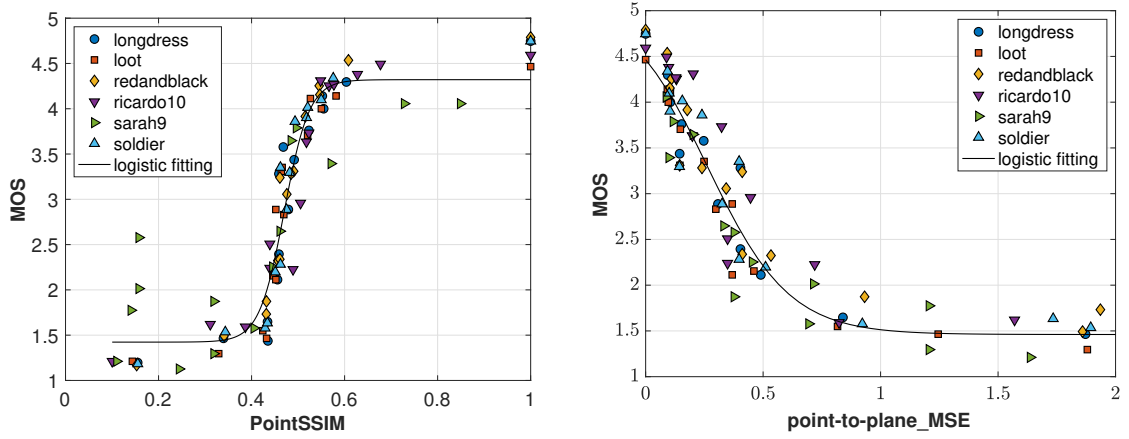


Figure 6.19 – J-PCED2: Subjective against objective scores from the best-performing configuration (i.e., curvature-based features, voxel depth of 9 bits, dispersion estimator σ^2 , neighborhood size of 6) of the proposed (left) and anchor (right) quality metrics. The right plot is zoomed-in to provide a more informative view, capturing the majority of stimuli.

Table 6.6 – J-PCED2: Performance indexes of objective quality metrics. For PointSSIM, the best-performing configuration is reported, using the following notation: [attribute, voxel depth, dispersion estimator, neighborhood size].

	PLCC	SROCC	RMSE	OR
PointSSIM [curvature, 9 bits, σ^2 , 6]	0.948	0.951	0.363	0.578
point-to-point_MSE	0.947	0.935	0.369	0.667
point-to-plane_MSE	0.958	0.954	0.327	0.589
PSNR point-to-point_MSE	0.869	0.855	0.540	0.753
PSNR point-to-plane_MSE	0.911	0.915	0.449	0.612
PSNR_Y	0.888	0.893	0.526	0.689

neighborhood size will correspond to a larger region of a content under evaluation. These results suggest that there is sweet spot for determining of the region over which the features are extracted. A voxelization below 9 bits leads to a performance decline, when considering the features with the best-performing estimators σ , σ^2 and μAD . This is reasonable if we consider that by excessive down-sampling of the models' topology, higher-resolution details are lost, while also sparser regions or missing points in the encoded models, are alleviated. On the other hand, the COV and QCD estimators perform better at lower voxel resolutions, independently of the neighborhood size.

In Table 6.6, performance indexes for the best-performing configuration of the proposed metric and the anchor algorithms are reported. As can be seen, the PointSSIM using curvature-based features that are extracted from voxelized stimuli at 9 bit-depth with σ^2 and $k = 6$, attains the best prediction, when compared to the alternative configurations of the metric. Under these settings, the performance of PointSSIM is high and very similar to the best-performing

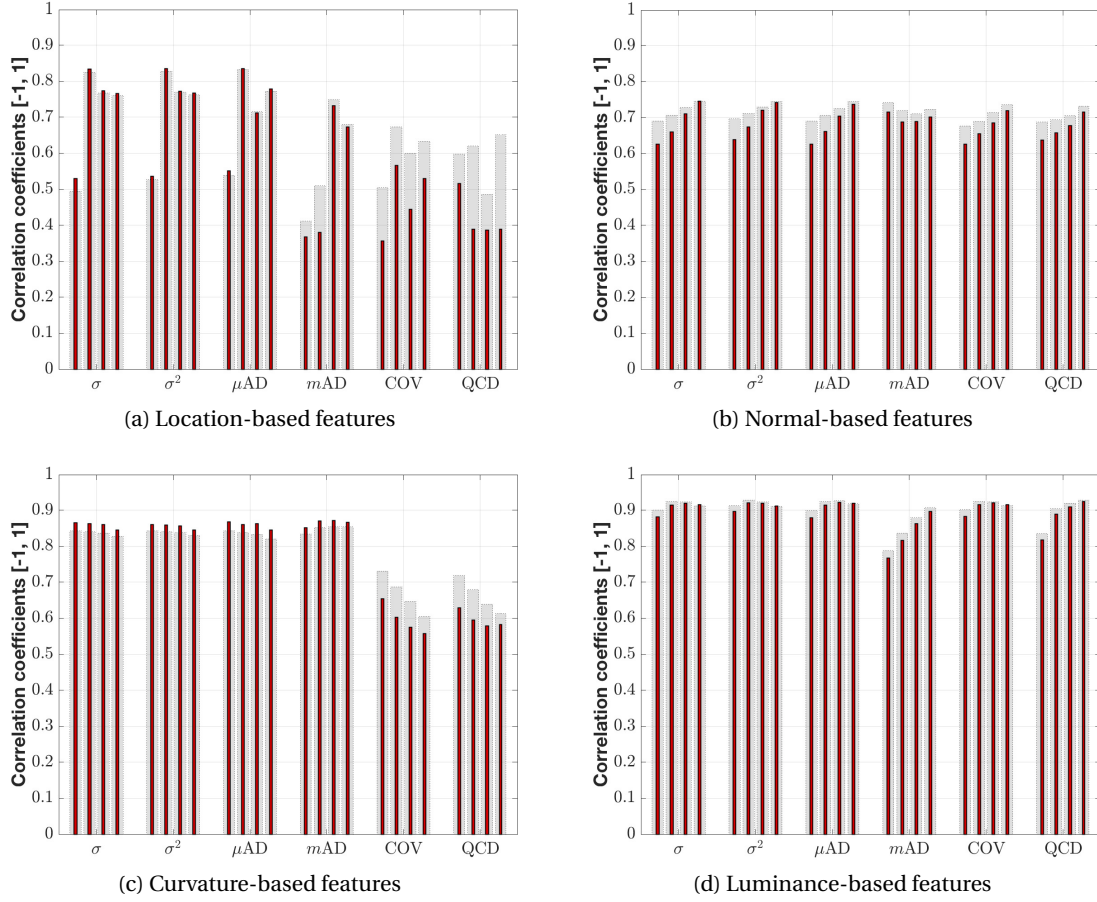


Figure 6.20 – M-PCCD: Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the neighborhood size is 6, 12, 24 and 48, from left to right.

point-to-plane with MSE. As reported in section 6.1.3, geometry-based predictors perform well on this data set due to the similarities in the topology of the contents.

Finally, in Figure 6.19, scatter plots with subjective against objective quality scores are provided, using the best-performing anchor method and PointSSIM configuration, visually confirming the high performance indexes. We remark that PointSSIM leads to very accurate results for all contents, with *sarah9* following a slightly different trend.

Performance evaluation on M-PCCD

In Figure 6.20, the performance of the PointSSIM is indicated on M-PCCD without enabling voxelization, and for every neighborhood size, dispersion estimator, and attribute under consideration, analogously to the previous section.

Our results show that luminance-based features are superior in this data set, in terms of

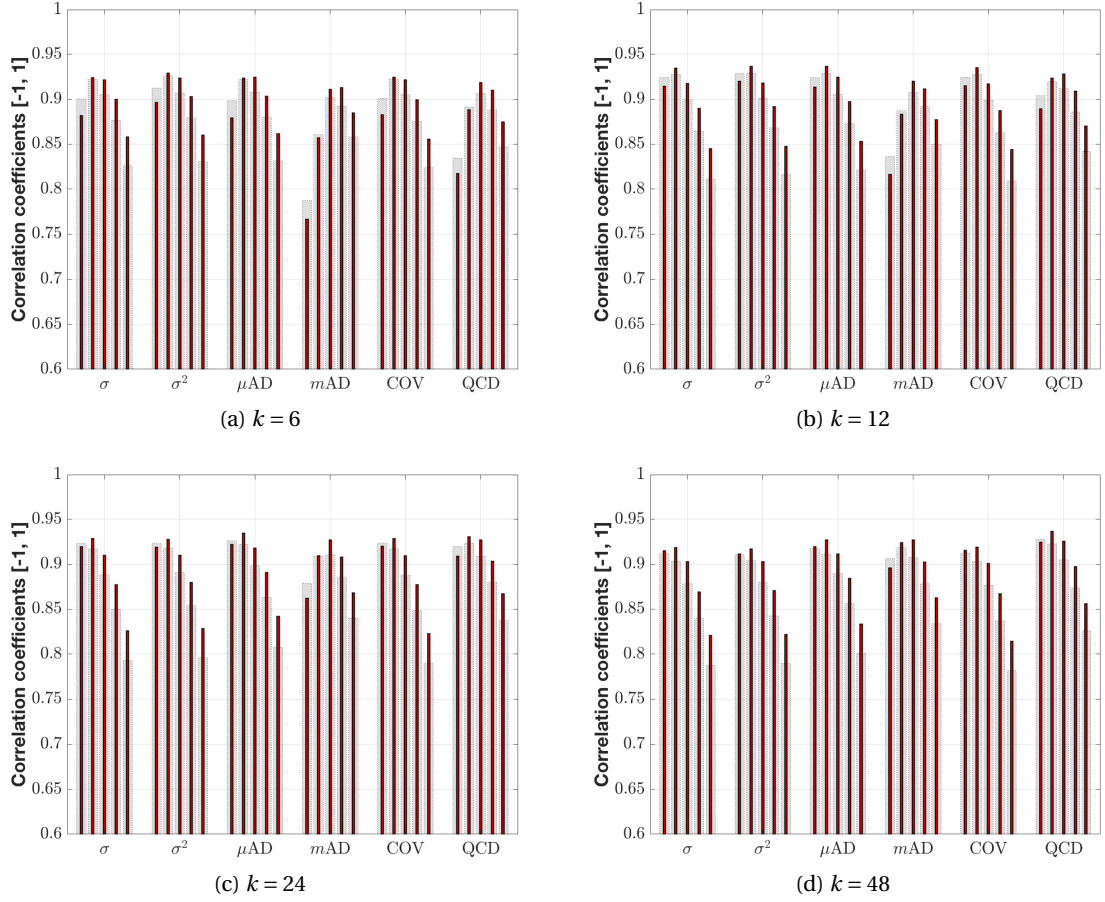


Figure 6.21 – M-PCCD: Luminance-based features. Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the corner left bar corresponds to no voxelization, whereas the rest of the bars correspond to voxel bit-depths equal to 9, 8, 7 and 6, from left to right.

prediction accuracy. In principle, the performance of all estimators is very similar, with mAD under-performing at smaller neighborhoods, although still achieving good results. We observe that the neighborhood size is not critical, albeit slightly better performance is obtained for the majority of the estimators in mid-ranges (i.e., k equal to 12 or 24).

The curvature-based features denote the second best-performing solution. For the dispersion estimators that work better, namely, σ , σ^2 , μAD and mAD , the number of neighbors k is also not crucial. Regarding location-based features, they are rather unstable with respect to the local neighborhood size, while the majority of normal-based features tend to improve as the neighborhoods are enlarging, similarly to the behavior observed in our analysis over J-PCED2.

In Figure 6.21, the PLCC and SROCC indexes for the curvature-based features are presented after voxelization at bit-depths d , starting at the lowest resolution present in the data set and progressively decreasing. In particular, M-PCCD consists of stimuli with voxel resolution equal

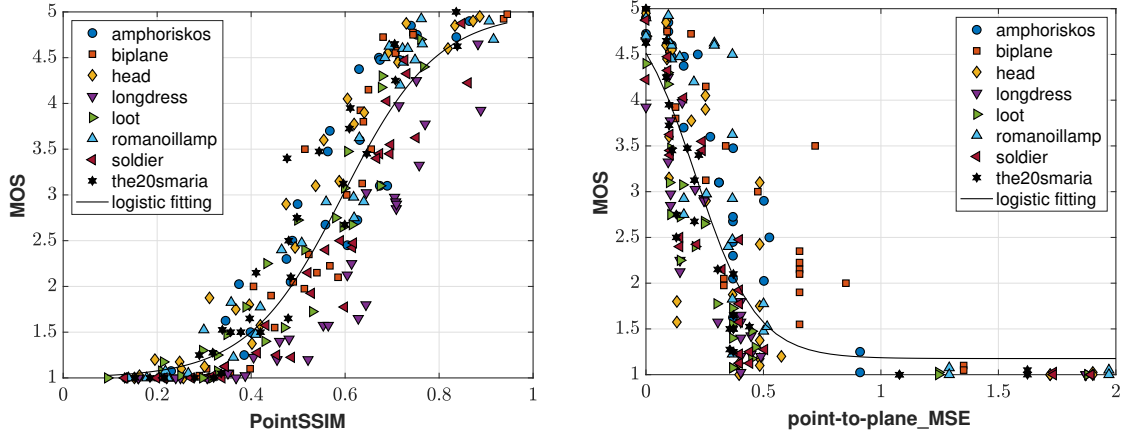


Figure 6.22 – M-PCCD: Subjective against objective scores from the best-performing configuration (i.e., luminance-based features, voxel depth of 9 bits, dispersion estimator σ^2 , neighborhood size of 12) of the proposed (left) and anchor (right) quality metrics. The right plot is zoomed-in to provide a more informative view, capturing the majority of stimuli.

Table 6.7 – M-PCCD: Performance indexes of objective quality metrics. For PointSSIM, the best-performing configuration is reported, using the following notation: [attribute, voxel depth, dispersion estimator, neighborhood size].

	PLCC	SROCC	RMSE	OR
PointSSIM [luminance, 9 bits, σ^2 , 12]	0.929	0.936	0.504	0.716
point-to-point_MSE	0.845	0.868	0.728	0.841
point-to-plane_MSE	0.858	0.884	0.700	0.832
PSNR point-to-point_MSE	0.720	0.759	0.885	0.819
PSNR point-to-plane_MSE	0.756	0.807	0.834	0.852
PSNR_Y	0.671	0.662	1.011	0.871

to 9 and 10 bit-depth. In our simulations, we employ $d = \{9, 8, 7, 6\}$, with $d = 9$ indicating that all stimuli are voxelized at a resolution of 512.

As mentioned earlier, for the original voxel resolution, a general increase in performance can be observed as the neighborhood size increases for all estimators excluding QCD, achieving a peak at $k = 12$ or 24, whereas for larger neighborhoods, performance starts decreasing. However, when the voxel resolution is set progressively lower, we observe that the best performance for a particular estimator is obtained with increasingly smaller neighborhoods, analogously to the trend observed on J-PCED2. Moreover, we note that for a given neighborhood size, the performance decays progressively at lower voxel resolutions, after reaching a peak. The deteriorated performance that is observed at lower voxel bit-depths (e.g., $d = 6, 7$), is justified by the increasing levels of blurring artifacts that appear due to voxelization, which are further enhanced for models with color compression distortions.

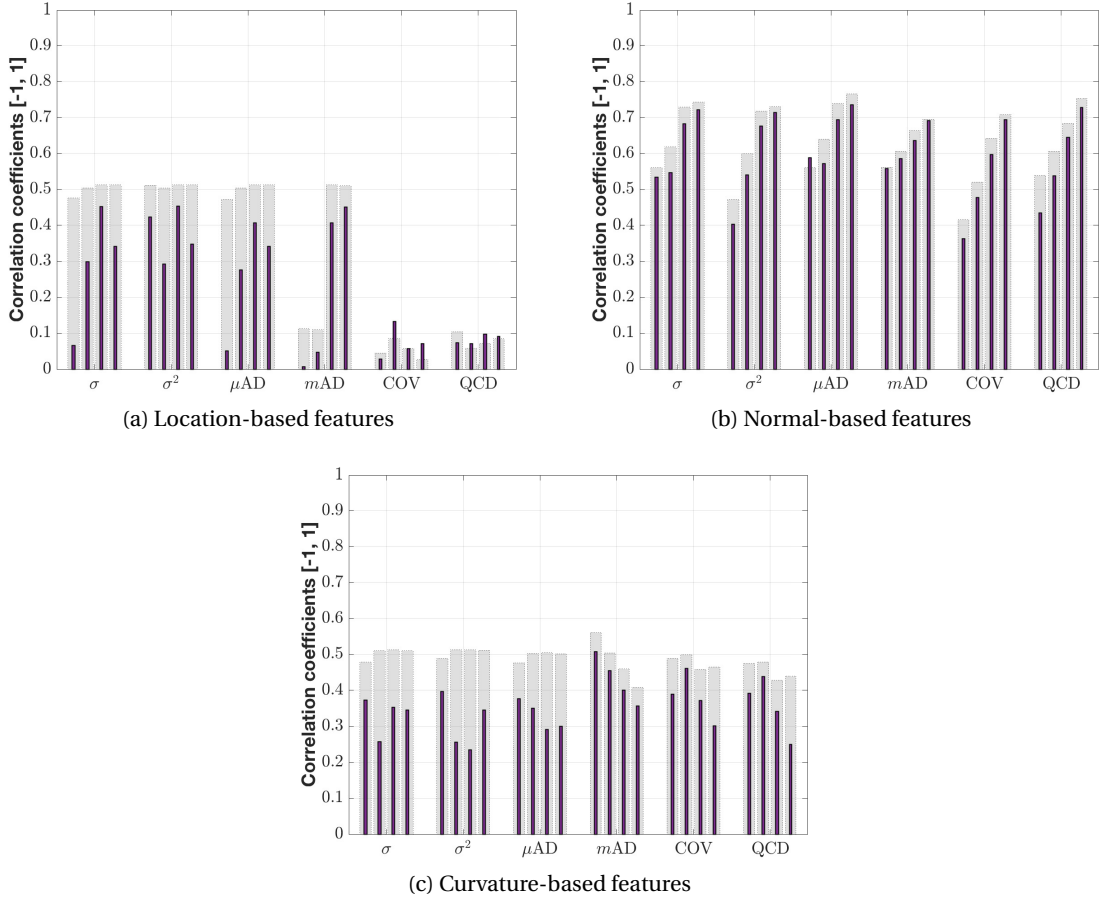


Figure 6.23 – IRPC *rpoint*: Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the neighborhood size is 6, 12, 24 and 48, from left to right.

In Table 6.7 and Figure 6.22, performance indexes and scatter plots of the best-performing configuration of the proposed metric and the most efficient anchor method are provided, respectively. As can be seen, the PointSSIM using luminance-based features that are extracted from voxelized stimuli at 9 bit-depth with σ^2 and $k = 12$ attains the best prediction, when compared to the alternatives. Similarly good results are obtained when applying no-voxelization, or under different estimators and neighborhood sizes, as can be observed in Figure 6.21, indicating that these features provide a robust solution for this data set.

We refer to chapter 8 for a performance evaluation study of the state-of-the-art objective quality metrics over this data set.

Performance evaluation on IRPC

***Rpoint* session:** In Figure 6.23, plots with the performance of geometry-relevant features are presented, using as ground truth subjective scores collected from the *rpoint* session of

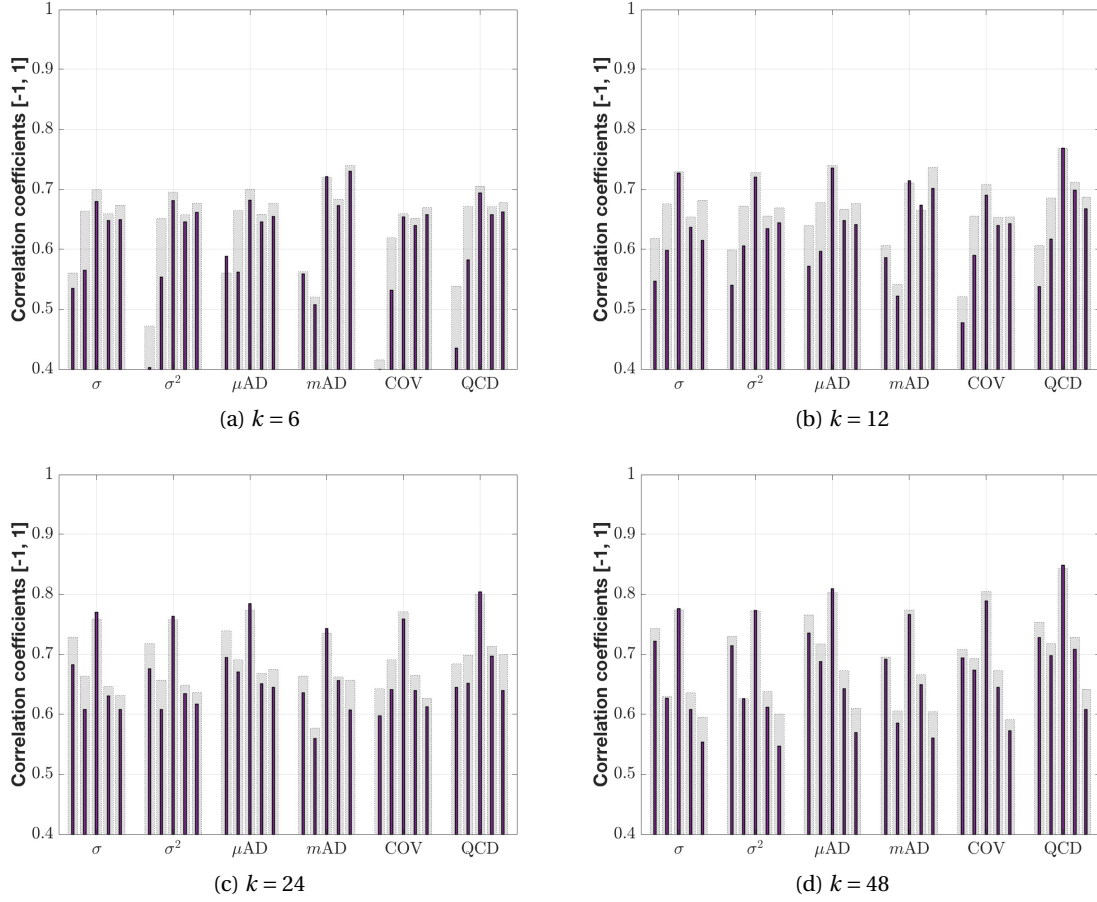


Figure 6.24 – IRPC *rpoint*: Normal-based features. Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the corner left bar corresponds to no voxelization, whereas the rest of the bars correspond to voxel bit-depths equal to 10, 9, 8 and 7, from left to right.

the IRPC data set; this is the session where the models were presented without any color information.

Our results show that location-based and curvature-based features perform very poorly in predicting perceptual impairments in this data set. Using PointSSIM with normal-based features and large neighborhoods leads to better performance, which is still overall limited with respect to previous data sets.

In Figure 6.24, the performance of the normal-based features is illustrated after voxelizing the stimuli at bit-depths $d = \{10, 9, 8, 7\}$, provided that IRPC consists of 10 and 12 bit-depth stimuli. As usual, the performance without applying voxelization is also displayed for comparison purposes.

As can be observed, the performance of the majority of estimators is improving as the neigh-

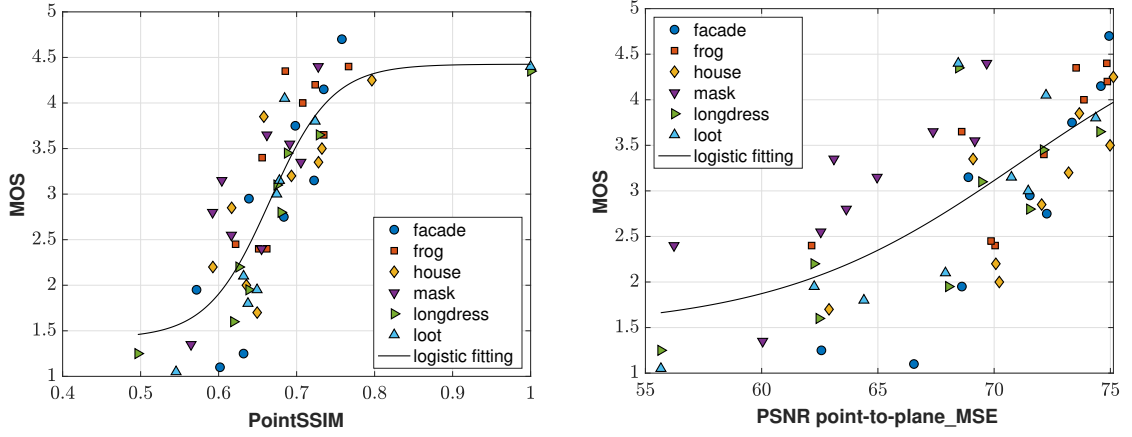


Figure 6.25 – IRPC *rpoint*: Subjective against objective scores from the best-performing configuration (i.e., normal-based features, voxel depth of 9 bits, dispersion estimator QCD, neighborhood size of 48) of the proposed (left) and anchor (right) quality metrics.

Table 6.8 – IRPC *rpoint*: Performance indexes of objective quality metrics. For PointSSIM, the best-performing configuration is reported, using the following notation: [attribute, voxel depth, dispersion estimator, neighborhood size].

	PLCC	SROCC	RMSE	OR
PointSSIM [normal, 9 bits, QCD, 48]	0.843	0.848	0.537	-
point-to-point_MSE	0.460	0.319	0.886	-
point-to-plane_MSE	0.537	0.428	0.842	-
PSNR point-to-point_MSE	0.668	0.647	0.743	-
PSNR point-to-plane_MSE	0.724	0.704	0.689	-

borhood size is increasing for voxel depths higher than 9, whereas for voxel depths lower than 9, the performance is increasing as the neighborhood size is decreasing. A global peak is noted at a voxel depth equal to 9, across all estimators and neighborhood sizes. At this particular voxel depth, the number of points across the stimuli of this data set spans in a narrower range (i.e., similar density), while the models are still represented adequately (i.e., details are preserved). Voxelizing at lower bit-depths leads to very simplistic representations of the models, thus, the performance reasonably drops, whereas at bit-depth larger than 9, the point count across stimuli spans in a wide range.

Note that the point density has an effect on the estimation of attributes (i.e., normals, curvatures) as well as in the computation of features, provided that a k -nn approach is adopted to form neighborhoods in both cases. As shown in annex C and section 6.1.3, the region over which the surface is approximated has a strong influence on the accuracy of the estimated attributes, and the performance of metrics that make us of them, respectively. Yet, this effect can be moderated by an appropriate selection of a target voxel resolution.

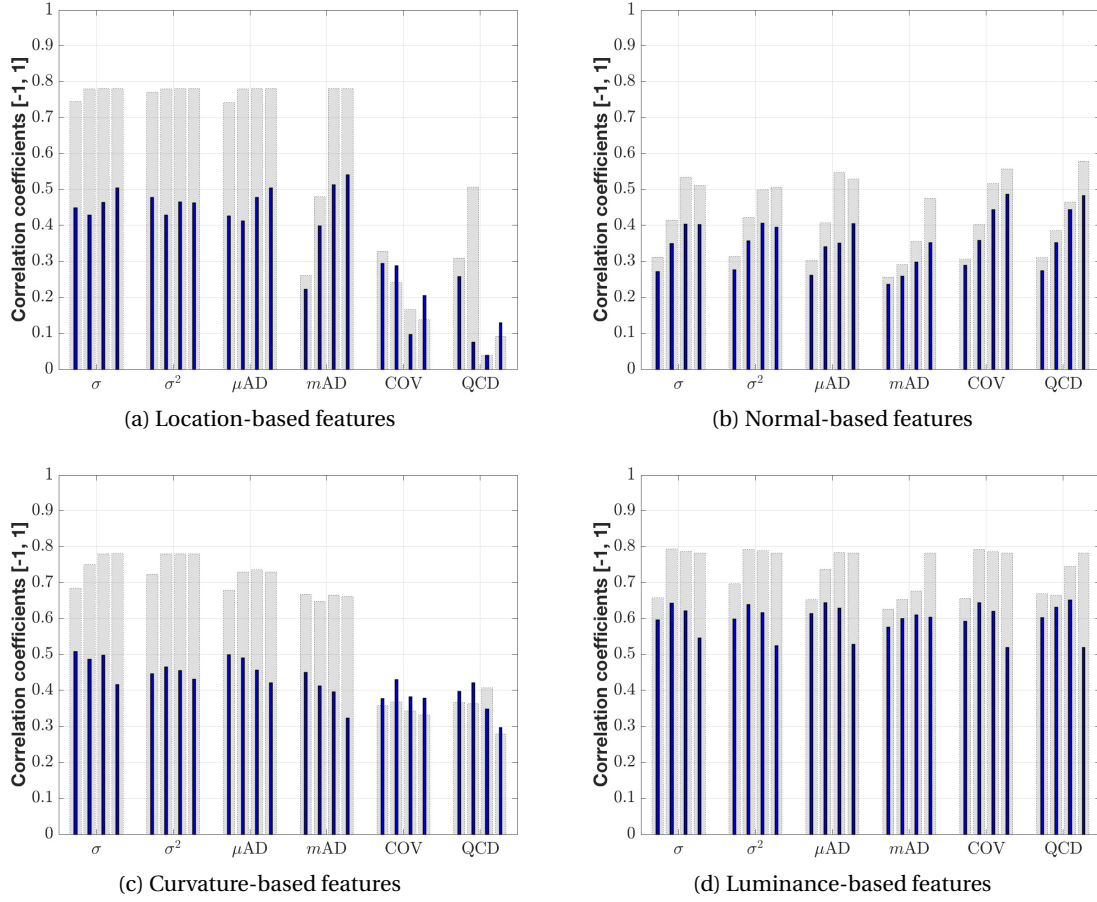


Figure 6.26 – IRPC *rcolor*: Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the neighborhood size is 6, 12, 24 and 48, from left to right.

In Table 6.8, performance indexes for the best-performing configuration of PointSSIM and anchor methods are reported. It is clear that the proposed metric provides a better alternative in this data set, showing substantial performance gains with respect to the competitors. The good performance of normal-based features in this data set is in accordance with the results obtained under the benchmarking of the plane-to-plane metric in section 6.1.3.

Finally, in Figure 6.25, scatter plots with subjective against objective quality scores are provided, using the highest performing anchor method and the aforementioned configuration for PointSSIM. Evidently, the performance of the former metric is rather limited, while the correlation of the PointSSIM scores with the subjective ground truth is fairly accurate.

It is noteworthy that the best-performing metric reported in the literature is given in (Javaheri et al., 2020a), with PLCC = 0.801 and SROCC = 0.777, using the point-to-plane with Hausdorff over the 99% of the ranked distances, and min pooling to obtain a symmetric error.

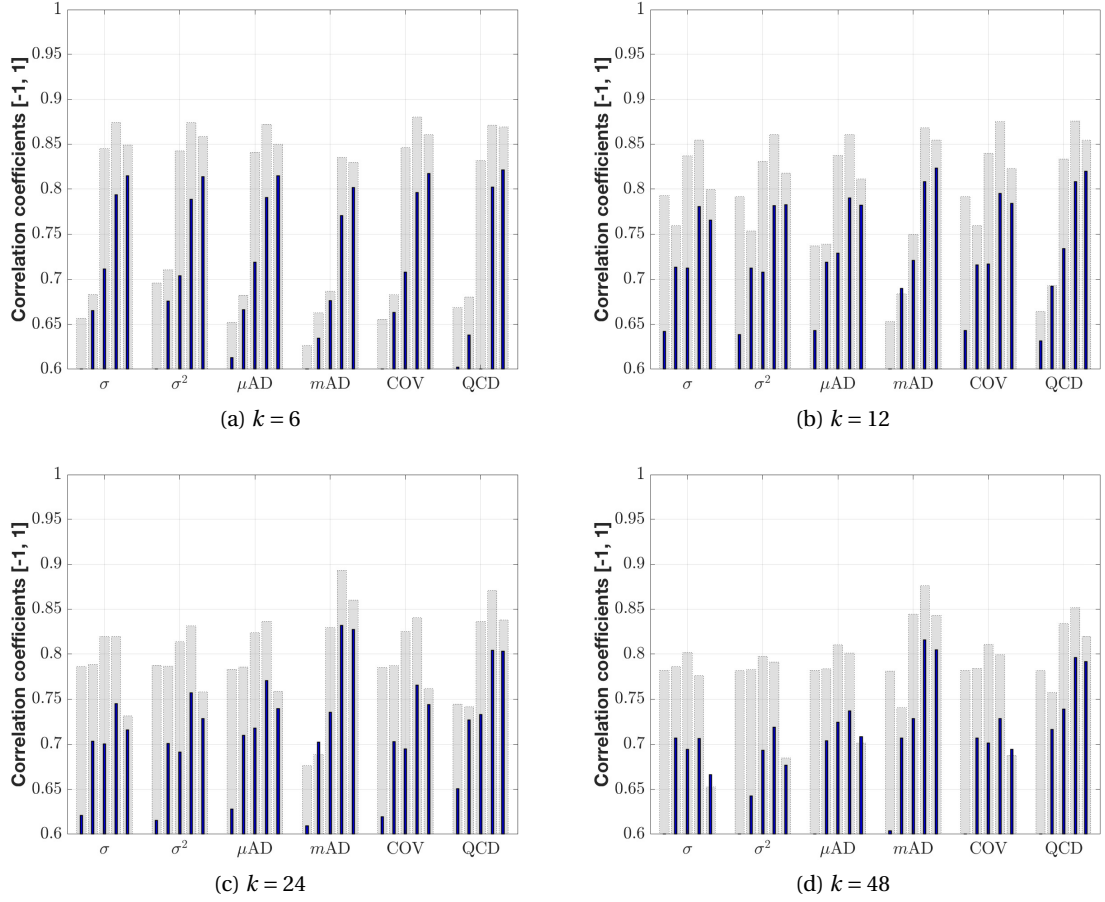


Figure 6.27 – IRPC *rcolor*: Luminance-based features. Performance indexes PLCC (thick bars) and SROCC (thin bars) are grouped per metric. In every group, the corner left bar corresponds to no voxelization, whereas the rest of the bars correspond to voxel bit-depths equal to 10, 9, 8 and 7, from left to right.

Rcolor session: In Figure 6.26, similar plots are provided to present the performance of the metrics in the IRPC data set, under the *rcolor* session.

In this case, the luminance-based features are found to be the most accurate predictors. However, their performance is notably deteriorated with respect to the J-PCED2 and M-PCCD data sets. This performance decrease can be explained by the fact that the color is not directly degraded in this case. Nonetheless, distortions are inherently added from point re-positioning and down-sampling due to geometry encoding.

The second best option is given by the location-based features, in regard to the PLCC index. However, the low SROCC values indicate that the predictions are not very reliable. The majority of features that capture surface roughness perform very poorly, with the exception of some metrics, namely, σ , σ^2 , μAD and mAD , applied on curvature values. The overall limited performance can be justified by the diverse topology of the contents of this data set, which

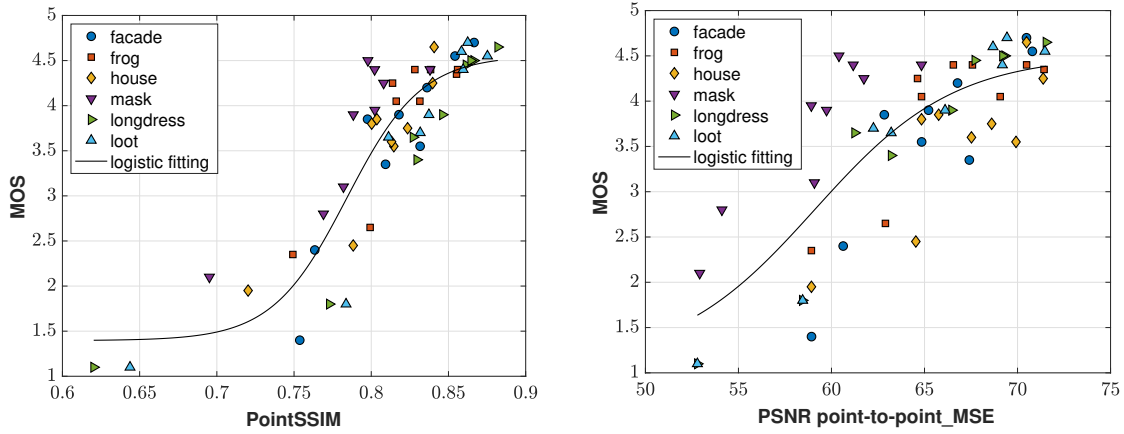


Figure 6.28 – IRPC *rcolor*: Subjective against objective scores from the best-performing configuration (i.e., luminance-based features, voxel depth of 8 bits, dispersion estimator *mAD*, neighborhood size of 24) of the proposed (left) and anchor (right) quality metrics.

Table 6.9 – IRPC *rcolor*: Performance indexes of objective quality metrics. For PointSSIM, the best-performing configuration is reported, using the following notation: [attribute, voxel depth, dispersion estimator, neighborhood size].

	PLCC	SROCC	RMSE	OR
PointSSIM [luminance, 8 bits, <i>mAD</i> , 24]	0.893	0.832	0.448	-
point-to-point_MSE	0.629	0.606	0.775	-
point-to-plane_MSE	0.650	0.631	0.758	-
PSNR point-to-point_MSE	0.792	0.737	0.608	-
PSNR point-to-plane_MSE	0.780	0.685	0.624	-
PSNR_Y	0.695	0.672	0.717	-

may impact negatively the estimated quantities (i.e., normals, curvatures). Moreover, recall that the models were subjectively evaluated in the presence of color in this session, which may act as a distractor for judging the underlying geometric distortions.

Interestingly, though, the curvature-based features predict better subjective scores obtained from the *rcolor* session, when compared to their performance over the *rpoint* session, whereas the opposite holds for the normal-based features.

In Figure 6.27, the performance of luminance-based features is demonstrated for different voxel bit-depths, and under every neighborhood size and dispersion estimator. Based on our results, a remarkable performance increase is observed, when voxelizing at a voxel depth of 8 bits. This outcome can be explained by the fact that, as part of the voxelization process, color values belonging to the same block are blended. Geometry degradations applied to the same block will then affect the blending process. Thus, through voxelization, geometry degradations are essentially reflected on the output color attributes.

In Table 6.9, performance indexes for the best-performing configuration of PointSSIM and the anchor metrics under consideration are reported. As can be seen, PointSSIM achieves the most accurate predictions, when compared to the alternative solutions. Finally, in Figure 6.28, scatter plots with subjective against objective quality scores are provided, using the highest performing anchor method and the aforementioned configuration for PointSSIM, confirming the superior performance of the latter.

It is noteworthy that the best-performing metric reported in the literature for this data set is given in (Meynet et al., 2020) from PCQM, with PLCC = 0.90 and SROCC = 0.83.

6.2.4 Discussion

As introduced at the beginning of the chapter, the core idea behind our work was to adapt the idea of “structural similarity” to irregular, multi-dimensional topologies. Translating image metrics to this type of domain underlines the need to include surface information along with texture, in order to get a more comprehensive model for visual distortion.

In this study, we consider three types of attributes which aim at quantifying visual degradations of 3-D shapes, along with the luminance-based features which explore the perception of color. Depending on the data set under study, different attributes achieve the best performance, which can be explained by the different geometric and color characteristics of the models under examination. However, the human visual system does not easily separate between shape and color distortion. Indeed, textural information can mask or worsen geometric imperfections, and vice versa.

Throughout our analysis, it is evident that the luminance-based features are the most consistent and accurate predictors across the tested data sets. Moreover, we observed that, when geometric impairments are reflected on the color attributes, color-based features were found to be a rather effective way to accurately predict perceptual degradations. The formulation of neighborhoods to compute local statistics is essential, permitting color-based features to capture topological distortions, which explains their higher prediction power.

Another key component of our work is the possibility of defining a desired voxel bit-depth on which the metric can be computed. In general, voxelization enables color smoothing and regular down-sampling of geometry, allowing to simulate visual inspection from farther distances. In our context, it is employed as a way to reduce cross-content density differences, enabling measurements that capture distortions of the model at different scales, potentially, providing a more suitable range for objective quality predictors. In that sense, it introduces the concept of “multi-scale” quality evaluation approaches to point cloud contents.

A fine balance, of course, needs to be established between the voxel resolution and the neighborhood size. A too narrow neighborhood may not capture low-frequency components of the underlying surface; on the other hand, too large neighborhoods will not accurately reflect the local properties of the area under consideration. However, the optimal configuration can

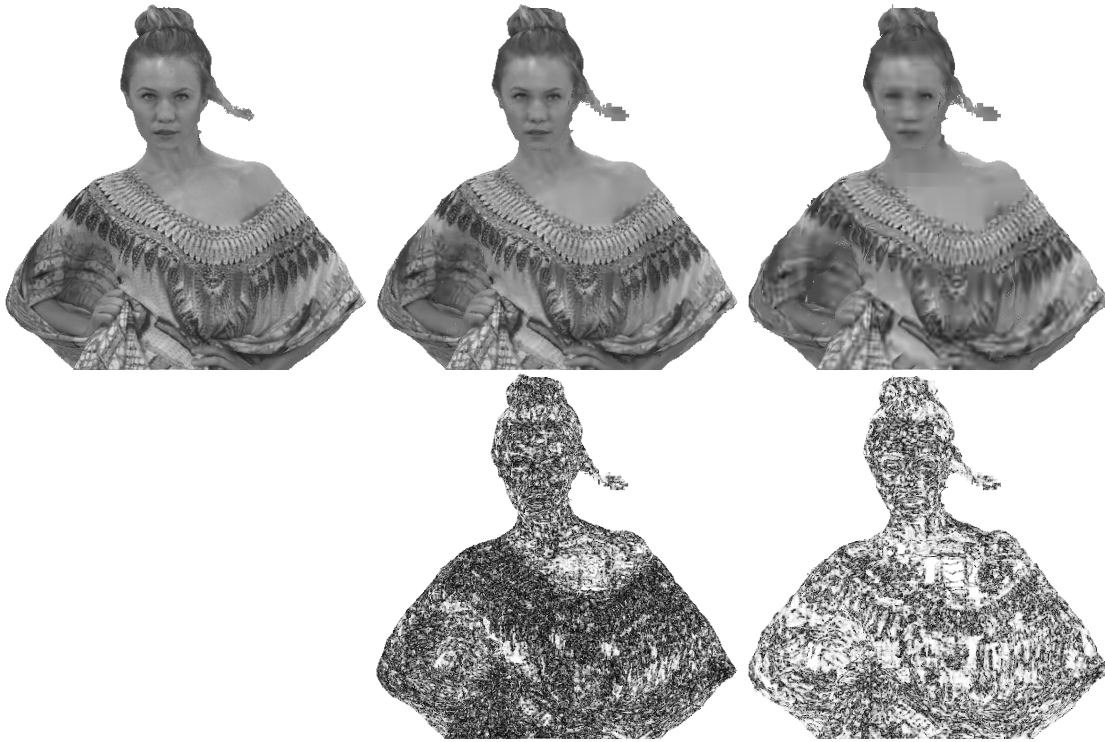


Figure 6.29 – Illustration of a point structural similarity map (black indicates similarity of 1). In the first row, the luminance component of the reference model *longdress*, and two versions after encoding with V-PCC at R03 and R01 following the MPEG Common Test Conditions are displayed, from left to right. In the bottom row, the corresponding structural similarity scores using luminance-based features are provided. The obtained PointSSIM scores are 0.519 and 0.359, for the second and the third stimulus, respectively.

differ per data set, depending on the implicit characteristics of the contents, as well as the acquisition technology that was used to capture them.

The formulation of neighborhoods to compute local statistics, and voxelization prior to features extraction, are two approaches that enable inheritance of distortions from the geometry domain to color error values. The former introduces a spatial dimension to the measurements, whereas the second reduces topological differences between the reference and the model under evaluation, while mapping corresponding distortions to the output color attributes. A representative example was given in our analysis, showcasing the success of color-based features in generalizing to geometry-only distortions, while the effectiveness of the luminance-based features was confirmed over every tested data set.

In Figure 6.29, a visual example of structural similarity maps that were obtained from two encoded stimuli is provided, using the best-performing configuration of the luminance-based features. The luminance component of the point clouds under exam are additionally given, to allow visual comparisons between the erroneous parts of the image and the corresponding structural similarity scores. In this example, the metric's ability to capture blurriness artifacts

of the models under evaluation is demonstrated.

Note that combining the attributes under exam to provide a unified distortion value might lead to more robustness and better performance across data sets. Moreover, consolidating individual measurements under different point cloud scales to obtaining a final score, may further improve the prediction accuracy of the quality estimates. Automatically derived configurations for the selection of features, estimation of attributes, voxel resolutions, and neighborhood sizes, at multiple point cloud scales, is left as a future work. Finally, remark that our algorithmic pipeline can be extended to other point cloud attributes, while it is not limited to the set of quantities, or the estimators that were defined as part of this study.

6.3 Conclusions

In this chapter we describe our proposed objective quality metrics that directly operate on the point cloud domain to predict visual impairments. We initiate by introducing a geometry-based predictor, namely plane-to-plane, which relies on the angular similarity of unoriented normal vectors between associated points that belong to an original and a distorted model. To benchmark the metric, several subjectively annotated data sets were recruited. Provided the sensitivity of this method to the normal attribute information, our performance evaluation study is considering 3 widely used normal estimation algorithms and a variety of configurations. Our results show that, the algorithm that is employed for the neighborhood identification over which the normals are estimated, is crucial. In particular, among the examined cases, the range search-based algorithms were found to behave better, driving the plane-to-plane metric to higher performance. Moreover, they were found to be less sensitive in regard to the neighborhood size selection with respect to the k -nn variant. This can be explained by the fact that they permit comparison of normal vectors that reflect surfaces, which correspond to the same local region of the content. Moreover, from the range search-based variants that were examined, quadric led to slightly better results than plane fitting, albeit the performance of the metric wasn't critically impacted by the order of polynomial surface. Best-performing configurations show high prediction power and competitive results across a number of subjectively annotated data sets, with respect to widely-used quality metrics that served as anchors in our experimentation.

We then proceed with the description of the PointSSIM metric. The principle of operation for this predictor relies on estimates of dispersion for the local distribution of quantities that reflect properties of a point cloud attribute. In this framework, structural features that measure local topological or color consistencies were defined and evaluated. Our results show that, under different data sets, features extracted from different point cloud attributes may be more effective in providing accurate predictions. Intrinsic characteristics, and topological and color distributions of the stimuli that comprise a data set are essential on this matter. As part of the metric, a voxelization step is introduced, which can be optionally enabled. This module unleashes interesting properties that can be exploited during feature extraction. For instance,

different combinations of target voxel resolutions and neighborhood sizes over which features are computed, or attributes are estimated (i.e., normals, curvatures) can be activated. Thus, relevant measurements that reflect perceptual impairments at different scales can be obtained. Another advantage is that it offers the possibility of eliminating cross-content point density variations. This property was demonstrated in benchmarking of data sets including contents at multiple voxel resolutions. The performance of the metric was extensively analysed on several data sets, outperforming well-established alternative methods, under proper configurations.

Prototype implementations of the plane-to-plane and PointSSIM metrics are made publicly available. Information on how to retrieve the scripts and where to refer for additional information are provided in annex E.

7 Image-based objective quality metrics

Point cloud objective quality assessment is typically performed by metrics, which can be distinguished in two main categories: (a) point-based, and (b) image-based, as introduced in the previous chapter. The former denotes a class of algorithms that operates on the 3D point cloud domain, whereas the latter signifies the application of methods that operate on the 2D image domain, after capturing views of the rendered 3D models under evaluation.

Regarding point-based metrics, predictions rely on the quantification of distortions that are present in point cloud attributes. Such attributes can be either explicitly stored in a point cloud format, or can be estimated from the given data. In metrics that depend on more than one attributes, each of them is most commonly treated separately, and a pooling method is issued on the individual measurements. Ideally, the pooling should take under consideration the sensitivity of human perception in order to compute a total quality score. In practice, a weighted average is employed and optimal weights are identified per data set through regression analysis. However, for such schemes there is ambiguity regarding the generalization abilities with unseen degraded models. Other limitations of point-based metrics include potential dependencies on attributes that need to be estimated before execution, such as normal vectors or curvature values. As seen in previous chapters, the performance of relevant metrics can be strongly affected by the selection and configuration of the estimation algorithm. The uncertainty of the predictions is further enhanced when considering that a different rendering mechanism may be employed to display a model. Specifically, a rendering method may mask or enhance distortions, thus, affecting the visual appearance and in turn the perceived quality of a model. Such refinements are often not taken into account and, therefore, they are not typically captured by current point-based approaches.

Image-based counterparts offer a solution to evaluate visual quality in a more holistic way. Specifically, by obtaining views of the displayed model, both geometric and color degradations are reflected as introduced by the corresponding rendering device. Then, high-performing 2D imaging algorithms can be employed to assess perceptual quality. Yet, there are several drawbacks that are coming with this approach. Specifically, image-based metrics denote view-dependent and rendering-dependent solutions (Lavoué et al., 2016; Alexiou et al., 2019a).

The former indicates that the predictions may vary for a different set of views, which implies that the camera distance, position and orientation, as well as the number of acquired images, will affect the quality scores. The latter denotes that the rendering pipeline together with the environmental and lighting conditions of the virtual scene can govern the visual outcome. For this purpose, it is often advised to reproduce the same rendering settings to the ones employed during subjective consumption, in order to simulate the visual experience of the end-user. Although this might be feasible in some applications, for some others it can be considered too tedious and, often, too complicated for quality evaluation purposes. For instance, when tuning transmission or compression schemes in real-time communications, relaxations in complexity are required.

In this chapter, we investigate the performance of image-based objective quality metrics on point cloud contents. In the first part, a relevant quality assessment framework is defined. The scheme is validated with two data sets that contain the same testing material, which was subjectively evaluated under two different rendering schemes. This way, generalization capabilities of the approach are examined. Influencing factors such as the relevance of the background information in the computation of the metrics, and the number of views that are employed for estimating the perceived quality of a model are also investigated. In the second part, we exploit user's recorded behavior during subjective assessment of the stimuli in objective quality evaluation. We initiate by examining whether an average objective score across the frames that were inspected by subjects leads to accurate predictions. Anticipating limitations of the approach, we proceed by proposing a weighting scheme that is applied on model views acquired from fixed camera arrangements. The weights are computed using human interactivity data, under the hypothesis that the importance of a view is related to the duration of inspection from participants.

This chapter is based on material that has been published in (Torlig et al., 2018a; Alexiou and Ebrahimi, 2019).

7.1 Exploiting model views

In this section, we describe a framework for image-based point cloud objective quality evaluation. The system relies on snapshots of the models, which are displayed using the same rendering settings that were applied during subjective assessment. Thus, the visual appearance of the point clouds as were displayed to the users is reproduced. Model views are captured from camera layouts that allow evenly distributed viewpoints from a fixed distance. Then, 2D imaging metrics are applied on the acquired images. Corresponding quality scores are computed and pooled together to provide a prediction of visual quality for the model under assessment. Using this framework, we evaluate the effectiveness of 2D imaging algorithms in capturing point cloud distortions, as shown through different renderers. For this task, we use a set of point clouds that was subjectively assessed under two different rendering schemes. Objective scores are computed using a fixed set of viewpoints that grant coverage

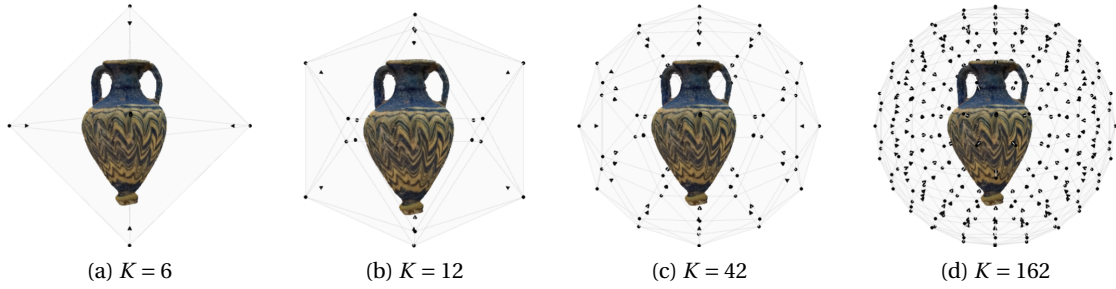


Figure 7.1 – Camera layouts to capture views of the models.

of the model's outer view. In this context, the performance of the metrics is evaluated, and insights are drawn regarding their generalization capabilities. In a second step, we evaluate the impact of background pixels removal from the computations in order to improve prediction accuracy. Finally, we examine the performance of the system by integrating additional views in the derivation of a global quality score for the model under evaluation.

7.1.1 Model views generation

Given a distance, a 3D model can be inspected from an infinite number of points of the surrounding view sphere. Enabling a vast amount of viewpoints, though, is both impractical and unnecessary, as in a dense configuration two successive points provide very similar information. In our analysis, a model can be captured by K regularly-spaced viewpoints with the following camera layouts: (a) a single point that captures the frontal view (i.e., $K = 1$), in order to examine whether a single image corresponding to the initial view of the model that was displayed to the subjects provides a good approximation of its visual quality; (b) the vertices of a surrounding octahedron (i.e., $K = 6$), which provides the minimum set of most diverse views that capture the entire outer view of a model; and (c) points lying on a surrounding geodesic sphere with coordinates determined by iterative subdivisions of a regular icosahedron up to 2 levels (i.e., $K = 12, 42, 162$). The latter is a commonly used arrangement in studies for view selection (Lavoué et al., 2016; Bonaventura et al., 2018), that provides a consistent approach to approximate uniformly distributed samples that are lying on the surface of a sphere. By iteratively subdividing the regular icosahedron, gradual granularity with progressive integration of new viewpoints on the previous set is achieved. This is important in order to identify whether additional views can improve the prediction accuracy. In Figure 7.1, indicative examples of the camera arrangements are illustrated.

Besides the number of viewpoints, additional influencing factors, such as the rendering configurations, the distance between the content and the camera, the direction of the camera, the lighting conditions, and the type of projection (e.g., orthographic, perspective) may vary. In our set-up, we enable, wherever possible, the exact same settings that were used in the subjective experiments in order to better simulate the user's experience and decrease the



Figure 7.2 – Model views captured from a camera layout with $K = 6$.

parameter space, simplifying our analysis. In particular, the rendering configurations (i.e., splat size or voxel grid resolution), bitmap resolutions, lighting conditions, shading models and types of projection are replicated to render the models. The distance between the camera and the model is fixed to match the one that was determined for the initial view presented to the subjects, ensuring that a model can be comfortably seen in its entirety from every point of the view sphere. The direction of the camera points at the center of the sphere, which is also the origin of the models, while the pose of each content can be determined at will. Under these settings, from each camera position, a 2D image is captured for every model, as shown in Figure 7.2.

7.1.2 Validation methodology

Data sets

A set of 6 static point clouds degraded using the CWI-PCL (Mekuria et al., 2017a) encoding engine are evaluated under two different rendering schemes, namely, voxel-based and splat-based, resulting in two subjectively annotated data sets, hereafter named after the type of rendering solution that was applied. Detailed information regarding the experiments can be found in section 4.2. In brief, the selected point clouds represent both human figures and inanimate objects, while different combinations of geometry and color quality levels are employed for encoding. The subjective evaluations were held in two separate experiments using identical testing environment and equipment. The simultaneous DSIS test method with 5-grading scale was employed in both, while interactivity without limitations was granted through corresponding testbed platforms that were deployed. The rendering schemes are summarized below.

The voxel-based renderer relies on real-time voxelization that is performed on the transformed coordinates of the model under inspection, according to user's interactions. The texture information lying on the voxels is orthographically projected onto an image that is displayed into the screen, with every voxel occupying a neighborhood of pixels that is determined by the viewing distance. A mid-grey color, (127, 127, 127) in RGB colorspace, was employed as

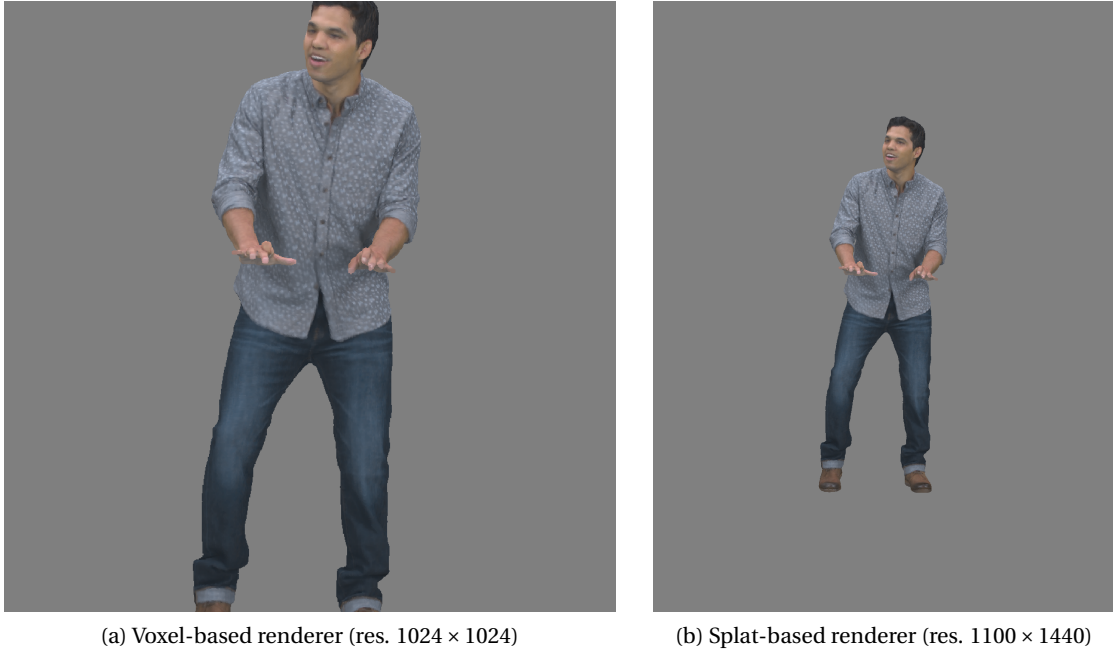


Figure 7.3 – Initial frontal view of reference model *loot* as displayed to participants during subjective evaluations in corresponding testbeds.

background. No lighting, or shading models were applied. More details regarding this renderer can be found in annex D.1.

The splat-based renderer makes use of cubic primitives of adaptive size based on local densities, in order to provide watertight views of the models under inspection. In this experiment, a perspective projection was adopted, while the background color was set to the same RGB color (127, 127, 127). The default lighting conditions were enabled; that is, a headlight located at the current camera position, without any shading model. More details on this rendering solution are provided in annex D.2.

Note that the aforementioned rendering schemes lead to very different types of artifacts. That is, perception of missing pixels in the case of voxel-based, and coarse surface approximations in the case of splat-based counterpart are observed, as illustrated in Figures 4.14 and 4.15. Further details regarding the voxel-based and the splat-based experiments can be found in section 4.2.

Computation of quality metrics

In this study, we evaluate the performance of 4 well-established 2D methods, namely, PSNR, SSIM (Wang et al., 2004), MS-SSIM (Wang et al., 2003) and VIFp (Sheikh and Bovik, 2006) (i.e., multi-scale in pixel domain), applied on the luminance channel, after conversion of the original RGB color attributes to the YCbCr colorspace using the ITU-R Recommendation

BT.709-6 (ITU-R BT.709-6, 2015).

Following the procedure detailed in section 7.1.1, views of the rendered model under evaluation and its reference version are acquired from every camera position, for a given camera layout. For each metric, an objective score is computed per viewpoint, and an overall quality prediction is obtained by average pooling. Notice that during image acquisition, even small misalignments between model views will have a dramatic effect on the computed scores. Remark also that in this data set, the orientation of the models was aligned; for instance, for $K = 1$, the frontal view was obtained for every model, thus, no modifications were applied in their original pose.

The distance between the position of the cameras and the origin of the models was set equal to the initial view that was presented to the subjects. The captured images match the resolution of the bitmaps that were displayed during subjective evaluations. That is, 1024×1024 for the voxel-based, and 1100×1440 for the splat-based experiment. The rest of the rendering parameters were replicated from the corresponding subjective evaluation testbed, as mentioned earlier.

An illustrative example of the initial frontal view of *loot*, as presented to the users from both renderers, can be found in Figure 7.3. The images are exported in either RGB or RGBA colorspace, depending on the application. For instance, transparency information may assist in the identification of the foreground, thus, determining the effective part of the image that contains the displayed model. Under all circumstances, the objective metrics were computed by MATLAB implementations and, when applicable, official script releases were employed¹.

To obtain a reference regarding the performance of the image-based metrics with respect to well-established point-based approaches, the point-to-point with MSE, point-to-plane with MSE, the corresponding geometric PSNR variants, and the color PSNR computed on the luminance channel are additionally evaluated, using the software ver. 0.13.5 (Tian et al., 2017c). Details regarding their execution are provided in section 6.1.2, under the computation of metrics sub-section. Note that in this data set, we estimate the normals of all contents using k -nn plane fitting with $k = 12$, as implemented in PCL.

Benchmarking of quality metrics

Following the methodology detailed in section A.3, the PLCC, the SROCC, the RMSE, and the OR performance indexes are computed between pairs of MOS and predicted MOS, to measure the performance of the plane-to-plane metric against the subjective ground truth. The predicted MOS is obtained after applying the logistic fitting function on the objective quality scores.

¹http://live.ece.utexas.edu/research/Quality/index_algorithms.htm

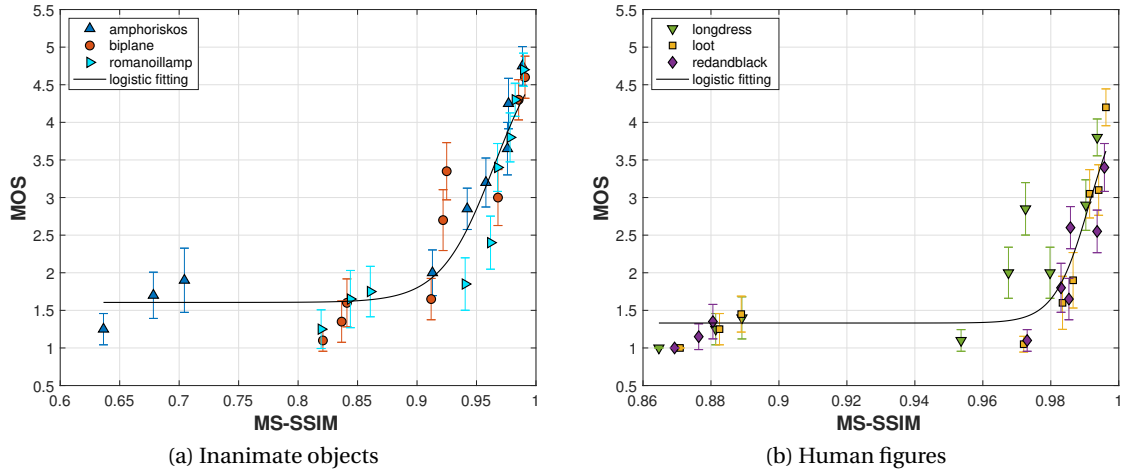


Figure 7.4 – Subjective against objective scores from the best-performing image-based quality metrics per type of content, under voxel-based rendering.

Table 7.1 – Performance indexes of image-based quality metrics per type of content, under voxel-based rendering.

	Inanimate objects				Human figures			
	PLCC	SROCC	RMSE	OR	PLCC	SROCC	RMSE	OR
PSNR	0.807	0.791	0.719	0.630	0.728	0.735	0.650	0.667
SSIM	0.883	0.836	0.572	0.593	0.847	0.820	0.503	0.667
MS-SSIM	0.934	0.931	0.436	0.407	0.883	0.875	0.444	0.556
VIFp	0.908	0.936	0.511	0.556	0.898	0.859	0.417	0.556
point-to-point_MSE	0.825	0.822	0.688	0.630	0.700	0.702	0.678	0.593
point-to-plane_MSE	0.816	0.755	0.705	0.667	0.699	0.693	0.679	0.593
PSNR point-to-point_MSE	0.825	0.822	0.688	0.630	0.701	0.709	0.675	0.593
PSNR point-to-plane_MSE	0.822	0.755	0.693	0.667	0.701	0.693	0.675	0.593
PSNR_Y	0.694	0.712	0.876	0.815	0.823	0.701	0.539	0.704

7.1.3 Results

For analysis purposes and to highlight differences introduced by every parameter under examination in our quality evaluation framework, we opt to split the point clouds according to their type of content; that is, human figures and inanimate objects. This decision is supported by the analysis carried out in section 4.2.3, reporting different rating behaviors for the two types of content, in both data sets, which was confirmed by corresponding statistical comparison.

Rendering schemes

As a first step, the generalization capabilities of the image-based metrics are investigated. For this analysis, we set $K = 6$, which essentially denotes that every model is projected on the faces

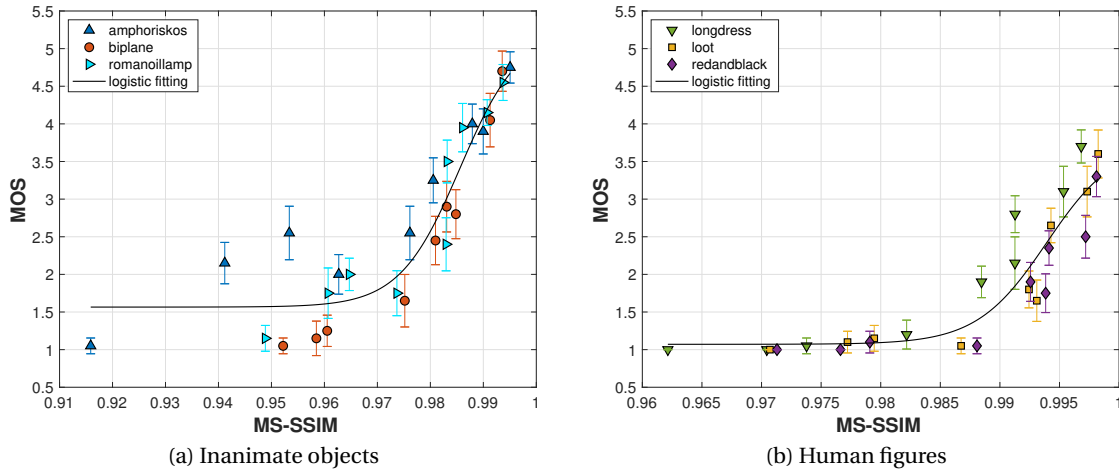


Figure 7.5 – Subjective against objective scores from the best-performing image-based quality metrics per type of content, under splat-based rendering.

Table 7.2 – Performance indexes of image-based quality metrics per type of content, under splat-based rendering.

	Inanimate objects				Human figures			
	PLCC	SROCC	RMSE	OR	PLCC	SROCC	RMSE	OR
PSNR	0.677	0.672	0.891	0.852	0.713	0.763	0.640	0.889
SSIM	0.768	0.717	0.774	0.741	0.848	0.819	0.484	0.667
MS-SSIM	0.937	0.910	0.422	0.593	0.918	0.925	0.361	0.593
VIFp	0.900	0.893	0.529	0.630	0.885	0.889	0.427	0.630
point-to-point_MSE	0.763	0.769	0.781	0.815	0.783	0.789	0.567	0.704
point-to-plane_MSE	0.759	0.693	0.787	0.852	0.782	0.763	0.570	0.704
PSNR point-to-point_MSE	0.771	0.769	0.770	0.815	0.784	0.799	0.566	0.704
PSNR point-to-plane_MSE	0.759	0.684	0.787	0.852	0.784	0.763	0.566	0.704
PSNR_Y	0.782	0.756	0.753	0.741	0.765	0.658	0.587	0.852

of a surrounding cube.

In Table 7.1, the performance indexes for the image-based metrics and the point-based anchor methods against the ground truth subjective scores are reported, under the voxel-based rendering. In Figure 7.4, scatter plots with the overall best-performing quality predictor (see below) are illustrated, per type of content. Based on the performance indexes, it is evident that the image-based metrics perform better than the point-based counterparts, showing higher accuracy in predicting perceptual quality in both sets of contents. It should be noted that point-based approaches are limited by the fact that they either examine geometry-only or color-only distortions for a model under evaluation. For instance, encoding the geometry of a point cloud at a specific degradation level and increasing the color quality, doesn't lead to any improvements in the predictions of geometry-only metrics. On the contrary, image-based



Figure 7.6 – Model view of *longdress* as consumed by subjects (left) and after removing the background information (right).

methods are able to capture such distortions. Among the examined metrics, the MS-SSIM was found to achieve the best performance. One justification for this outcome could be the multiple scaling that takes part in the computations, which simulates perception of models from different distances.

Similarly, in Table 7.2 and Figure 7.5, performance indexes and scatter plots are provided using the scores obtained from the splat-based experiment. Our results are equivalent. In fact, the prediction power of the image-based metrics is even better in this test, outperforming the point-based alternatives with larger margins.

A general remark is that, in both data sets, better performance is attained for point clouds that represent inanimate objects. Note also that the two data sets are comprised essentially from the same point clouds, thus, the point-based quality scores are the same. However, these point clouds were rendered differently, which implies that the visual outcome was different and, hence, the subjective ratings differ (see section 4.2). These variations cannot be captured by point-based metrics, which is evident by the differences of PLCC and SROCC between the two tests. At the same time, substantially narrower variations and higher scores are observed for the same performance indexes using MS-SSIM and VIFp, which suggests better adaptation.

Background removal

Considering Figures 7.4 and 7.5, it can be observed that the objective scores in the second case are higher. This can be explained by the larger region that background covers in the second model view, which is evident in Figure 7.3. When the entire image is taken into

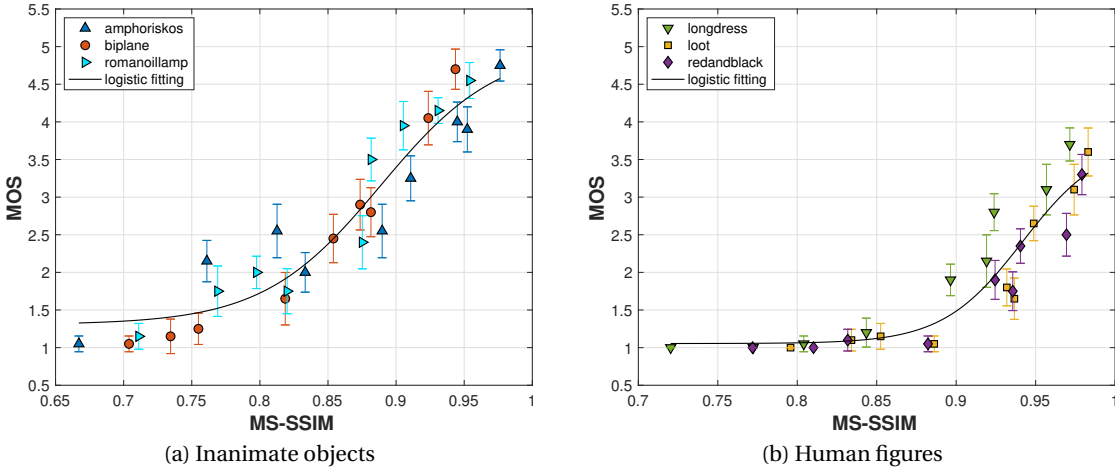


Figure 7.7 – Subjective against objective scores from the best-performing image-based quality metrics per type of content, computed after background removal, under splat-based rendering.

Table 7.3 – Performance indexes of image-based quality metrics per type of content, computed on the entire model views and after background removal, under splat-based rendering.

	Inanimate objects				Human figures			
	PLCC	SROCC	RMSE	OR	PLCC	SROCC	RMSE	OR
PSNR	0.677	0.672	0.891	0.852	0.713	0.763	0.640	0.889
PSNR (F)	0.639	0.604	0.929	0.815	0.692	0.723	0.659	0.852
SSIM	0.768	0.717	0.774	0.741	0.848	0.819	0.484	0.667
SSIM (F)	0.907	0.898	0.510	0.741	0.838	0.788	0.498	0.741
MS-SSIM	0.937	0.910	0.422	0.593	0.918	0.925	0.361	0.593
MS-SSIM (F)	0.955	0.944	0.359	0.556	0.933	0.927	0.328	0.519
VIFp	0.900	0.893	0.529	0.630	0.885	0.889	0.427	0.630
VIFp (F)	0.926	0.924	0.456	0.704	0.921	0.921	0.357	0.556

account in the computation of the metrics, background pixels contribute in the objective score that is obtained. However, the quality of a model is subjectively assessed based on the foreground; moreover, objective scores shouldn't depend on the amount of background pixels that surround a displayed model.

In this second step, the impact of discarding background pixels from the computation of image-based metrics is evaluated. A visual illustration of a model view by including and excluding background information is provided in Figure 7.6. For this task, the splat-based data set is selected, using $K = 6$ views. For the application of the metrics on the foreground of the images, the original scripts are modified accordingly. Note that there are several approaches for the determination of the region over which they can be computed. In this study we preliminarily tested the foreground of the reference, the union and the intersection between the reference

and the distorted foregrounds. Based on our results, the union of foregrounds was found to perform better than the alternatives, while constituting an intuitively more coherent approach. Thus, results adopting this solution are reported. Regarding the application of filters on pixels that belong to (foreground) edges of the model, as part of the execution of particular metrics (e.g., SSIM), the background color was accounted to better simulate the way the views were consumed by subjects.

In Table 7.3 the performance indexes are reported by computing objective scores on the entire image and the identified foreground region (F). In Figure 7.7, scatter plots of the best-performing metrics are illustrated. Based on our results, improvements are remarked for the SSIM when evaluating objects, and for MS-SSIM and VIFp in both types of content. This comes as a result of the enhanced generalization capabilities of the metrics across models, which can be seen when comparing Figures 7.5 and 7.7. Finally, based on the same figures, it is noted that the objective scores span a larger range when considering only the foreground pixels, with respect to the whole image.

Camera layouts

In our previous efforts, we set $K = 6$ model views to compute the image-based metrics, for analysis purposes. In this third step, we investigate whether further improvements in terms of correlation are achieved by considering additional viewing angles of the model under evaluation. For simplicity reasons, in the forthcoming analysis only the MS-SSIM applied on the foreground of the images is considered, as the best-performing metric. It is noteworthy that similar results are obtained using VIFp.

In Table 7.4, the performance indexes using MS-SSIM are depicted, under every supported camera layout from the proposed framework. Based on our results, the performance of the metrics remains stable in the case of inanimate objects, or even decreases in the case of human figures, when introducing additional views. In fact, for the latter case, the best performance is achieved when using only the frontal view of the model, while similarly accurate predictions are achieved in the former case. Scatter plots indicating the performance of the metrics using $K = 1$ are depicted in Figure 7.8.

By repeating the same procedure using only the frontal view from the voxel-based rendering experiment, we confirm that the performance indexes are high. In particular, using the MS-SSIM, we observe $\{PLCC, SROCC, RMSE, OR\}$ equal to $\{0.921, 0.882, 0.475, 0.482\}$ for inanimate objects and $\{0.871, 0.840, 0.466, 0.556\}$ for human figures, respectively. These indexes denote comparable performance with the one observed using $K = 6$ views, reported in the first part of this section. Note that, in this case, we include the background information in the computations, in order to permit comparisons.

Our results suggest that enabling additional views doesn't necessarily lead to better visual quality predictions of the rendered model. At the same time, even one view could be sufficient

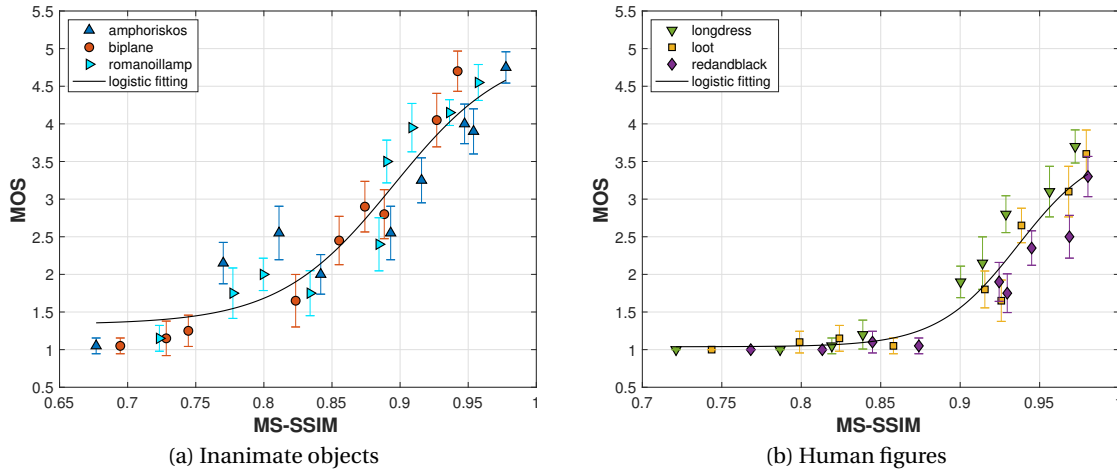


Figure 7.8 – Subjective against objective scores from MS-SSIM per type of content, computed on $K = 1$ model view after background removal, under splat-based rendering.

Table 7.4 – Performance indexes of MS-SSIM per type of content, computed on model views from different camera layouts after background removal, under splat-based rendering.

	Inanimate objects				Human figures			
	PLCC	SROCC	RMSE	OR	PLCC	SROCC	RMSE	OR
$K = 1$	0.951	0.944	0.373	0.519	0.952	0.935	0.279	0.519
$K = 6$	0.955	0.944	0.359	0.556	0.933	0.927	0.328	0.519
$K = 12$	0.949	0.944	0.381	0.519	0.926	0.920	0.344	0.519
$K = 42$	0.949	0.945	0.383	0.519	0.926	0.915	0.345	0.556
$K = 162$	0.949	0.945	0.384	0.519	0.925	0.915	0.347	0.519

to achieve high performance.

7.2 Exploiting user views

In the analysis that was conducted in the previous section, a fixed set of viewpoints was defined in order to characterize the visual quality of a model. Moreover, each view was treated with equal importance. However, this might not be representative of the experience of the users when interacting with the contents. In this section we examine whether user navigation data recorded during subjective evaluation can be exploited to improve the prediction performance of image-based metrics.

The most straightforward approach to include user behavior is to objectively quantify the entire visual experience of a user and compare the aggregated prediction to the corresponding subjective rating. This can be performed by pooling across objective scores that are extracted from frames that were inspected by the user during subjective assessment. However, a main

down-side of this method is the computational overhead. Moreover, the impact of the camera distance on the obtained scores should be taken under consideration. In particular, by increasing or decreasing the camera distance under a fixed orientation, the same model view is inspected from closer or distant locations. The difference of the objective scores, though, might not reflect the dissimilarity that is perceived, raising the need for a properly configured distance-aware weighting function.

Another approach is to employ the user interaction behavior as a weighting base for our fixed model set-up. Specifically, we fix the camera distance, and we consider a finite set of camera positions on the viewing sphere, analogously to what was done in the previous section. This allows us to reduce the computational overhead and essentially normalize the objective scores at a given distance. We opt to map the camera positions of the user onto the closest viewpoints in a given camera layout, with larger weights assigned to more frequently visualized viewpoints. This approach ensures that a pre-determined number of model views will be employed for quality prediction, and that the score of a model view that was visited more often will have a higher impact on the final quality score.

This algorithm is based on the logic that different perspectives of a 3D model might be of different importance, as they could be more or less representative or informative regarding the presented content. Our hypothesis is that model views inspected for larger time duration during subjective evaluations are more important for the characterization of the overall perceived quality. For objective quality evaluation, this has been considered in (Lavoué et al., 2016) using importance weights obtained based on a surface visibility algorithm, typically used for viewpoint preference selection (Bonaventura et al., 2018).

7.2.1 User interactivity

Navigation tracks

In this scenario, we consider the entire set of frames that was inspected by each user in the splat-based experiment. Let us define a navigation track as a set of recorded interactions that corresponds to the inspection of a model by a subject. To obtain an objective score for a particular stimulus, each navigation track of every user is employed and the corresponding experiences are reproduced. Note that the recorded information is extracted by periodical system calls at screen refresh rate, which record the camera parameters and a corresponding timestamp. To compute a score that characterizes a navigation track, the corresponding frames are exported in an off-line playback module using this timestamp information. For instance, if a particular view was inspected for 1 second, and we assume 60 fps, this frame will be considered 60 times. For each frame, an objective score is computed on the foreground of the displayed model, and the estimated average across all extracted frames provides a corresponding prediction. The same procedure is repeated for each navigation track and, for each stimulus, the global average across users is employed.

It is remarked that this scenario serves as a first, naive attempt to provide an anchor performance for predictions that consider behavioral information during subjective evaluations. Note that the computations are taking place on the raw captured data, without considering any algorithm to identify potentially outlier behaviors from subjects. Thus, the scope is limited. Yet, it is examined to provide a basis for comparison.

Importance weights

In this alternative scenario, an importance weight based on the same navigation tracks is assigned to every view of a model from a predefined camera layout. To this aim, it was decided to pre-filter the interactivity information to reduce the noise. In particular, as a first step, a time threshold is applied on each navigation track (i.e., interactions of a subject while inspecting a stimulus), in order to remove transitional views that were not carefully examined. In our case, the time threshold is set as one second. This step resulted in keeping viewpoints that correspond to the $\sim 66\%$ of the total interaction time. Secondly, interactivity data that corresponds to translations of the objects and, thus, different camera directions is excluded, as the translations are not considered in the camera layouts for the generation of views in our framework. In our experiment, a total of $\sim 18\%$ of the recorded data from the previous step is further discarded. On the remaining data, each viewpoint of every navigation track is mapped to the nearest camera position in the selected camera arrangement. The total duration of inspection of a stimulus from one view can be obtained by aggregating the individual times of its inspection from that particular view across every subject. The total duration of inspection of a content from a particular view can be analogously derived by combining the individual times of inspection of stimuli that correspond to the content's variations (i.e., compressed versions). The weights of a stimulus or a content are computed as the ratio of the duration of inspection of the corresponding views, divided by the total time of interaction. In our case, weights per content are computed, and these weights are applied on the views that are obtained for every content's variation.

7.2.2 Validation methodology

Data sets

The subjectively annotated data set using the splat-based rendering scheme described in section 4.2, is employed in this study. Note that, in this experiment, behavioral information was recorded in real-time and the design was adjusted accordingly to be able to test our assumptions. Specifically, considering that the occurrence of fatigue could add bias on the time duration the subjects would spend on every evaluation, the test was split in two sessions of less than 10 minutes each.

Computation of quality metrics

The same procedures for model views generation and objective quality scores computation detailed in section 7.1 are followed in this experiment. To compute the objective scores that correspond to every navigation track, a play-back module was additionally developed, reproducing the experience of each user in the subjective evaluation platform, and extracting every frame that was inspected based on the recorded information (i.e., camera parameters and timestamp). The quality metrics are applied on the luminance channel, considering only the foreground of the images, similarly to what was described in the previous section.

Benchmarking of quality metrics

Identically to previous efforts, following section A.3, the PLCC, SROCC, RMSE and OR indexes are employed to characterize the performance of the objective quality metrics.

7.2.3 Results

Navigation tracks

In Table 7.5, performance indexes are reported using objective scores that were computed considering all frames that were inspected by users, as detailed earlier. For comparison purposes, we additionally reprint the performance indexes that were obtained using the objective quality framework described in section 7.1, with $K = 6$ views and after computing the objective scores on the foreground (i.e., sub-section background removal from 7.1.3).

In Figure 7.9, scatter plots indicating the accuracy of the best-performing objective predictions considering the navigation tracks against the ground-truth subjective scores are illustrated, per type of content.

Our results show notable performance drops when compared to the prediction accuracy that is achieved by employing the fixed camera set-up with $K = 6$ (or, in fact, any fixed camera layout presented in section 7.1). This outcome can be explained by the substantial fluctuation of objective quality scores within a single navigation track, mainly due to the viewing distance. As an indicative example, in Figure 7.10, the distribution of objective scores from the navigation tracks of all users during subjective evaluation of encoded models from two contents, namely *amphoriskos* and *longdress*, are illustrated. In particular, the y-axis depicts indexes that correspond to subjects, whereas the x-axis denotes MS-SSIM scores. Each marker corresponds to a frame that was inspected by a particular subject. The color of each marker indicates the subjective score that was given by this subject; thus, reasonably, all markers that correspond to a navigation track of a user are annotated with the same score. By observing this plot, we can confirm that the objective scores are indeed spanning over a large interval for each user, thus, increasing the uncertainty of the obtained measurements.

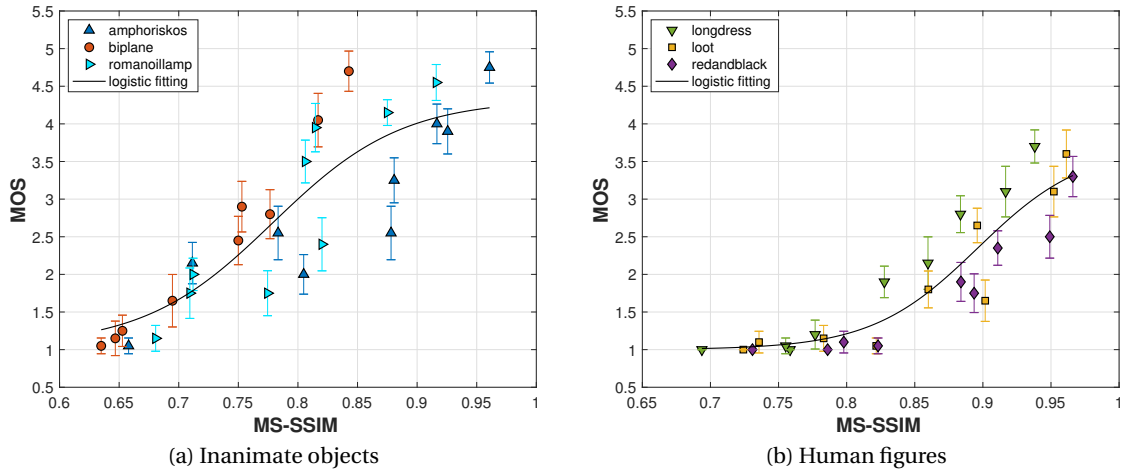


Figure 7.9 – Subjective against objective scores from the best-performing image-based quality metrics per type of content, computed on the navigation tracks of the users.

Table 7.5 – Performance indexes of image-based quality metrics per type of content, computed on the navigation tracks of the users and by using the camera layout with $K = 6$ model views.

	Inanimate objects				Human figures			
	PLCC	SROCC	RMSE	OR	PLCC	SROCC	RMSE	OR
PSNR	0.652	0.601	0.916	0.778	0.740	0.795	0.615	0.815
PSNR ($K = 6$)	0.639	0.604	0.929	0.815	0.692	0.723	0.659	0.852
SSIM	0.807	0.789	0.715	0.778	0.797	0.777	0.550	0.778
SSIM ($K = 6$)	0.907	0.898	0.510	0.741	0.838	0.788	0.498	0.741
MS-SSIM	0.872	0.878	0.593	0.630	0.918	0.897	0.362	0.667
MS-SSIM ($K = 6$)	0.955	0.944	0.359	0.556	0.933	0.927	0.328	0.519
VIFp	0.845	0.860	0.647	0.704	0.914	0.905	0.372	0.704
VIFp ($K = 6$)	0.926	0.924	0.456	0.704	0.921	0.921	0.357	0.556

It is noteworthy that this approach essentially considers model views that hold the subjects attention, since the more time a user spends on a particular view, the higher is its influence on the estimated average. Thus, at a first glance, the results counter our initial intuition regarding the identification of important views based on human interactivity. More importantly, though, they highlight the sensitivity of the image-based metrics on the viewing distance.

Note that the same analysis was repeated by considering unique frames from every navigation track, essentially considering only the views that were visualized, regardless of the time. Despite marginal gains, the performance was still lower than computing the metrics using $K = 6$, as reported in Table 7.5.

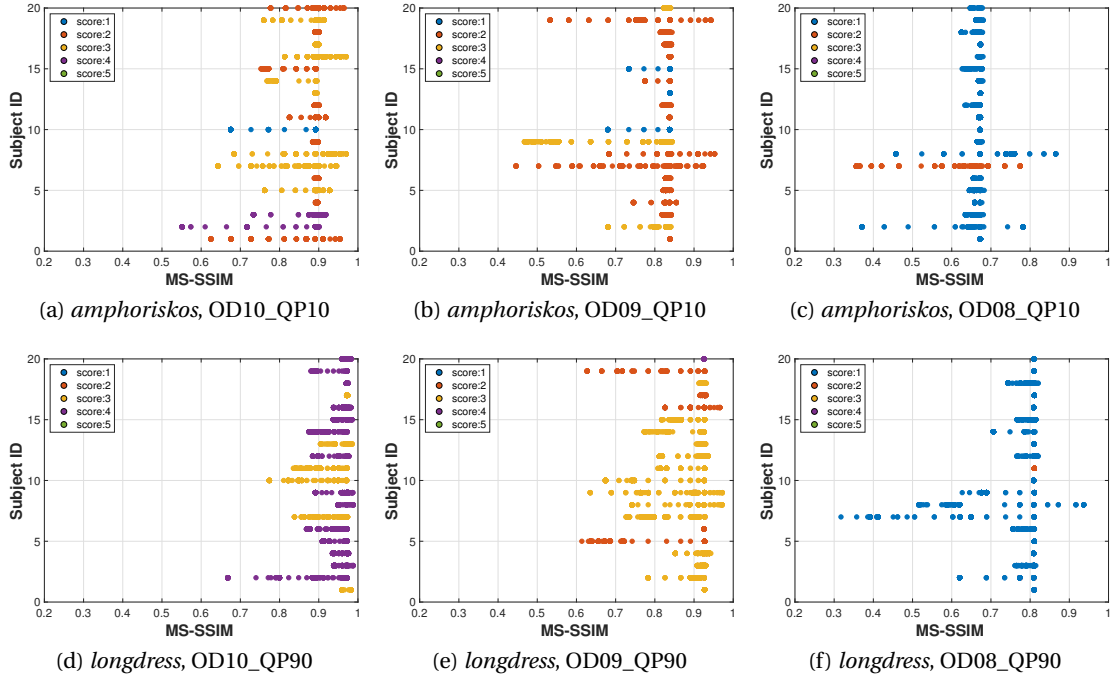


Figure 7.10 – MS-SSIM scores from the navigation tracks of every user inspecting a stimulus. In the first and second row, encoded versions of the model *amphoriskos* and *longdress* are depicted, respectively. Corresponding frontal views of the stimuli under consideration are provided in Figures 4.14 and 4.15. Remark that the evaluation of a given stimulus starts with the same frontal view for all the users; thus, the corresponding MS-SSIM value will be present for all subjects. Also note that at low quality levels, users are interacting less.

Importance weights

Table 7.6 provides the performance indexes obtained by excluding (AVG) and including (WAVG) user interactivity, over all camera layouts that were examined. The former denotes that the quality prediction of a model is obtained by averaging the individual objective scores that are computed based on the model views acquired from a given camera layout (i.e., reprinted from sub-section camera layouts of 7.1.3). The latter signifies the use of interactivity data in order to assign weights on the individual objective scores, before computing the weighted average as our prediction of visual quality for a model. As described in section 7.2.1, these weights are obtained after mapping the camera position of each view inspected by every user to the closest viewpoint of a given camera layout, and by considering the ratio of the aggregated time duration across all variations of a content divided by the total time of inspection. Note that the metrics are computed on the foreground of the images. Moreover, the MS-SSIM metric is only displayed, since it was found to be the best predictor across all tested cases. Similar trends are observed when using the VIFp.

According to our results, it is remarked that this method leads to substantial improvements with respect to the performance of the naive anchor paradigm that was described earlier

Chapter 7. Image-based objective quality metrics

Table 7.6 – Performance indexes of MS-SSIM per type of content, computed by including (WAVG) and excluding (AVG) user interactivity information on model views obtained under all camera layouts.

	Inanimate objects				Human figures			
	PLCC	SROCC	RMSE	OR	PLCC	SROCC	RMSE	OR
$K = 1$ (AVG)	0.951	0.944	0.373	0.519	0.952	0.935	0.279	0.519
$K = 1$ (WAVG)	0.951	0.944	0.373	0.519	0.952	0.935	0.279	0.519
$K = 6$ (AVG)	0.955	0.944	0.359	0.556	0.933	0.927	0.328	0.519
$K = 6$ (WAVG)	0.956	0.944	0.356	0.519	0.949	0.935	0.289	0.519
$K = 12$ (AVG)	0.949	0.944	0.381	0.519	0.926	0.920	0.344	0.519
$K = 12$ (WAVG)	0.951	0.949	0.376	0.556	0.943	0.935	0.303	0.519
$K = 42$ (AVG)	0.949	0.945	0.383	0.519	0.926	0.915	0.345	0.556
$K = 42$ (WAVG)	0.951	0.947	0.374	0.519	0.949	0.933	0.289	0.519
$K = 162$ (AVG)	0.949	0.945	0.384	0.519	0.925	0.915	0.347	0.519
$K = 162$ (WAVG)	0.949	0.942	0.382	0.556	0.948	0.936	0.290	0.519

and reported in Table 7.5. Moreover, when comparing the performance indexes by including and excluding importance weights, it is evident that equal and consistently better results are obtained for inanimate objects and human bodies, respectively, under any camera layout. For inanimate objects, marginal differences are observed under all configurations, whereas for human bodies, it is evident that the performance notably worsens by excluding interactivity data, as the number of views is increasing.

In Figure 7.11, the importance weights associated with every view on the camera layout with $K = 162$ are presented for every model. For contents that represent inanimate objects, subjects tend to allocate more time on views that are more informative, as indicated in Figures 7.11a-7.11c. For instance, greater weights are observed at viewpoints that are located on top of the *biplane* and the *romanoillamp* contents, and around the equator of *amphoriskos*, which is a rather symmetric model. For models that represent human figures, users consistently spend more time in frontal views, as can be seen in Figures 7.11d-7.11f. This outcome is in accordance with (Dutagaci et al., 2010), where subjects were explicitly asked to select the best view of a wide range of 3D models, in which a clear preference for frontal views in human bodies and faces is reported. This may explain why for human bodies data set the frontal view is found to be among the best configurations. In Figure 7.12, different views of two models with corresponding importance weights are indicatively presented.

In Table 7.7, the percentages of views with nonzero weights at every camera arrangement are reported, along with the average duration of inspection per content. Interestingly, subjects spent on average 30% – 40% less time with human bodies with respect to inanimate objects. In particular, *romanoillamp* and *loot* were inspected from the least number of views from the inanimate objects and human bodies sets, respectively. As can be seen, capturing and

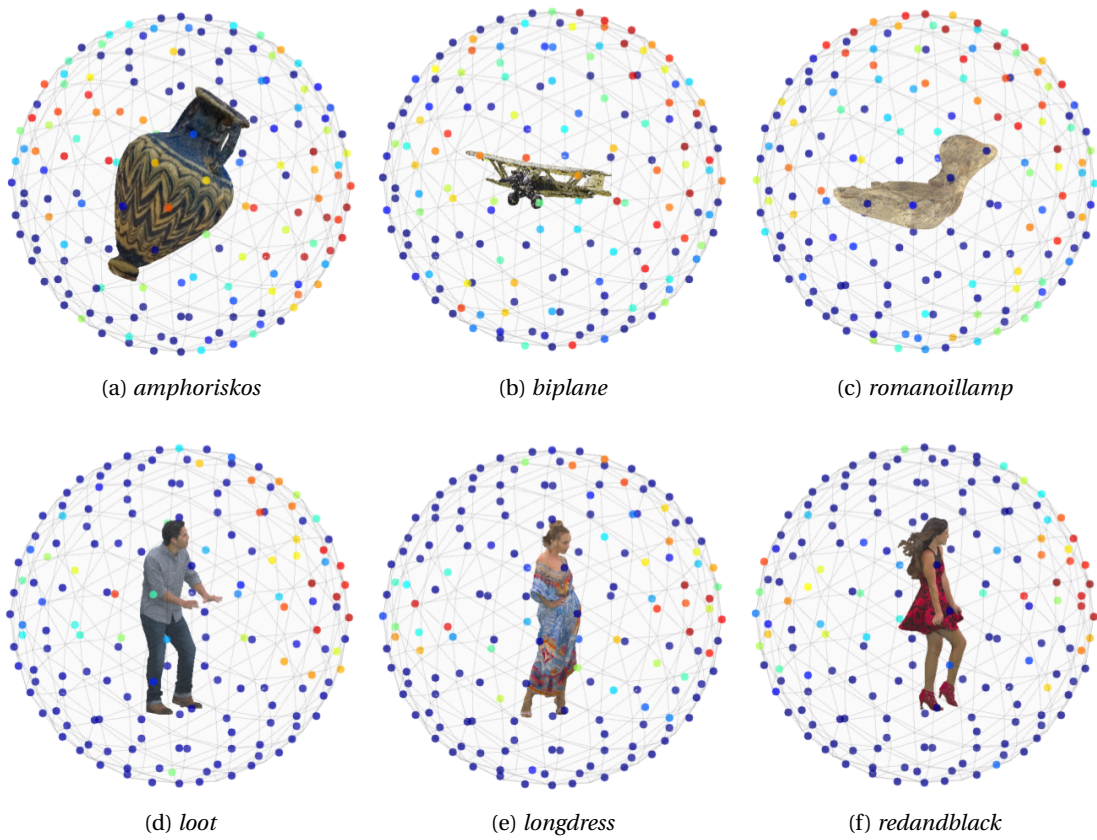


Figure 7.11 – Dot markers on the view sphere correspond to camera positions for a 2-level subdivision of an icosahedron ($K = 162$). The color code represents the ranking of weights, ranging from dark blue (minimum) to dark red (maximum).

computing the metrics on viewpoints with nonzero weights is beneficial performance- and complexity-wise, with greater gains as the number of viewpoints is increasing.

Table 7.7 – Percentage of viewpoints with non-zero weights under all camera layouts and average time of users inspection, per content.

	Inanimate objects			Human figures		
	<i>amphoriskos</i>	<i>biplane</i>	<i>romanoillamp</i>	<i>longdress</i>	<i>loot</i>	<i>redandblack</i>
$K = 6$	100%	100%	100%	100%	100%	100%
$K = 12$	100%	100%	100%	91.67%	91.67%	83.33%
$K = 42$	95.24%	92.86%	88.10%	57.14%	61.90%	64.29%
$K = 162$	64.81%	65.43%	61.73%	32.72%	32.10%	31.48%
Avg. time	16.47 sec	14.44 sec	14.01 sec	10.25 sec	9.19 sec	9.75 sec



Figure 7.12 – Views of model *biplane* on top and *loot* on the bottom, with corresponding importance weights.

7.3 Conclusions

In this chapter we investigate the performance of image-based objective quality metrics for point cloud contents. Our results suggest that they denote an effective solution that can outperform the prediction accuracy of point-based algorithms. Notably, the MS-SSIM was consistently identified as the best option among the metrics under examination, potentially due to its ability to simulate views from different distances.

As part of the study, we initially define a quality evaluation framework that can support camera layouts granting uniformly distributed positions on the view-sphere, from where projections of the model are acquired. We proceed by examining the generalization capabilities of image-based metrics using subjective opinions for the same point clouds, displayed under different

rendering schemes in two separate experiments. Benchmarking results show high prediction power in both cases, outperforming point-based counterparts. Moreover, better adjustability is observed with respect to the latter approaches; that is, smaller variations of correlation coefficients, which remain at high levels. On a second stage, the performance is improved by proposing the removal of background information from the computations. Furthermore this modification secures coherency, considering that the obtained scores are not affected by the presence of information that is irrelevant to the quality of the displayed model. Next, the integration of additional views for quality prediction is evaluated. Our results indicate that additional projections do not necessarily lead to further improvements, showing that even one view could be sufficient for good performance.

Finally, we investigate whether interactions from subjects could be exploited to improve the prediction accuracy of image-based metrics. For this purpose, the visual experiences of users evaluating point cloud stimuli were reproduced and objective quality metrics were computed considering all the displayed frames. It was shown that using the entire recorded information denotes a sub-optimal approach, both performance- and complexity-wise. On the contrary, exploiting the recorded interactivity in terms of salient viewing perspectives led to performance improvements. In particular, an alternative strategy was devised, assuming that more frequently visited views have a higher influence in the subjective opinions. Given a camera arrangement, model views obtained from the closest camera positions were weighted higher based on this assumption. Our results show that the proposed weighting function can be beneficial, as the prediction power of the objective quality metrics is improved, especially in the case of models representing human bodies in our data set.

8 Benchmarking of objective quality metrics

In this chapter, we evaluate the performance of popular objective quality metrics for point clouds, using a subjectively annotated data set that consists of diverse contents, encoded using the state-of-the-art MPEG test models. In particular, the point-to-point and point-to-plane (Tian et al., 2017b) variants, the color MSE and PSNR of the luminance and each chrominance component computed separately and combined based on a weighted average, comprise the so-called MPEG PCC metrics. Moreover, we consider the point-to-distribution based on the Mahalanobis distance (Javaheri et al., 2020c), the plane-to-plane (Alexiou and Ebrahimi, 2018c), the PCQM (Meynet et al., 2020), for which every feature and the proposed formula to obtain a global weighted average are evaluated separately, the PointSSIM (Alexiou and Ebrahimi, 2020), which is computed on best configurations per attribute-based features, and the PCM_RR (Viola and Cesar, 2020). Finally, popular image-based metrics are evaluated using projected views of the models that are obtained from different camera layouts.

This chapter is based on material that has been published in (Alexiou et al., 2019a).

8.1 Validation methodology

8.1.1 Data set

The subjectively annotated data set that is recruited in this performance evaluation analysis, labelled as M-PCCD, is assembled in the context of our efforts published in (Alexiou et al., 2019a). Briefly, the data set consists of 8 static colored point clouds with varying characteristics in terms of topological and textural compositions that represent both human figures and inanimate objects, namely, *amphoriskos* [10-bit], *biplane* [10-bit], *head* [9-bit], *longdress* [10-bit], *loot* [10-bit], *romanoillamp* [10-bit], *soldier* [10-bit], and *the20smaria* [10-bit]. Each pristine model is either by default (i.e., *longdress*, *loot* and *soldier*) or manually voxelized, with the voxel bit-depth resolution indicated in brackets. The contents are compressed using the MPEG encoding engines. In particular, the V-PCC, and the four G-PCC variants (i.e., Octree-plus-RAHT, Octree-plus-Lifting, TriSoup-plus-RAHT, TriSoup-plus-Lifting) are

employed and configured following the MPEG Common Test Conditions document (MPEG 3DG and Requirements, 2017). The subjective experiments were conducted in two dislocated laboratories. The participants rated the visual quality of the stimuli by means of an interactive evaluation platform in a desktop setting under the simultaneous DSIS test method, with the point clouds rendered using screen-faced splats of adaptive size. The rating populations collected in both laboratories exhibited strong correlation, as described in section 9.2.3, thus, they were pooled together in order to compute the MOS and the corresponding CIs that serve as our ground truth. We refer to details about the generation of stimuli and the subjective experiment in sections 9.1 and 9.2, respectively.

8.1.2 Computation of quality metrics

The implementation of the point-to-point and point-to-plane metrics relies on the software version 0.13.5 that is presented in (Tian et al., 2017c). Both the MSE and the Hausdorff distance (i.e., HSD) are used as pooling methods to produce global degradation scores from individual errors that are extracted from pairs of associated points. The geometric PSNR scores are additionally considered using the corresponding voxel grid resolution of each pristine model to obtain the peak value. For the color PSNR and MSE metrics, the color attributes are converted from the original RGB to the YCbCr color space, following the ITU-R Recommendation BT.709-6 (ITU-R BT.709-6, 2015), as implemented in the same software release. To compute a weighted average incorporating luminance and chrominance distortions, the Equation 8.1 is used, following (Ohm et al., 2012). Note that the same formulation is employed to also compute the weighted average of the color MSE scores (i.e., MSE_YUV).

$$\text{PSNR_YUV} = (6 \cdot \text{PSNR_Y} + \text{PSNR_U} + \text{PSNR_V}) / 8 \quad (8.1)$$

For each aforementioned metric, the symmetric error is adopted. Moreover, the default normal vectors that are associated with a pristine model are employed to compute the point-to-plane metric, when available (i.e., *longdress*, *loot*, and *soldier*). For the rest of the contents, normals are estimated based on plane fitting using 12 nearest neighbors, as implemented in PCL.

For the point-to-distribution metric, the source code provided in (Javaheri et al., 2020c) is executed. The default neighborhood size of 31 was set for the computations. To compute a global degradation score, both the mean and the MSE pooling methods are employed, named after MMD and MSMD, respectively. In both cases, the symmetric error provides the final predictions.

For the plane-to-plane metric, the version 1.0 of the software that is released with (Alexiou and Ebrahimi, 2018a) is employed. The normal vectors are estimated using quadric fitting with range search of radius equal to 30, following the results of our study in section 6.1. To compute a global angular similarity score, the average (i.e., AVG) and the MSE are employed as pooling methods and, for both cases, the symmetric error is used.

For the PointSSIM, the scripts that are released with (Alexiou and Ebrahimi, 2020) are employed. The PointSSIM is applied on location, normal, curvature and color point cloud attributes, using every dispersion estimator, neighborhood size and target voxel resolution presented in section 6.2. Moreover, following the results of the analysis, the impaired model is set as the reference (i.e., the metric is computed looping for every point of the pristine model) and average pooling is applied to compute a global degradation score. In this analysis, best-performing configurations per attribute are reported.

For PCQM and PCM_RR, the software coming with the studies (Meynet et al., 2020) and (Viola and Cesar, 2020), respectively, are employed as such.

Regarding image-based approaches, the same rendering settings that were adopted during subjective evaluation are employed, and projected views of the displayed point clouds are acquired. The exported bitmaps are 1024×1024 , which is identical to the resolution of the canvas that was used to present the models to the participants. The stimuli are captured from $K = 1, 6$, and 42 viewpoints, using the camera layout that is described in section 7.1.1. Note that the orientation of the contents from this data set is not aligned; thus, the frontal view of each model is manually selected for $K = 1$. Considering $K > 1$ viewpoints, a simple average, the RMS and the MSE were tested as pooling methods in order to obtain a global degradation score. Moreover, four approaches were examined to define a mask that determines the region of pixels over which the 2D metrics are computed: (a) the entire captured image without removing any background information, (b) the foreground of the projected reference, (c) the union and (d) the intersection of foregrounds of the projected pristine and impaired models.

The PSNR, SSIM (Wang et al., 2004), MS-SSIM (Wang et al., 2003), and VIFp (Sheikh and Bovik, 2006) (i.e., multi-scale in pixel domain) metrics are selected and evaluated. The implementations of the last 3 algorithms are based on open-source MATLAB scripts¹, which were modified to account for the application of the mask. All metrics are applied on the luminance channel, after conversion of the RGB color values to the YCbCr colorspace using the ITU-R Recommendation BT.709-6 (ITU-R BT.709-6, 2015). Based on our results, the union of foregrounds was found to outperform the alternatives and, hence, the performance indexes following this approach are reported. It is noteworthy that clear performance drops are observed when using the entire image, especially for the metrics PSNR, SSIM and MS-SSIM, suggesting that involving background pixels in the computations is not recommended, as also seen in the corresponding analysis of section 7.1.3. Regarding the pooling methods, minor differences were remarked with slight improvements when using the average score; thus, the latter is employed to report our results.

8.1.3 Benchmarking of quality metrics

The objective quality metrics are benchmarked against ground-truth subjective MOS, and performance indexes are computed to indicate their prediction power, as described in sec-

¹http://live.ece.utexas.edu/research/Quality/index_algorithms.htm

tion A.3. In this case, we enrich our analysis by considering additional fitting functions, as suggested in the Recommendation ITU-T J.149 (ITU-T J.149, 2004). In particular, a linear, a monotonic polynomial of third order, and a logistic model are employed, which are given by Equations 8.2, 8.3 and 8.4, respectively.

$$P(x) = a \cdot x + b \quad (8.2)$$

$$P(x) = a \cdot x^3 + b \cdot x^2 + c \cdot x + d \quad (8.3)$$

$$P(x) = a + \frac{b}{1 + \exp^{-c \cdot (x-d)}} \quad (8.4)$$

where a , b , c and d are determined using a least squares method for each regression model, separately. To evaluate the performance of an objective quality metric, the PLCC, SROCC, RMSE, and OR indexes are computed between MOS and $P(\text{MOS})$ under every fitting function.

8.2 Results

Entire data set

In Table 8.1 the performance indexes of our benchmarking analysis are reported, for each tested regression model. Note that values close to 0 indicate no-linear for PLCC and no-monotonic relationship for SROCC, while values close to 1 or -1 indicate high positive or negative correlation, respectively. RMSE and OR are ranging between 0 and 1, with lower values indicating higher accuracy and consistency, respectively. Also remark that in order to report the per-attribute configurations of PointSSIM, we use the following notation: [attribute, voxel depth, dispersion estimator, neighborhood size].

In Figure 8.1, scatter plots of subjective against objective quality scores are presented for a selection of metrics, in order to provide a visual illustration of their performance. Note that for point-to-plane with MSE, PCQM, MMD, and PCM_RR metrics, a closer view in a narrower range of the objective scores is provided.

According to the indexes of Table 8.1, the PointSSIM is the best-performing objective quality metric when applied on luminance-based features that are extracted using σ^2 as dispersion estimator in neighborhoods of 12 nearest points at a 9-bit voxel depth. Notably, the performance indexes of this predictor are not substantially affected by the regression model selection. This indicates that the metric follows a largely linear trend with respect to the subjective scores, a property that is desirable as it allows for easier differentiation among values. In contrast, performance drops with lower order polynomial fitting functions indicate that a metric doesn't exhibit a linear behavior and its energy is concentrated in a limited range of values.

Table 8.1 – Performance indexes computed over the entire data set.

		Linear				Cubic			Logistic		
		SROCC	PLCC	RMSE	OR	PLCC	RMSE	OR	PLCC	RMSE	OR
MPEG PCC metrics	point-to-point_HSD	-0.370	0.004	1.363	0.905	0.056	1.361	0.901	0.004	1.363	0.905
	point-to-point_MSE	0.868	0.484	1.193	0.858	0.691	0.985	0.841	0.845	0.728	0.841
	point-to-plane_HSD	0.505	0.207	1.334	0.875	0.279	1.309	0.884	0.672	1.009	0.866
	point-to-plane_MSE	0.884	0.448	1.219	0.862	0.663	1.021	0.841	0.858	0.700	0.832
	PSNR point-to-point_HSD	0.225	0.236	1.239	0.884	0.476	1.121	0.907	0.559	1.056	0.866
	PSNR point-to-point_MSE	0.759	0.679	0.935	0.833	0.723	0.880	0.801	0.720	0.885	0.819
	PSNR point-to-plane_HSD	0.382	0.405	1.165	0.866	0.511	1.095	0.931	0.511	1.095	0.921
	PSNR point-to-plane_MSE	0.807	0.711	0.896	0.833	0.757	0.833	0.833	0.756	0.834	0.852
	MSE_Y	0.662	0.407	1.246	0.884	0.525	1.160	0.892	0.656	1.029	0.853
	MSE_U	0.440	0.358	1.273	0.862	0.381	1.261	0.897	0.399	1.250	0.897
	MSE_V	0.624	0.314	1.294	0.884	0.380	1.261	0.888	0.604	1.086	0.853
	MSE_YUV	0.663	0.410	1.244	0.884	0.528	1.158	0.888	0.653	1.033	0.849
	PSNR_Y	0.662	0.654	1.031	0.884	0.670	1.012	0.879	0.671	1.011	0.871
	PSNR_U	0.440	0.432	1.229	0.892	0.451	1.217	0.888	0.428	1.232	0.897
	PSNR_V	0.624	0.587	1.103	0.884	0.597	1.094	0.888	0.604	1.086	0.862
	PSNR_YUV	0.660	0.646	1.040	0.879	0.654	1.032	0.866	0.653	1.033	0.849
	MMD	0.887	0.617	1.073	0.853	0.827	0.767	0.819	0.865	0.685	0.802
	MSMD	0.882	0.488	1.190	0.862	0.678	1.002	0.866	0.853	0.713	0.819
	PSNR MMD	0.822	0.740	0.918	0.845	0.790	0.836	0.828	0.789	0.837	0.832
	PSNR MSMD	0.837	0.753	0.898	0.832	0.795	0.827	0.810	0.799	0.821	0.819
PCQM	plane-to-plane_AVG	0.822	0.618	1.071	0.879	0.782	0.849	0.853	0.819	0.782	0.823
	plane-to-plane_MSE	0.831	0.648	1.039	0.875	0.800	0.819	0.836	0.827	0.766	0.819
	Curvature comparison	0.754	0.669	1.013	0.832	0.737	0.921	0.832	0.754	0.896	0.806
	Curvature contrast	0.805	0.707	0.964	0.841	0.777	0.857	0.810	0.792	0.832	0.815
	Curvature structure	0.771	0.718	0.950	0.871	0.769	0.871	0.823	0.760	0.887	0.832
	Lightness comparison	0.883	0.452	1.216	0.875	0.607	1.084	0.888	0.836	0.748	0.875
	Lightness contrast	0.908	0.669	1.013	0.888	0.830	0.761	0.802	0.865	0.685	0.793
	Lightness structure	0.891	0.829	0.762	0.823	0.887	0.630	0.789	0.883	0.640	0.806
	Chroma comparison	0.840	0.475	1.199	0.866	0.671	1.011	0.875	0.795	0.828	0.858
	Hue comparison	0.631	0.503	1.178	0.905	0.609	1.081	0.845	0.618	1.072	0.819
PointSSIM	PCMQ	0.915	0.607	1.083	0.866	0.829	0.763	0.845	0.899	0.596	0.750
	Location, orig., σ^2 , 12	0.835	0.782	0.850	0.828	0.820	0.780	0.819	0.828	0.765	0.754
	Normal, orig., σ , 48	0.745	0.718	0.949	0.853	0.742	0.914	0.849	0.745	0.910	0.810
	Curvature, orig., mAD , 24	0.871	0.782	0.850	0.853	0.817	0.787	0.849	0.854	0.710	0.797
	Luminance, 9-bit, σ^2 , 12	0.936	0.905	0.581	0.789	0.922	0.529	0.772	0.929	0.504	0.716
Image-based	PCM_RR	0.888	0.555	1.134	0.871	0.796	0.825	0.853	0.868	0.677	0.802
	PSNR [1-view]	0.621	0.589	1.102	0.853	0.604	1.086	0.836	0.651	1.035	0.871
	PSNR [6-views]	0.598	0.577	1.113	0.875	0.591	1.100	0.862	0.636	1.052	0.819
	PSNR [42-views]	0.628	0.597	1.093	0.871	0.611	1.079	0.858	0.667	1.015	0.802
	SSIM [1-view]	0.649	0.617	1.073	0.888	0.644	1.043	0.875	0.625	1.065	0.884
	SSIM [6-views]	0.636	0.611	1.079	0.879	0.635	1.053	0.858	0.615	1.076	0.884
	SSIM [42-views]	0.633	0.609	1.081	0.879	0.636	1.052	0.862	0.613	1.078	0.871
	MS-SSIM [1-views]	0.731	0.610	1.080	0.892	0.675	1.005	0.884	0.667	1.017	0.884
	MS-SSIM [6-views]	0.746	0.618	1.072	0.849	0.698	0.976	0.875	0.692	0.985	0.879
	MS-SSIM [42-views]	0.752	0.623	1.067	0.862	0.701	0.972	0.879	0.694	0.982	0.888
	VIFp [1-view]	0.714	0.675	1.006	0.845	0.685	0.993	0.866	0.670	1.012	0.849
	VIFp [6-views]	0.734	0.690	0.987	0.841	0.708	0.963	0.836	0.690	0.988	0.858
	VIFp [42-views]	0.742	0.697	0.978	0.853	0.716	0.951	0.823	0.698	0.977	0.858

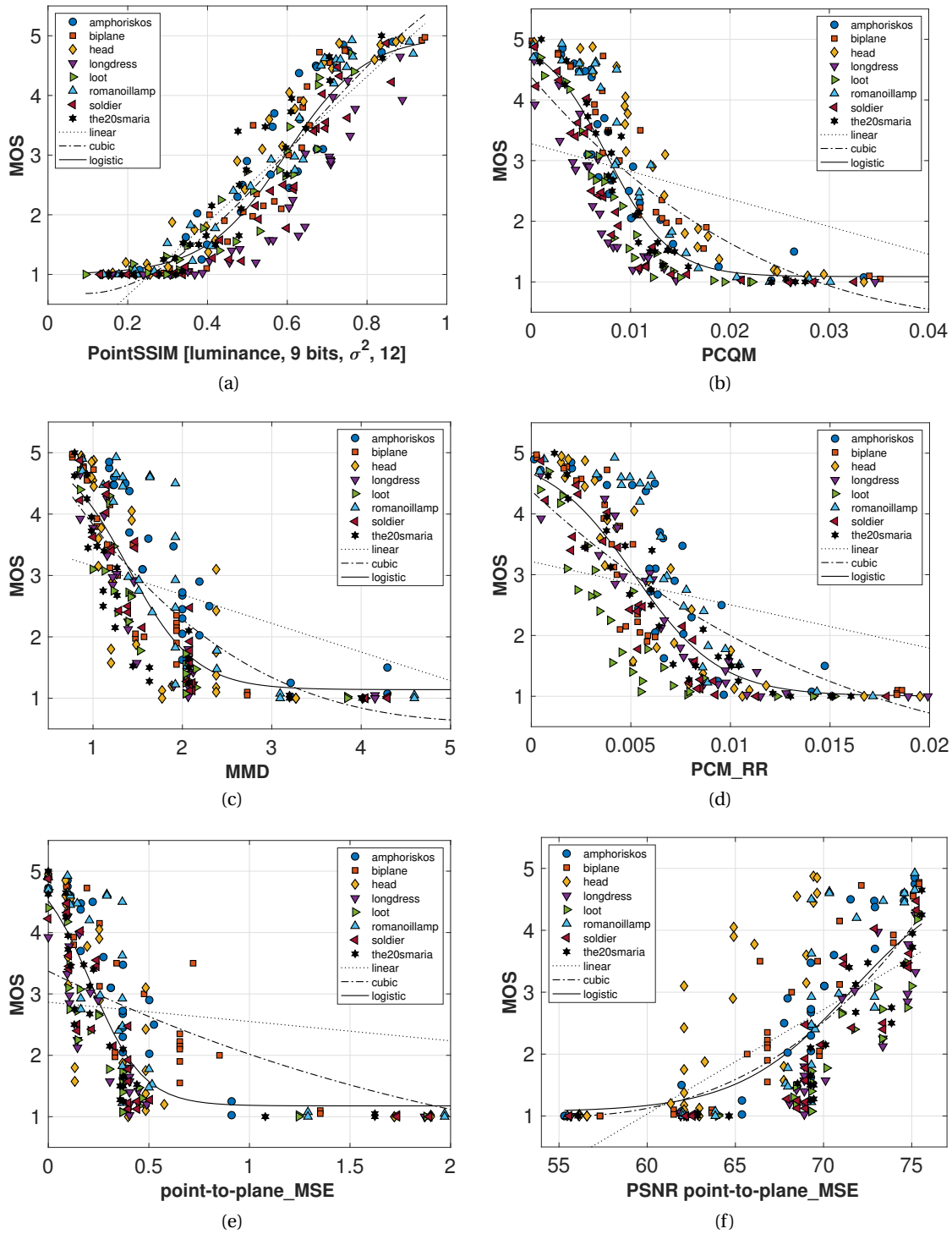


Figure 8.1 – Subjective against objective scores for a selection of metrics, considering the entire data set. A zoomed view is provided for PCQM, MMD, PCM_RR and point-to-plane with MSE.

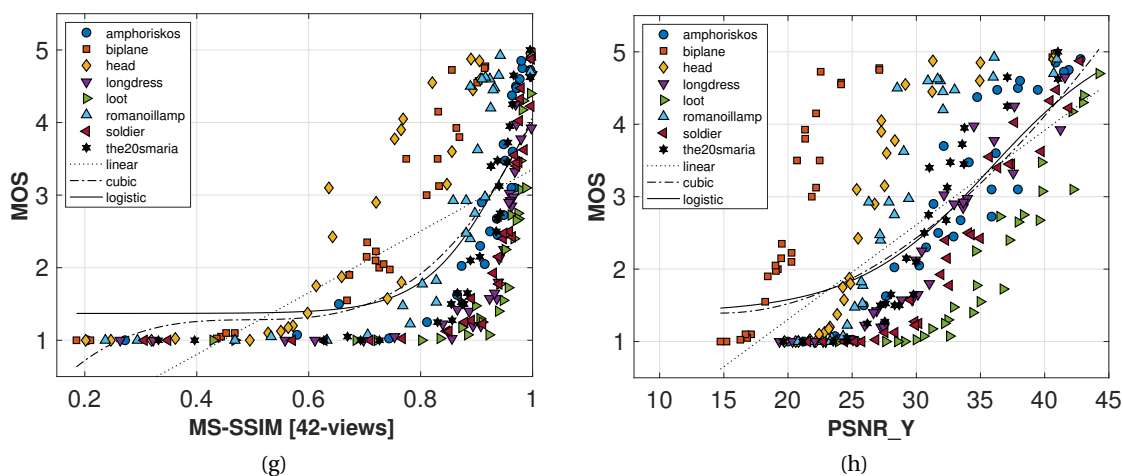


Figure 8.1 – Subjective against objective scores for a selection of metrics, considering the entire data set.

The PCQM and, particularly, the lightness-based features were found to achieve similarly high performance with PointSSIM, validating the effectiveness of such measurements to capture visual impairments. Moreover, the performance of PCM_RR is noteworthy, especially considering that it denotes a reduced reference metric. The results imply the efficiency of the global feature maps that are proposed as part of this method.

Considering color-only predictors that rely in simple point-to-point comparisons, limited performance is observed. Among these metrics, the PSNR computed on the luminance is better correlated with subjective opinions. Marginal differences are identified when compared to the weighted average performed by PSNR_YUV that also involves chrominance components. Based on Figure 8.1h, we observe that PSNR_Y provides rather accurate predictions on a per content basis; however, its generalization capabilities are poor, showing the inability of the metric to adapt to different topologies.

On this aspect, our results indicate the great benefits that are brought by approaches that rely on local pooling for the extraction of color-related features, with respect to point-to-point comparison. That is, the formulation of neighborhoods incorporates a spatial dimension to the measurements. Thus, related statistics not only assess textural information, but also carry underlying geometric distortions in an implicit manner. This can explain the higher performance of the luminance-based features of PointSSIM and PCQM, with respect to the more simplistic comparisons that are performed in PSNR_Y.

Regarding geometry-only predictors, the MMD was found the best choice, closely followed by the point-to-plane with MSE, the curvature-based features of PointSSIM and the point-to-point with MSE. This result suggests that taking under consideration the correlations among the distribution of points in a reference neighborhood in order to quantify the error of an

impaired point, has benefits with respect to capturing deviations from a reference position, or a planar surface approximation, as implemented from the point-to-point and point-to-plane methods, respectively. The most evident gains are visible in the form of higher robustness against the selection of a regression method. The same properties are observed from the curvature-based features of PointSSIM, which show even less PLCC divergences across fitting functions, indicating a stronger linear relationship. A similar observation can also be made for the slightly worse-performing location-based features of PointSSIM, the plane-to-plane and the curvature-based statistics of PCQM.

Considering PSNR variants of geometric-only predictors, it is noteworthy that they show less deviations across fitting functions in relation to the non-normalized errors. However, their performance is consistently worse under the logistic fitting. Note that the content *head* is of lower voxel resolution, which leads the PSNR-based metrics to map the corresponding scores to a lower quality range, which doesn't properly reflect the judgement of subjects. This trend is obvious when comparing the scatter plots of point-to-plane with MSE and its PSNR variation that are shown in Figures 8.1e and 8.1f, respectively.

It should be remarked that for the majority of metrics that assess geometry-only distortions, despite the high values observed for linearity and monotonicity indexes, the lower performance in RMSE and OR implies lower accuracy and consistency of the predictions. This is reasonable given that the data set consists of colored models, and in particular considering that it includes stimuli with the same geometry information and different color artifacts (e.g., stimuli encoded with Octree-plus-Lifting and Octree-plus-RAHT at the same degradation level). In such cases, geometry-only methods would assign the same score since they ignore color impairments. Despite such limiting factors, the overall performance indexes suggest good prediction power across the entire data set. Notice that the simultaneous quality reduction of the stimuli in both geometry and color information at each higher degradation level, assists to the obtained results.

Regarding image-based metrics, it is evident that the performance is substantially lower when compared to state-of-the-art point-based counterparts. The best predictions are achieved by the MS-SSIM algorithm for this class of algorithms. The additional views were not found to crucially alter the performance of the algorithms, confirming previous observations made in section 7.1. In Figure 8.1g, we observe that the MS-SSIM provides good results per content, however, performs poorly in generalizing predicted scores across different contents.

Per codec

To obtain further insights regarding the performance of the metrics, we repeat our benchmarking analysis after splitting the data set per stimuli compressed by the same codec. In Table 8.2, the PLCC and SROCC performance indexes for a sub-set of representative metrics is reported.

It can be observed that PointSSIM achieves the best results in all cases, with luminance-based

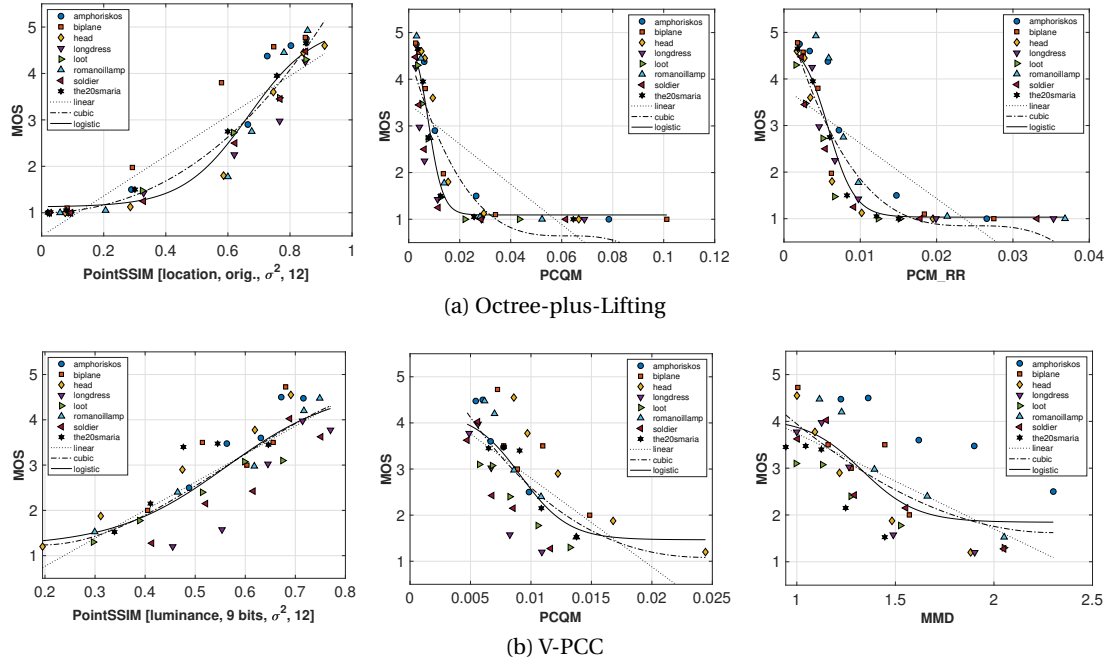


Figure 8.2 – Subjective against objective scores for a selection of metrics, considering stimuli clustered per codec.

Table 8.2 – Performance indexes computed over stimuli clustered per codec.

	Octree+Lifting		Octree+RAHT		TriSoup+Lifting		Trisoup+RAHT		V-PCC	
	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC
point-to-point_MSE	0.934	0.922	0.930	0.931	0.885	0.828	0.824	0.781	0.460	0.420
point-to-plane_MSE	0.921	0.931	0.918	0.936	0.909	0.871	0.856	0.839	0.735	0.688
PSNR point-to-point_MSE	0.880	0.900	0.846	0.884	0.539	0.591	0.589	0.486	0.299	0.282
PSNR point-to-plane_MSE	0.867	0.897	0.836	0.876	0.653	0.673	0.707	0.546	0.603	0.553
PSNR_Y	0.726	0.723	0.714	0.726	0.789	0.658	0.734	0.632	0.376	0.333
PSNR_YUV	0.723	0.724	0.713	0.735	0.770	0.645	0.715	0.623	0.324	0.316
MMD	0.895	0.889	0.892	0.893	0.922	0.907	0.879	0.875	0.718	0.690
plane-to-plane_MSE	0.908	0.903	0.876	0.881	0.886	0.813	0.800	0.754	0.635	0.553
PCM_Q	0.941	0.932	0.939	0.923	0.909	0.893	0.884	0.895	0.748	0.736
PointSSIM [location, orig., σ^2 , 12]	0.960	0.951	0.939	0.937	0.824	0.722	0.768	0.649	0.273	0.142
PointSSIM [normal, orig., σ , 48]	0.812	0.852	0.787	0.829	0.850	0.763	0.769	0.690	0.657	0.644
PointSSIM [curvature, orig., mAD , 24]	0.941	0.932	0.919	0.918	0.870	0.826	0.796	0.747	0.675	0.653
PointSSIM [luminance, 9 bits, σ^2 , 12]	0.953	0.943	0.955	0.948	0.942	0.932	0.941	0.948	0.858	0.876
PCM_RR	0.934	0.931	0.918	0.926	0.870	0.822	0.865	0.877	0.716	0.648
MS-SSIM [1-view]	0.802	0.807	0.777	0.812	0.820	0.700	0.748	0.664	0.282	0.348
MS-SSIM [42-views]	0.832	0.831	0.834	0.831	0.814	0.708	0.746	0.690	0.305	0.354

features showing superior performance under the majority of codecs, with notable differences in the case of V-PCC. Based on the performance indexes, visual quality prediction of models that are compressed by this codec is a bottleneck for all metrics, which can be explained by the types of impairments that are introduced. In particular, V-PCC leads to coarser local surface

approximations as a result of the patch-based encoding and, rather frequently, to a larger number of encoded points with respect to the original. Note that point count reductions facilitate the performance of the metrics, and denote distortions that are easier to be captured. This can be confirmed by the substantial improvements that are observed by all methods against Octree-based codecs. The latter leads to regular down-sampling and models of higher sparsity as the degradation level is increasing. Based on Table 8.4, we remark a plethora of metrics with high performance, with PCQM, point-to-point, point-to-plane and PCM_RR, closely following PointSSIM features and achieving very accurate results.

Considering TriSoup-based distortions, the same set of metrics provides good predictions, with the addition of MMD. Yet, the overall performance drops, with respect to the Octree-based counterpart. Interestingly, the point-to-plane behaves better than the point-to-point against these artifacts, whereas in the case of Octree-based distortions the opposite is true. The planar surface approximations that are employed to reconstruct the topology of the encoded models lead to visual impairments, that are not as effectively captured by the point-to-point approaches.

Finally, we observe that the performance of image-based metrics is poor across all codecs excluding Octree-based, stemming from the lack of generalization capabilities.

In Figure 8.2, scatter plots depicting subjective against objective scores from the three best-performing metrics are provided for stimuli encoded with Octree-plus-Lifting and V-PCC. For the former, smooth trends are observed for all three algorithms, confirming the high performance indexes. The latter plots, on the other hand, clearly indicate the performance decline of the metrics in the presence of V-PCC distortions. Focusing on the best-performing PointSSIM and PCQM, distinct trends per content are observed, indicating the difficulty of metrics to generalize quality predictions for this codec.

Per content type

We continue our analysis by splitting the data per type of content (i.e., human figures and inanimate objects), following the same approach we adopted in chapter 7. The majority of the human figure models have been captured by the same equipment, thus, they exhibit the same acquisition noise. On the other hand, such artifacts in the case of the inanimate objects class are more diverse. Moreover, from previous experimentation (see section 4.2.3), we have observed that subjects judge more critically distortions on human figures.

In the attached Table 8.3, the performance indexes of a representative sub-set of metrics is reported, against stimuli clustered per type of content. It is noteworthy that by splitting the data set in such a manner, the performance of all the metrics is remarkably improved, especially in the case of human figures.

The PointSSIM using color-based features and the PCQM are found to outperform the alternatives in the inanimate objects and the human figures set, respectively. In particular,

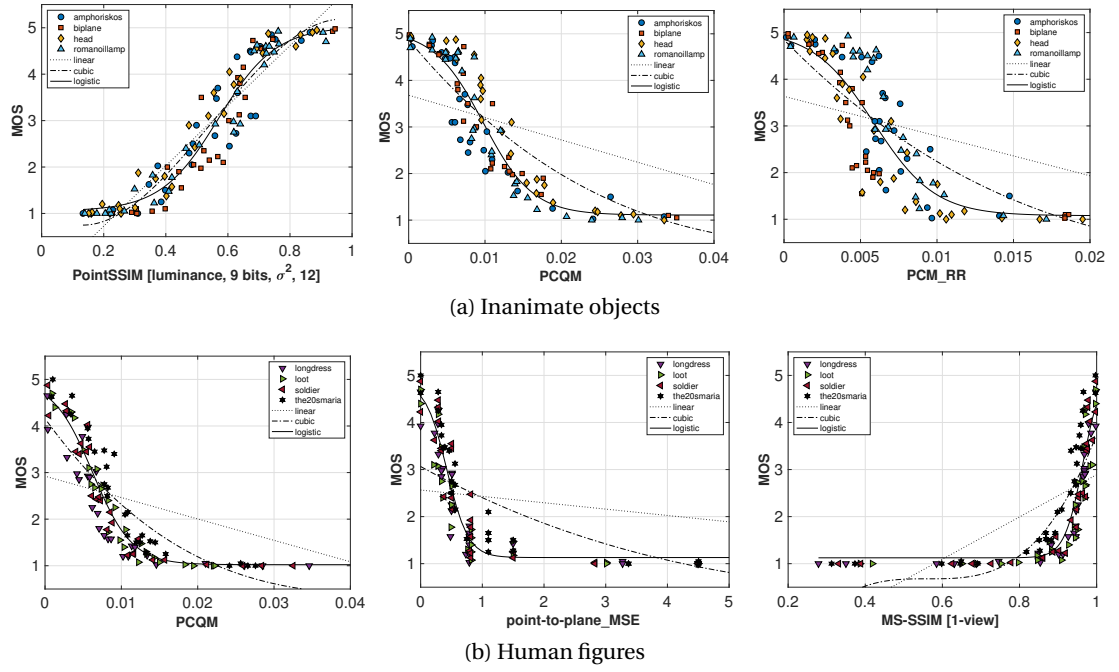


Figure 8.3 – Subjective against objective scores for a selection of metrics, considering stimuli clustered per type of content.

Table 8.3 – Performance indexes computed over stimuli clustered per type of content.

	Inanimate objects				Human figures			
	PLCC	SROCC	RMSE	OR	PLCC	SROCC	RMSE	OR
point-to-point_MSE	0.884	0.895	0.672	0.793	0.942	0.937	0.412	0.724
point-to-plane_MSE	0.869	0.892	0.711	0.810	0.933	0.941	0.441	0.664
PSNR point-to-point_MSE	0.795	0.814	0.834	0.824	0.926	0.924	0.413	0.694
PSNR point-to-plane_MSE	0.815	0.840	0.796	0.778	0.910	0.929	0.452	0.667
PSNR_Y	0.729	0.727	0.983	0.853	0.869	0.887	0.607	0.741
PSNR_YUV	0.738	0.742	0.968	0.862	0.839	0.855	0.667	0.750
MMD	0.857	0.871	0.739	0.793	0.934	0.931	0.436	0.690
plane-to-plane_MSE	0.774	0.779	0.908	0.879	0.911	0.925	0.504	0.690
PCMQ	0.952	0.958	0.439	0.586	0.949	0.960	0.385	0.612
PointSSIM [location, orig., σ^2 , 12]	0.770	0.770	0.917	0.845	0.921	0.905	0.478	0.707
PointSSIM [normal, orig., σ , 48]	0.679	0.709	1.054	0.836	0.849	0.797	0.648	0.784
PointSSIM [curvature, orig., mAD , 24]	0.813	0.840	0.836	0.767	0.893	0.913	0.552	0.750
PointSSIM [luminance, 9 bits, σ^2 , 12]	0.961	0.965	0.399	0.629	0.913	0.924	0.501	0.741
PCM_RR	0.881	0.896	0.679	0.681	0.911	0.917	0.506	0.707
MS-SSIM [1-view]	0.798	0.817	0.865	0.905	0.917	0.940	0.488	0.750
MS-SSIM [42-views]	0.829	0.846	0.802	0.845	0.917	0.937	0.488	0.741

PCQM is found to attain high performance, which is stable among both clusters. Considering the set of objects, the PCM_RR is ranked below PCQM with substantial differences, and is closely followed by the point-to-point and point-to-plane with MSE. In the human figures data set, which is comprised of contents with more regular topology, several metrics are excelling. Interestingly, the performance of PointSSIM is dropping, showing better performance in the set of objects.

After splitting the stimuli per type of content, the image-based predictors are found to be competitive with respect to the point-based alternatives. Interestingly, for human figures the best performance is attained using 1 view, whereas for inanimate objects, $K = 42$ views perform better, using MS-SSIM and logistic function in both cases, validating the trends that were observed in the results of section 7.1.

A reason behind the limited performance of image-based methods when considering the entire data set, is assumed to be the different levels and types of noise among the reference point cloud representations. Typical distortions due to acquisition may lead to the presence of noisy geometric structures, missing regions, or color noise and, thus, in reference models of varying quality. On top of that, compression degradations have a different impact in each content, which prevents generalization. Moreover, despite the fact that image-based metrics capture visual artifacts as displayed by the renderer, they are not optimized for impairments that occur due to geometry alterations, since they are tested and optimized to capture degradation in natural images. Computations in a pixel-by-pixel basis (or small pixel neighborhoods) will naturally be affected, without necessarily reflecting the impact of perceived distortions. This is especially true for point clouds that represent objects in this data set, whose geometry is rather irregular (e.g., *biplane*, *head*), and can be confirmed by Figure 8.1g.

Per content

Finally, in the last step of our analysis, we compute the performance indexes after splitting the data set per content. That is, correlation coefficients are computed over all encoded versions of each content, separately. Then, the average and the standard deviation of the performance indexes are computed across all contents, and reported in Table 8.4.

Based on our results, the best predictions per content are provided by PSNR_Y and PCQM, closely followed by the image-based MS-SSIM, luminance-based features of PointSSIM and PCM_RR. Interestingly, the performance of PSNR_Y surpasses more sophisticated solutions of higher complexity in this task. The prediction accuracy of this method per content can be confirmed by the corresponding trends that are presented in Figure 8.1h. This outcome highlights the efficiency of luminance-based measurements in evaluating the level of visual impairments, and underlines the limitations of point-to-point attribute comparisons in generalizing across different topologies.

Consulting Table 8.4, substantial gains are reported in the performance of the image-based

Table 8.4 – Performance indexes computed over each content’s variations (mean \pm standard deviation).

	PLCC	SROCC	RMSE	OR
point-to-point_MSE	0.930 ± 0.042	0.939 ± 0.027	0.480 ± 0.181	0.698 ± 0.118
point-to-plane_MSE	0.915 ± 0.043	0.924 ± 0.029	0.535 ± 0.172	0.677 ± 0.119
PSNR point-to-point_MSE	0.919 ± 0.049	0.926 ± 0.034	0.479 ± 0.196	0.681 ± 0.106
PSNR point-to-plane_MSE	0.900 ± 0.050	0.907 ± 0.036	0.537 ± 0.182	0.667 ± 0.134
PSNR_Y	0.971 ± 0.015	0.976 ± 0.010	0.313 ± 0.079	0.565 ± 0.082
PSNR_YUV	0.969 ± 0.017	0.972 ± 0.014	0.320 ± 0.084	0.547 ± 0.050
MMD	0.928 ± 0.044	0.932 ± 0.035	0.489 ± 0.179	0.733 ± 0.086
plane-to-plane_MSE	0.913 ± 0.043	0.914 ± 0.040	0.540 ± 0.168	0.681 ± 0.107
PCM_Q	0.972 ± 0.015	0.972 ± 0.010	0.305 ± 0.079	0.491 ± 0.126
PointSSIM [location, orig., σ^2 , 12]	0.868 ± 0.107	0.845 ± 0.148	0.630 ± 0.279	0.716 ± 0.110
PointSSIM [normal, orig., σ , 48]	0.803 ± 0.092	0.768 ± 0.094	0.786 ± 0.199	0.780 ± 0.080
PointSSIM [curvature, orig., mAD , 24]	0.868 ± 0.075	0.886 ± 0.064	0.642 ± 0.216	0.724 ± 0.092
PointSSIM [luminance, 9 bits, σ^2 , 12]	0.963 ± 0.018	0.960 ± 0.013	0.350 ± 0.087	0.621 ± 0.107
PCM_RR	0.949 ± 0.031	0.946 ± 0.029	0.412 ± 0.117	0.629 ± 0.122
MS-SSIM [1-view]	0.952 ± 0.021	0.966 ± 0.012	0.407 ± 0.108	0.647 ± 0.097
MS-SSIM [42-views]	0.962 ± 0.020	0.969 ± 0.011	0.364 ± 0.119	0.586 ± 0.104

metrics. In fact, our results suggest that estimating the visual quality of a compressed model using even 1 view, can lead to very accurate estimations. This can be explained by the fact that the image-based metric are able to capture progressively higher distortions with respect to the same reference content; that is, for each content, the monotonicity of the distortion scores is followed closely. However, the same distortion score might be mapped to different quality levels when considering other contents. Our results coincide with observations of (Lavoué et al., 2016) in regard to the performance of image-based metrics on geometric mesh models, showing that correlation results are very high when considering variations of a content under a single degradation type, while substantially decreasing otherwise.

8.3 Conclusions

In this section, existing point cloud objective quality metrics are rigorously benchmarked, using subjectively annotated models that were encoded with the state-of-the-art MPEG codecs. Our results indicate that the PointSSIM provides accurate predictions under all testing conditions that was examined in this data set, with PCQM attaining an equivalently high performance. Luminance-based statistics were found to provide consistently accurate results, as proven by the accuracy of the corresponding measurements that are part of both aforementioned metrics. Moreover, results show that capturing local changes is the most promising approach, showing remarkable improvements over earlier-developed metrics that rely on point-to-point comparisons. Metrics that make use of a combination of the above, showed

the best performance in this data set. This outcome can be explained by their underlying working principle. That is, luminance-based measurements are effective in quantifying color degradations, while geometric distortions are implicitly captured by the formulation of local neighborhoods in order to extract local features. Following our results, this is an efficient way to combine geometrical and textural distortions, leading to highly-performing predictors that are able to generalize across different topologies.

Towards efficient compression **Part III**

9 Benchmarking of MPEG codecs

In view of the increasing progress and development of 3-D scanning and rendering devices, acquisition and display of free viewpoint video (FVV) has become viable (Alexiadis et al., 2013; Collet et al., 2015; Schwarz et al., 2019). This type of visual data representation describes 3D scenes through geometry information (shape, size, position in 3D space) and associated attributes (e.g., color, reflectance), plus any temporal changes. FVV can be displayed in head-mounted devices, unleashing a great potential for innovations in XR applications. Industrial partners and manufacturers have expressed relevant interest in extending technologies available in consumer market with the possibility to represent real world scenarios in three dimensions. The development of immersive information and communication systems (e.g., tele-presence), 3D sensing for smart cities, robotics, and autonomous driving, are just some of the possibilities that can be envisioned to dominate in the near future.

There are several alternatives of advanced content representations that could be employed in such application scenarios. Point cloud imaging is well-suited for richer simulations in real-time because of the relatively low complexity and high efficiency in capturing, encoding and rendering of 3D models. Yet, the vast amount of information that is typically required to represent this type of content, indicates the necessity for efficient data representations and compression algorithms. Lossy compression solutions, although able to drastically reduce the amount of data and by extension the costs in processing, storage, and transmission, come at the expense of visual degradations. In order to address the trade-off between data size and visual quality, or more generally to evaluate the efficiency of an encoding solution, quality assessment of decompressed contents is of paramount importance. In this context, visual quality can be assessed through either objective or subjective means. The former is performed by algorithms that provide predictions, while the latter, although costly and time-consuming, is widely accepted to unveil the ground-truth for the perceived quality of a degraded model.

In this study, the objective is to benchmark the state-of-the-art MPEG point cloud encoding engines, using subjective quality assessment methodologies. In particular, a diverse set of point cloud contents is initially recruited and prepared for encoding. Then, a large-scale performance evaluation study is carried for geometry and color compression algorithms as



Figure 9.1 – Reference point cloud models. The set of objects is presented in the first row, whilst the set of human figures is illustrated in the second row.

Table 9.1 – Summary of content retrieval information, processing, and point specifications.

	Content	Repository	Pre-processing	Voxelization	Voxel dept	Input points	Output points
Objects	amphoriskos	Sketchfab	✓	✓	10 -bit	147.420	814.474
	biplane	JPEG	✗	✓	10-bit	106.199.111	1.181.016
	head	MPEG	✗	✓	9-bit	14.025.710	938.112
	romanoillamp	JPEG	✓	✓	10-bit	1.286.052	636.127
Human	longdress	MPEG	✗	✗	10 -bit	857.966	857.966
	loot	MPEG	✗	✗	10 -bit	805.285	805.285
	redandblack	MPEG	✗	✗	10 -bit	757.691	757.691
	soldier	MPEG	✗	✗	10 -bit	1.089.091	1.089.091
	the20smaria	MPEG	✗	✓	10 -bit	10.383.094	1.553.937

implemented in V-PCC (Mammou, 2017) and G-PCC (Mammou et al., 2019) test models using the MPEG Common Test Conditions (MPEG 3DG, 2017). Furthermore, different rate allocation schemes for geometry and texture encoding are analyzed and tested to draw conclusions on the best-performing approach in terms of perceived quality for a given bit-rate. The results of such a comprehensive evaluation provide useful insights for future development, or improvements of existing compression solutions.

This chapter is based on material that has been published in (Alexiou et al., 2019a).

9.1 Data set preparation

9.1.1 Content selection

A total of 9 static models are used in the experiments. The selected models denote a representative set of point clouds with diverse characteristics in terms of geometry and color details, with the majority of them being considered in recent activities of the JPEG and MPEG committees. The contents depict either human figures, or inanimate objects. The former set of point clouds consists of the *longdress* (longdress_vox10_1300), *loot* (loot_vox10_1200), *redandblack* (redandblack_vox10_1550) (Eon et al., 2017), *soldier* (soldier_vox10_0690), and *the20smaria* (HHI_The20sMaria_Frame_00600) models, which were obtained from the MPEG repository¹ and were provided by corresponding contributions (Eon et al., 2017; Ebner, 2018). The latter set is composed by *amphoriskos*, *biplane* (1x1_Biplane_Combined_000), *head* (Head_00039), and *romanoillamp*. The first model was retrieved from the online platform Sketchfab², the second and the last were selected from the JPEG Pleno repository³, while *head* was recruited from the MPEG database.

Such point clouds are typically acquired when objects are scanned by sensors that provide either directly or indirectly a cloud of points with information representing their 3D shapes. Typical use cases involve applications in desktop computers, hand-held devices, or head-mounted displays, where the 3D models are consumed outer-wise.

9.1.2 Content preparation

The selected codecs under assessment handle solely point clouds with integer coordinates. Thus, models that have not been provided as such in the selected databases were manually voxelized after eventual pre-processing. In particular, the contents *amphoriskos* and *romanoillamp* were initially pre-processed. For *amphoriskos*, the resolution of the original version is rather low; hence, to increase the quality of the model representation, the screened Poisson surface reconstruction algorithm (Kazhdan and Hoppe, 2013) was applied and a point cloud was generated by sampling the resulting mesh. The CloudCompare software was used with the default configurations of the algorithm and 1 sample per node, while the normal vectors that were initially associated to the coordinates of the original model were employed. From the reconstructed mesh, a target of 1 million points was set and obtained by randomly sampling a fixed number of samples on each triangle, resulting in a point cloud with irregular geometry. Regarding *romanoillamp*, the original model is essentially a polygonal mesh object. A point cloud version was produced by discarding any connectivity information and maintaining the original points' coordinates and color information.

In a next step, contents with non-integer coordinates are voxelized; that is, quantization of

¹<http://mpegfs.int-evry.fr/MPEG/PCC/DataSets/pointCloud/CfP/>, last accessed 01/2020

²<https://bit.ly/3nekULm>, last accessed 12/2020

³<https://jpeg.org/plenodb/>, last accessed 12/2020

coordinates, which leads to a regular down-sampling of geometry, while the color is obtained after sampling among the points that fall in each voxel to avoid texture smoothing, thus, leading to more challenging encoding conditions (see annex B.2 for implementation details). Voxel grids of 10-bit depth resolution are used for the contents *amphoriskos*, *biplane*, *romanoillamp* and *the20smaria*, whereas a 9-bit depth voxel grid is employed for *head*. It should be noted that, although a voxelized version of the latter model is provided in the MPEG repository, the number of output points is too large, making its usage cumbersome in the interactive rendering platform that was employed for subjective quality assessment. For this purpose, it was decided to use a smaller bit depth for this content. Another remark worth making is that for our tests design, it was considered important to eliminate influencing factors that are related to the sparsity of the models and would affect the visual quality of the rendered models. For instance, visual impairments naturally arise by assigning larger splats on models with lower resolutions, when visualization of watertight surfaces is required. At the same time, the size of the model, directly related to the number of points, should allow high responsiveness and fast interactivity in a rendering platform.

Representative poses of the reference contents after the preparation steps detailed above are shown in Figure 9.1, while related information is summarized in Table 9.1.

9.1.3 Degradation types

The model degradations under study were derived from the application of lossy compression. The contents were encoded using the latest versions of the state-of-the-art compression techniques for point clouds at the time of this writing, namely version 5.1 of V-PCC (Mammou, 2017) and version 6.0-rc1 of G-PCC (Mammou et al., 2019). The configuration of the encoders for our experiments respects the guidelines detailed in the MPEG Common Test Conditions document (MPEG 3DG, 2017). Below, a brief summary of the working principle of each test model and encoding module is presented.

Video-based Point Cloud Compression: V-PCC, also known as TMC2 (Test Model Category 2), takes advantage of already deployed 2D video codecs to compress geometry and texture information of dynamic point clouds (or Category 2). V-PCC's framework depends on a *Pre-processing* module, which converts the point cloud data into a set of different video sequences, as shown in Figure 9.2.

In essence, two video sequences, one for capturing the geometry information of the point cloud data (padded geometry video) and another for capturing the texture information (padded texture video), are generated and compressed using HEVC (Bross et al., 2012), the state-of-the-art 2D video codec. Additional metadata (occupancy map and auxiliary patch info) needed to interpret the two video sequences are also generated and compressed separately. The total amount of information is conveyed to the decoder in order to allow for the decoding of the compressed point cloud.

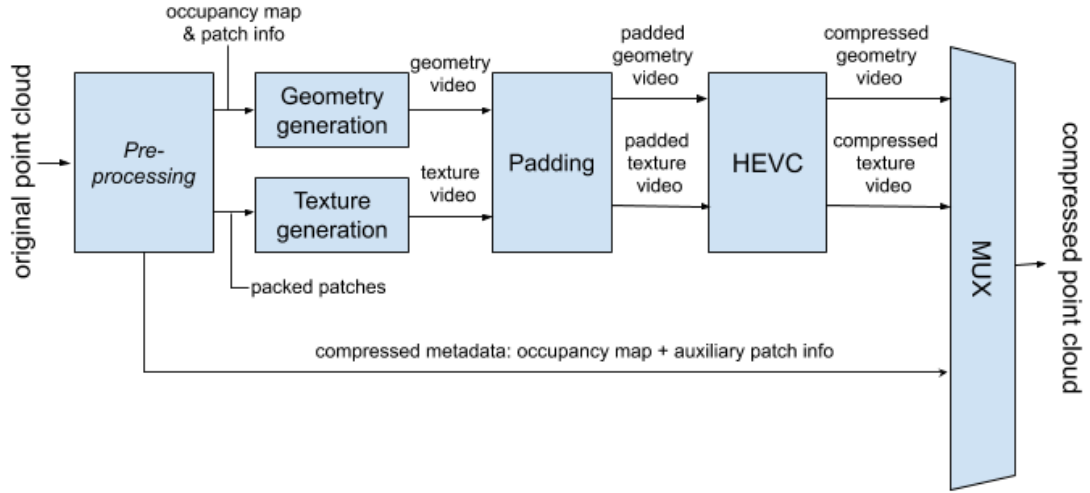


Figure 9.2 – V-PCC compression process. The original point cloud is initially decomposed into geometry video, texture video and metadata. Both video contents are smoothed by *Padding* to allow for the best HEVC (Bross et al., 2012) performance. The compressed bit-streams (metadata, geometry video and texture video) are packed into a single bit-stream: the compressed point cloud.

Geometry-based Point Cloud Compression: G-PCC, also known as TMC13 (Test Model Categories 1 and 3), is a coding technology to compress Category 1 (static) and Category 3 (dynamically acquired) point clouds. Despite the fact that our work is focused on models that belong by default to Category 1, the contents under test are encoded using all the available set-up combinations to investigate the suitability and the performance of the entire space of the available options. Thus, configurations typically recommended for Category 3 contents, are also employed. It is suitable, thus, to present an overview of the entire G-PCC framework.

The basic approach consists in encoding the geometry information at first and, then, using

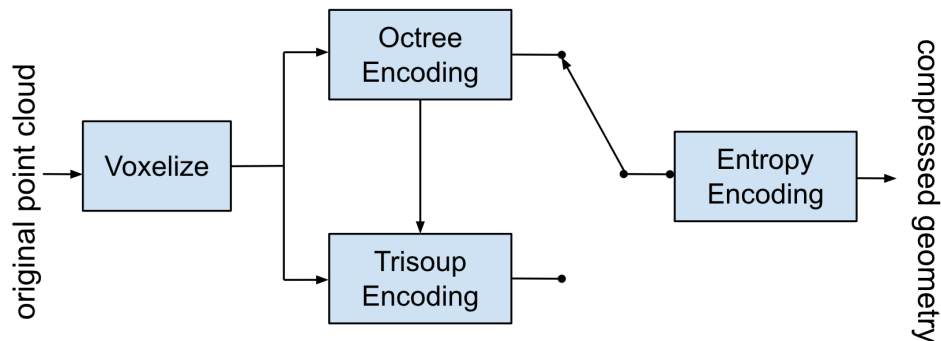


Figure 9.3 – Overview of G-PCC geometry encoder. After voxelization, the geometry is encoded either by Octree or by TriSoup modules, which depends on Octree.

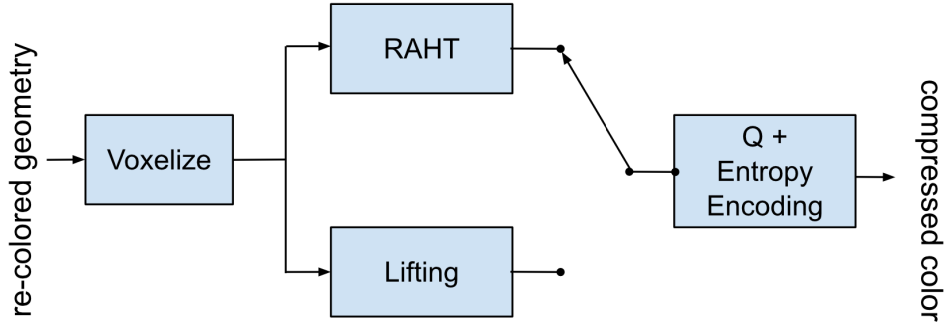


Figure 9.4 – Overview of G-PCC color attribute encoder. In the scope of this work, either RAHT or Lifting are used to encode contents under test.

the decoded geometry to encode the associated attributes. For Category 3 point clouds, the compressed geometry is typically represented as an octree (Meagher, 1982) (Octree Encoding module in Figure 9.3) from the root all the way down to a leaf level of individual voxels. For Category 1 point clouds, the compressed geometry is typically represented by a pruned octree (i.e., an octree from the root down to a leaf level of blocks larger than voxels) plus a model that approximates the surface within each leaf of the pruned octree, provided by the TriSoup Encoding module. The approximation is built using a series of triangles (a triangle soup (Schwarz et al., 2019; Pavez et al., 2018)) and yields good results for a dense surface point cloud.

In order to meet rate or distortion targets, the geometry encoding modules can introduce losses in the geometry information in such a way that the list of 3D reconstructed points, or refined vertices, may differ from the source 3D-point list. Therefore, a re-coloring module is needed to provide attribute information to the refined coordinates after lossy geometry compression. This step is performed by extracting color values from the original (uncompressed) point cloud. In particular, G-PCC uses neighborhood information from the original model to infer the colors for the refined vertices. The output of the re-coloring module is a list of attributes (colors) corresponding to the refined vertices list. Figure 9.4 presents the G-PCC's color encoder, which has as input the re-colored geometry.

There are three attribute coding methods in G-PCC: Region Adaptive Hierarchical Transform (RAHT module in Figure 9.4) coding (de Queiroz and Chou, 2016), interpolation-based hierarchical nearest-neighbor prediction (Predicting Transform), and interpolation-based hierarchical nearest-neighbor prediction with an update/lifting step (Lifting module). RAHT and Lifting are typically used for Category 1 data, while Predicting is typically used for Category 3 data. Since our work is focused on Category 1 contents, every combination of the two geometry encoding modules (Octree and TriSoup) in conjunction with the two attribute coding techniques (RAHT and Lifting) is employed.

9.2 MPEG Common Test Conditions

In this experiment, we evaluate the performance of the emerging MPEG compression approaches, namely, V-PCC, and G-PCC with geometry encoding modules Octree and TriSoup combined with color encoding modules RAHT and Lifting, for a total of five encoding solutions. The codecs are assessed under test conditions and encoding configurations defined by the MPEG committee in the Common Test Conditions document (MPEG 3DG, 2017), in order to ensure fair evaluation and to have a preliminary understanding of the level of perceived distortions with respect to the achieved bit-rate. In this section, the experiment design is described in details; the possibility of pooling results obtained in two different laboratory settings is discussed and analyzed, and the results of the subjective quality evaluation are presented. For the same purpose, the most popular objective quality metrics are employed, and their prediction performance is evaluated.

9.2.1 Data set

For this experiment, every model presented in section 9.1 is encoded using six degradation levels for the four combinations of the G-PCC encoders (from most degraded to least degraded: R1, R2, R3, R4, R5, R6). Moreover, five degradation levels for the V-PCC codec (from most degraded to least degraded: R1, R2, R3, R4, R5) were obtained, following the Common Test Conditions released by the MPEG committee (MPEG 3DG, 2017). Using the V-PCC codec, the degradation levels were achieved by modifying the geometry and texture Quantization Parameter (QP). For both the G-PCC geometry encoders, the `positionQuantizationScale` parameter was configured to specify the maximum voxel depth of a compressed point cloud. To define the size of the block on which the triangular soup approximation is applied, the `log2_trisoup_node_size` was additionally adjusted. From now on, the first and the second parameters will be referred to as *depth* and *level*, respectively, in accordance with (Schwarz et al., 2019). It is worth clarifying that, setting the *level* parameter to 0 reduces the TriSoup module to the Octree. For both the G-PCC color encoders, the color QP was adjusted per degradation level, accordingly. Finally, the parameters `levelOfDetailCount` and `dist2` were set to 12 and 3, respectively, for every content, when using the Lifting module.

9.2.2 Methodology

Test method

In this experiment, the simultaneous DSIS protocol with 5-grading scale was adopted (5: *Imperceptible*, 4: *Perceptible, but not annoying*, 3: *Slightly annoying*, 2: *Annoying*, 1: *Very annoying*). The reference and the distorted stimuli were clearly annotated and visualized side-by-side by the subjects. A division element with radio buttons was placed below the rendering canvases, enlisting the definitions of the selected grading scale among which the subjects had to choose. For the assessment of the visual quality of the models, an interactive

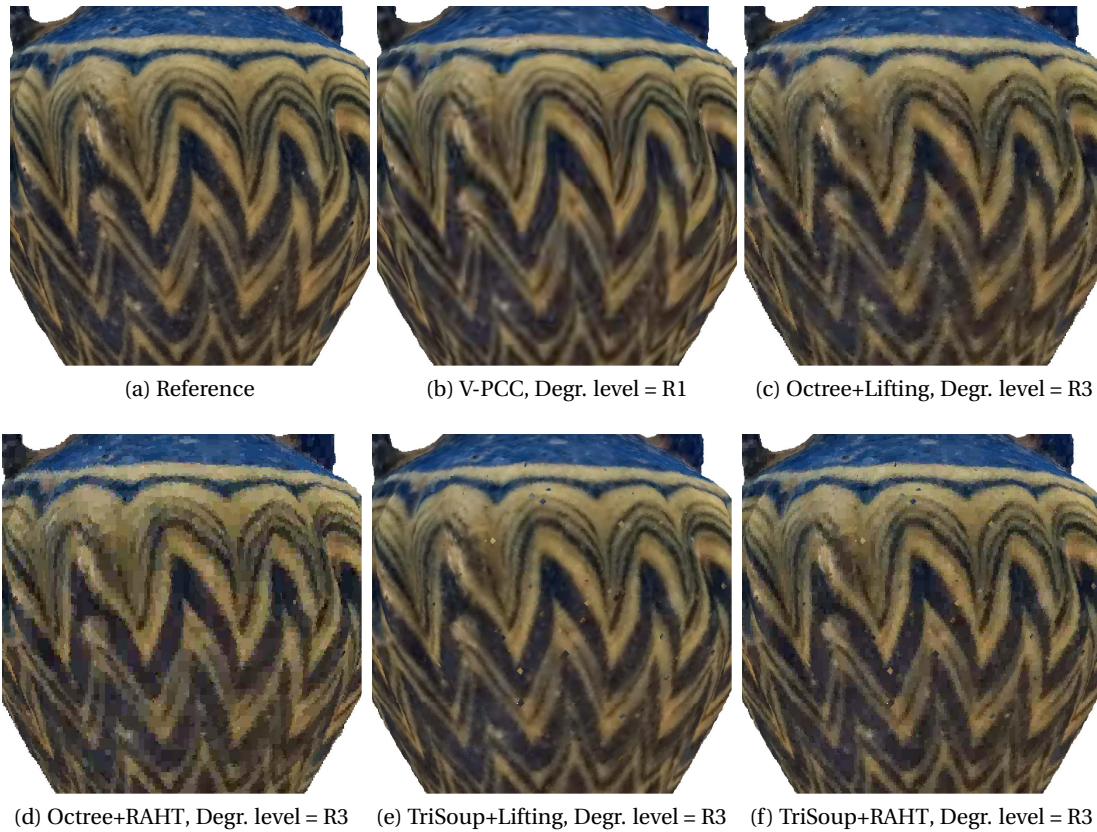


Figure 9.5 – Illustration of artifacts occurred after encoding the content *amphoriskos* with the codecs under evaluation. To obtain comparable visual quality, different degradation levels are selected for V-PCC and G-PCC variants.

evaluation protocol was adopted to simulate realistic consumption, allowing the participants to modify their viewpoint (i.e., rotation, translation and zoom) at their preference without imposing any time limitations. Notice that the interaction commands given by a subject were simultaneously applied on both stimuli (i.e., reference and distorted); thus, the same camera settings were always used in both models. A free viewing protocol was followed, allowing the users to adjust their position with respect to the screen.

Rendering

The stimuli were displayed using the renderer presented and described in annex D.3. A non-distracting mid-grey color was set as the background. The camera zoom parameter was limited in a reasonable range, allowing visualization of a model in a scale from 0.2 up to 5 times the initial size. Note that the initial view allows capturing of the highest dimension of the content in its entirety. This range was specified in order to avoid distortions from corner cases of close and remote viewpoints.

In simple splat-based rendering implementations for point cloud data, there is an obvious trade-off between sharpness and impression of watertight models; that is, as the splat size is increasing, the perception of missing regions in the model becomes less likely, at the expense of blurriness. Given that, in principle, the density of points varies across a model, adjusting the splat size based on local resolutions can improve the visual quality. Thus, in this study, an adaptive point size approach was selected to render the models, similarly to (Javaheri et al., 2017b; Alexiou and Ebrahimi, 2019). In particular, the splat size of a point was set adaptively based on the local mean distance of its 12 nearest neighbors, if it wasn't identified as an outlier; in the latter case, the global mean distance, computed over the same neighborhood population was used instead, to avoid magnification of isolated points. More details about this algorithm can be found in annex D.2. After assigning an initial size from the above procedure, every splat was additionally multiplied by a scaling factor that was determined per content. The scaling factor was selected after expert viewing, ensuring a good compromise between sharpness and perception of watertight surfaces for each reference content. In particular a value of 1.45 was chosen for *amphoriskos*, 1.1 for *biplane*, 1.3 for *romanoillamp* and 1.05 for the rest of the contents. Notice that the same scaling factor is applied for each variation of the content. In Figure 9.5, the reference model *amphoriskos* along with encoded versions at a comparable visual quality are displayed using the developed renderer, to indicatively illustrate the nature of impairments that are introduced by every codec under assessment.

Testing environment

The subjective evaluation experiments were conducted in two laboratories across two different countries, namely, MMSPG at EPFL in Lausanne, Switzerland and LISA at UNB in Brasilia, Brazil. In both cases, a desktop set-up involving an Apple Cinema Display of 27-inches and 2560x1440 resolution (Model A1316) calibrated with the ITU-R Recommendation BT.709-5 (ITU-R BT.709-6, 2015) color profile was installed. At EPFL, the experiments were performed in a room that fulfills the ITU-R Recommendation BT.500-13 (ITU-R BT.500-13, 2012) for subjective evaluation of visual data representations. The room is equipped with neon lamps of 6500 K color temperature, while the color of the walls and the curtains is mid gray. The brightness of the screen was set to 120 cd/m² with a D65 white point profile, while the lighting conditions were adjusted for ambient light of 15 lux, as was measured next to the screen, according to the ITU-R Recommendation BT.2022 (ITU-R BT.2022, 2012). At UNB, the test room was isolated, with no exterior light affecting the assessment. The wall color was white, and the lighting conditions involved a single ceiling luminary with aluminum louvres containing two fluorescent lamps of 4000 K color temperature.

Experimental design

A training session preceded the test, where the subjects got familiarized with the task, the evaluation protocol, and the grading scale by showing references of representative distortions using the *redandblack* content; thus, this model was excluded from the actual test. Identical

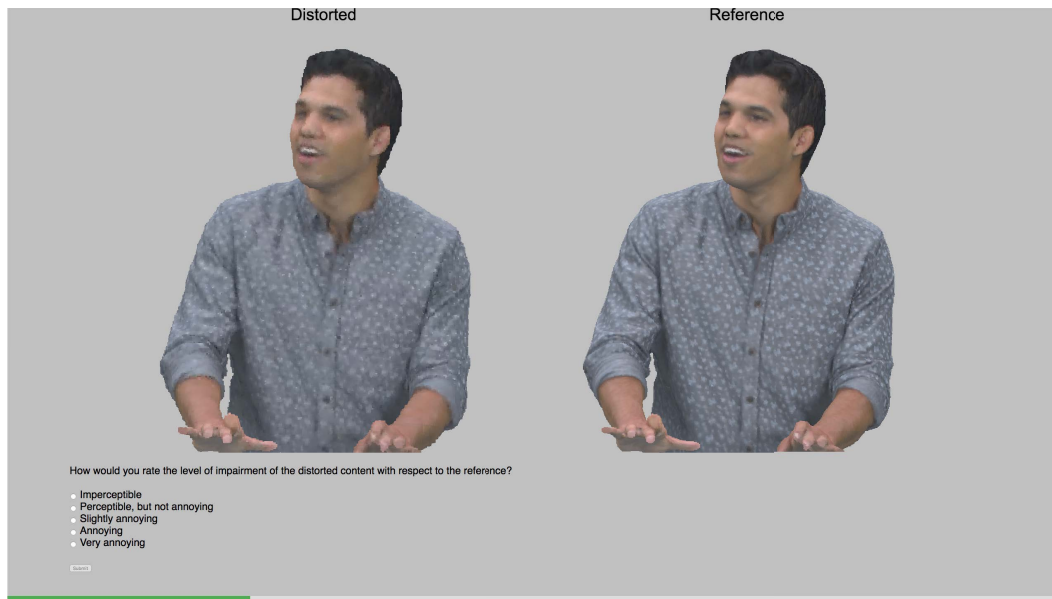


Figure 9.6 – Illustration of the evaluation platform. Both reference and distorted models are presented side-by-side while being clearly remarked. Users’ judgements can be submitted through the rating panel. The green bar at the bottom indicates the progress in the current batch.

instructions were given in both laboratories that participated in the subjective evaluations. At the beginning of each evaluation, a randomly selected view was presented to each subject at a fixed distance, ensuring entire model visualization. To avoid contextual effects, the side of the reference onto the screen was randomly picked for every participant. Moreover, the presentation order of stimuli was randomized per subject and the same content was never displayed consecutively throughout the test, in order to avoid temporal references. In Figure 9.6, an example of the evaluation platform is presented.

In each session, 8 contents and 29 degradations were assessed with a hidden reference and a dummy content for sanity check, leading to 244 stimuli per session. Each session was equally divided in four batches. Each participant was asked to complete two batches of 61 contents, with a 10-minute enforced break in between to avoid fatigue. A total of 40 subjects participated in the experiments at EPFL, involving 16 females and 24 males with an average of 23.4 years old. Another 40 subjects were recruited at UNB, comprising of 14 females and 26 males, with an average of 24.3 years of age. Thus, 20 scores per stimulus were obtained in each laboratory, for a total of 40 scores.

Data processing

Subjective quality evaluation: To evaluate the perceptual quality of the encoded stimuli based on subjective opinions, the MOS and the CIs were computed from the quality scores collected at each participated laboratory separately, as described in annex A.1.1.

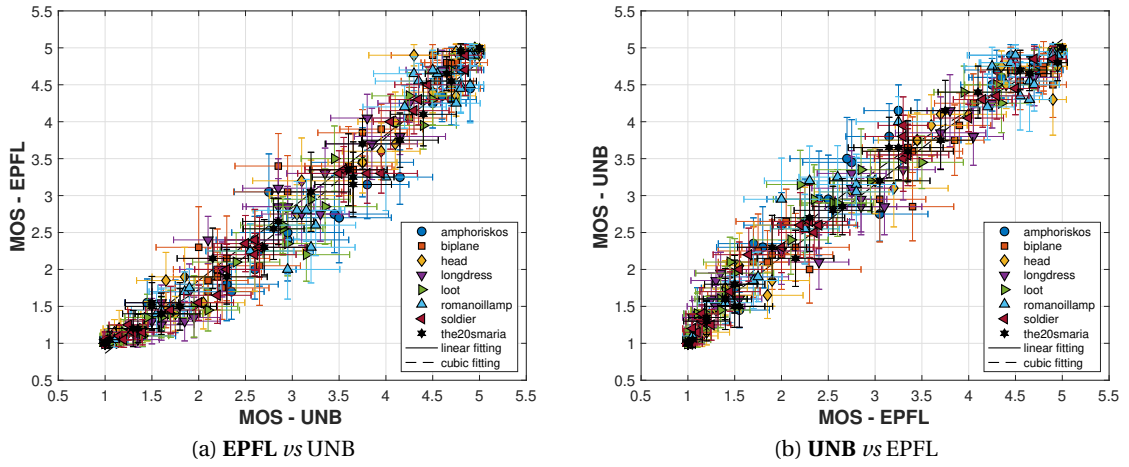


Figure 9.7 – Comparison of subjective scores obtained from the participated laboratories (Bold text represents the ground truth).

Table 9.2 – Performance indexes depicting the correlation between subjective scores from the participating test laboratories (Bold text represents the ground truth).

EPFL <i>vs</i> UNB							
	PLCC	SROCC	RMSE	OR	CE	UE	OE
No fitting	0.984	0.986	0.297	0.254	100%	0%	0%
Linear fitting	0.984	0.986	0.250	0.396	100%	0%	0%
Cubic fitting	0.988	0.986	0.221	0.300	100%	0%	0%
UNB <i>vs</i> EPFL							
	PLCC	SROCC	RMSE	OR	CE	UE	OE
No fitting	0.984	0.986	0.297	0.171	100%	0%	0%
Linear fitting	0.984	0.986	0.250	0.371	100%	0%	0%
Cubic fitting	0.989	0.986	0.211	0.283	100%	0%	0%

Inter-laboratory correlation: Subsequently, in order to determine the statistical equivalence of the results between the two tests, the statistical measurements described in annexes A.2.2 and A.2.3 were employed. In particular, the PLCC, SROCC, RMSE and OR indexes were computed to assess linearity, monotonicity, accuracy and consistency. Moreover, the CE, UE and OR percentages were calculated to decide whether statistically distinguishable scores are obtained for the stimuli under assessment from the two test population. Finally, to better understand whether the results from the two tests conducted in EPFL and UNB could be pooled together, the SOS coefficient was computed for both tests, as described in section A.2.5. Note that close values of a denote similarity among the distribution of the scores, and can be used to determine whether pooling is advisable.

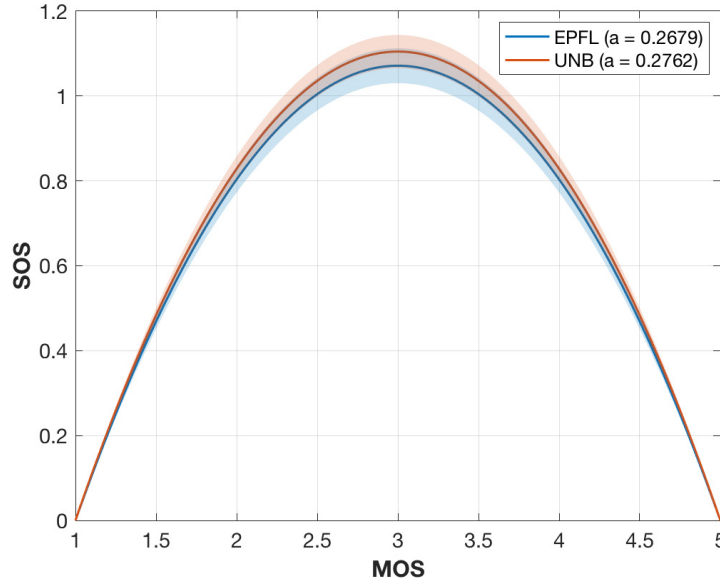


Figure 9.8 – MOS vs SOS fitting for scores obtained in EPFL and UNB, with relative SOS coefficient a . The shaded plot indicates the 95% confidence bounds for both fittings.

9.2.3 Results

Inter-laboratory analysis: In Figure 9.7, scatter plots indicating the relationship between the ratings of each stimulus from both laboratories are presented. The horizontal and vertical bars associated with every point depict the CIs of the scores that were collected in the laboratory indicated by the corresponding label. In Table 9.2, the performance indexes from the correlation analysis that was conducted using the scores from both laboratories as ground truth are reported. As can be observed, the subjective scores are highly-correlated. The CIs obtained from the UNB scores are on average 8.25% smaller with respect to the CIs from EPFL ratings, indicating lower score deviations in the former laboratory. Although the linear fitting function achieves an angle of 44.62° , with an intercept of -0.12 (using EPFL scores as ground truth), it is evident that for mid-range visual quality models, higher scores are observed in UNB. Thus, naturally, the usage of a cubic monotonic fitting function can capture this trend and leads to further improvements, especially when considering the RMSE index. The 100% correct estimation index signifies no statistical differences when comparing pairs of MOS from the two labs individually; however, the high CIs associated with each data point assist on obtaining such a result.

In Figure 9.8 the SOS fitting for scores obtained at EPFL and UNB is illustrated, with respective 95% confidence bounds. As shown in the plot, the values of a are very similar and lie within the confidence bound of the other, with an MSE of 0.0360 and 0.0355, respectively. When combining the results of both tests, we obtain $a = 0.2755$ with an MSE of 0.0317.

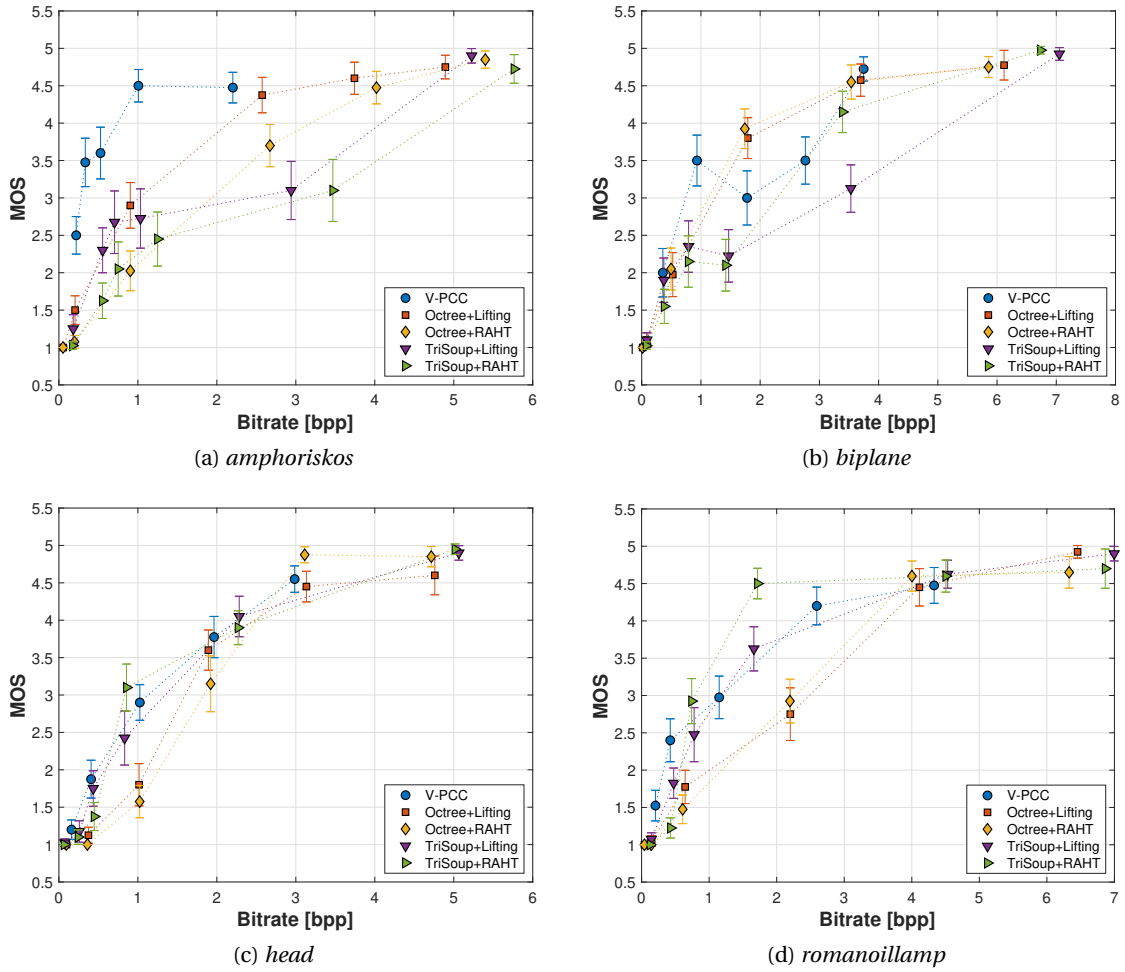


Figure 9.9 – Subjective scores against bit-rates from the degradation levels defined for every codec, grouped per content. Curves for point clouds that represent inanimate objects are illustrated.

The high performance indexes values and the similar α coefficients suggest that the results from the two experiments are statistically equivalent and the scores can be safely pooled together. Thus, for the next steps of our analysis, the two sets are merged and the MOS as well as the CIs are computed on the combined set, assuming that each individual rating is coming from the same population.

Subjective quality evaluation: In Figure 9.9 and 9.10, the MOS along with associated CIs are presented against bit-rates achieved by each codec, per type of content. The bit-rates are computed as the total number of bits of an encoded stimulus divided by the number of input points of its reference version. Our results show that for low bit-rates, V-PCC outperforms the variants of G-PCC, especially in the case of the cleaner set of point clouds that represents

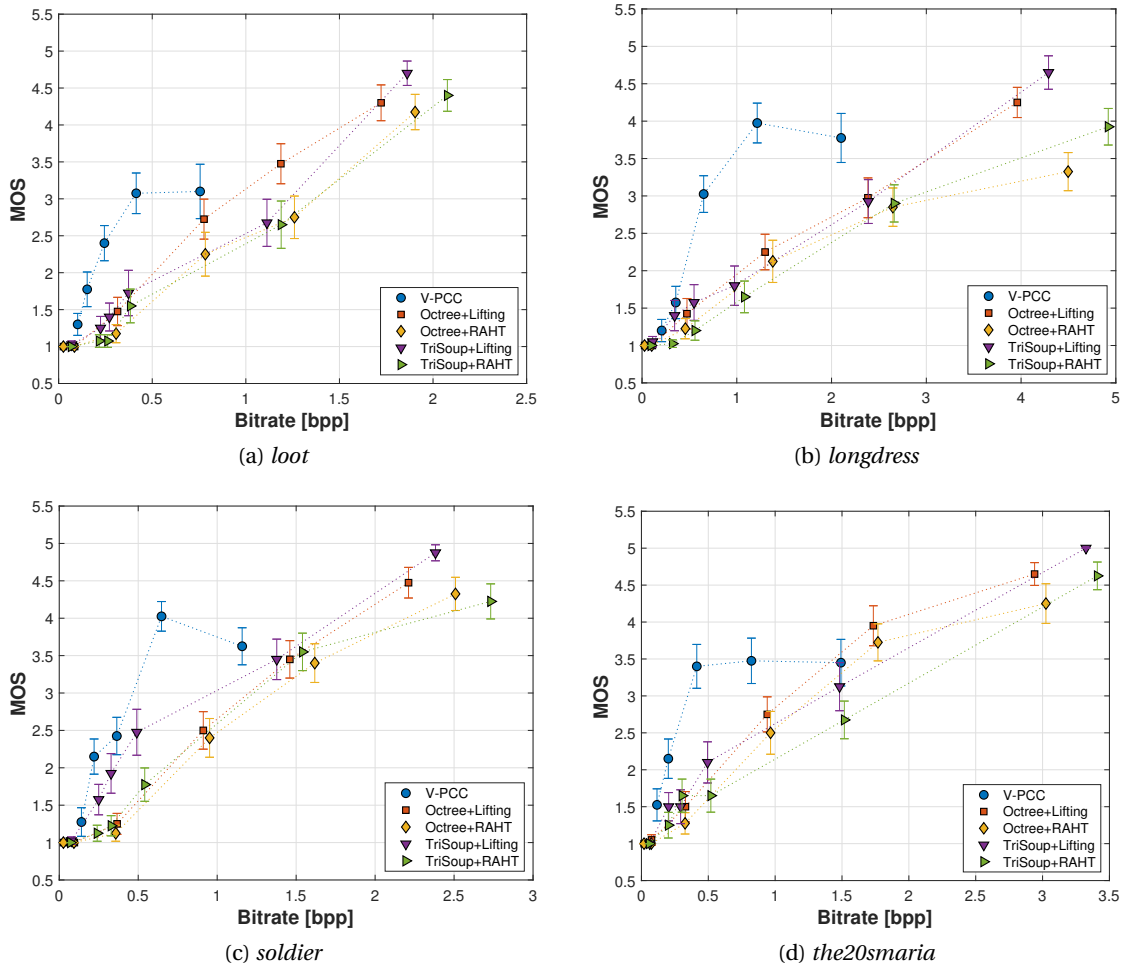


Figure 9.10 – Subjective scores against bit-rates from the degradation levels defined for every codec, grouped per content. Curves for point clouds that represent human figures are illustrated.

human figures. This trend is observed mainly due to the texture smoothing done through low-pass filtering, which leads to less annoying visual distortions with respect to the aggressive blockiness and blurriness that are introduced by the G-PCC color encoders at low bit-rates. Another critical advantage is the ability of V-PCC to maintain, or even increase the number of output points while the quality is decreasing. In the case of more complex and rather noisy contents, such as *biplane* and *head*, no significant gains are observed. This is due to the high bit-rate demands to capture the complex geometry of these models, and the less precise shape approximations by the set of planar patches that are employed.

Although highly efficient at low bit-rates, V-PCC doesn't achieve transparent, or close to transparent quality, at least for the tested degradation levels. In fact, a saturation, or even a drop in the ratings is noted for the human figures when reaching the lowest degradation.

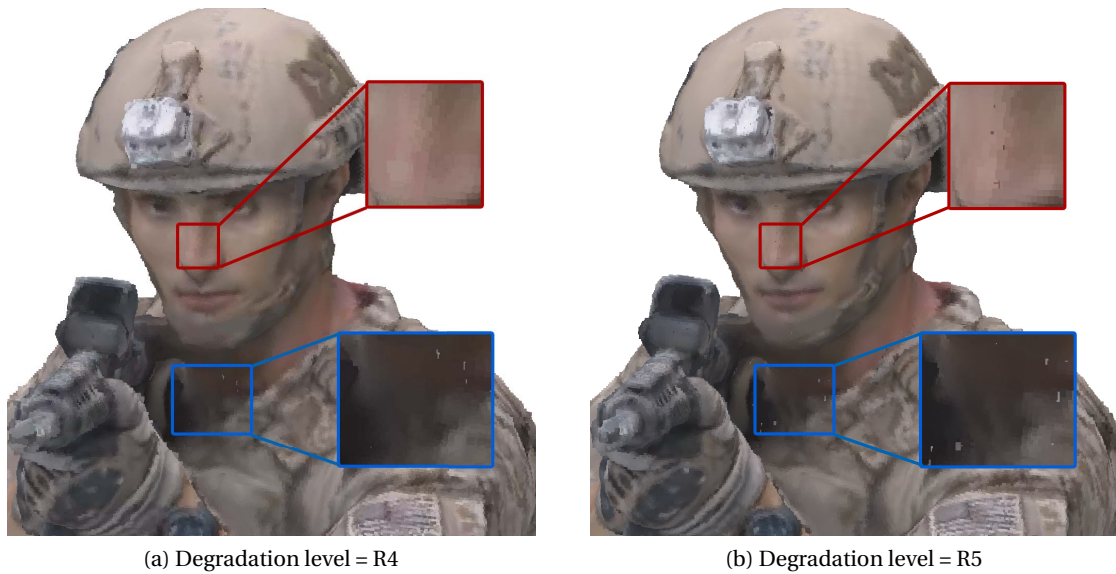


Figure 9.11 – *Soldier* encoded with V-PCC. Although the R4 degraded version is blurrier with respect to R5, missing points in the latter model were rated as more annoying. (examples are highlighted in the figures).

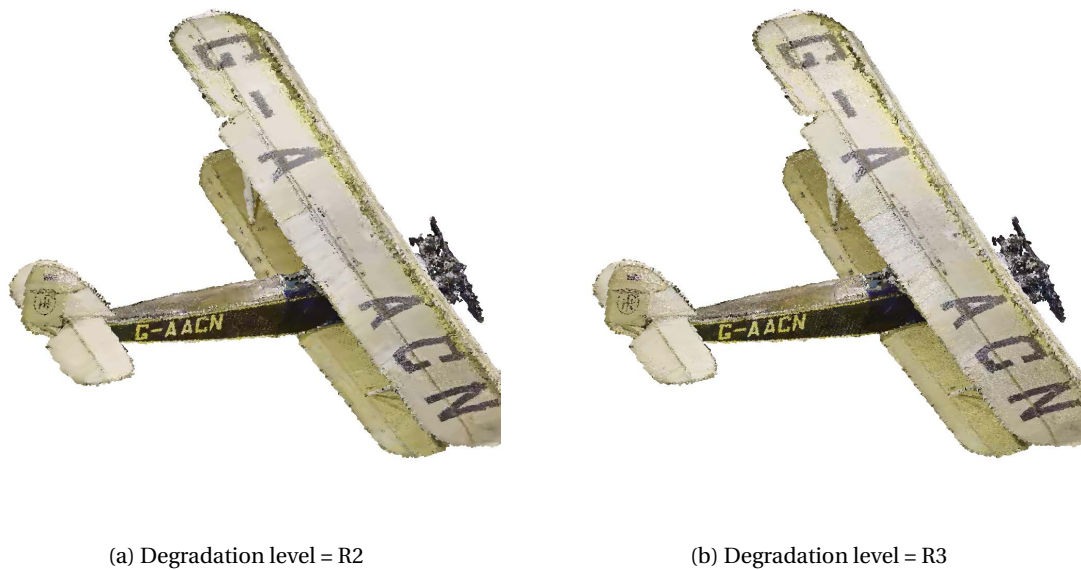


Figure 9.12 – *Biplane* encoded with V-PCC. The color smoothing resulting from the low-pass filtering in texture leads to less annoying artifacts for R2 with respect to R3.

This is explained by the fact that subjects were able to perceive holes across the models, which comes as a result of point reduction. The latter is a side effect of the planar patch

Chapter 9. Benchmarking of MPEG codecs

Table 9.3 – Results of the Welch’s t-test performed on the scores associated with color encoding module Lifting and RAHT, for geometry encoder Octree and TriSoup and for every degradation level. The number indicates the ratio of contents for which the color encoding module of each row is significantly better than the module of each column.

		R1		R2		R3		R4		R5		R6	
		Lifting	RAHT	Lifting	RAHT	Lifting	RAHT	Lifting	RAHT	Lifting	RAHT	Lifting	RAHT
Octree	Lifting	-	0	-	0.25	-	0.5	-	0.375	-	0.125	-	0.375
	RAHT	0	-	0	-	0	-	0	-	0.125	-	0.125	-
TriSoup	Lifting	-	0.25	-	0.875	-	0.625	-	0.25	-	0.125	-	0.5
	RAHT	0	-	0	-	0.125	-	0.25	-	0.125	-	0	-

approximation that does not improve the geometrical accuracy. An exemplar case can be observed in Figure 9.11 for the *soldier* model. Another noteworthy behavior is the drop of the visual quality for *biplane*, between the second and the third degradation level. This is observed because, while the geometric representation of both stimuli is equally coarse, in the first case the more drastic texture smoothing essentially reduces the amount of noise, leading to more visually pleasing results, as shown in Figure 9.12.

Regarding the variants of the G-PCC geometry encoding modules, no decisions can be made on the efficiency of each approach, considering that different bit-rates are in principle achieved. By fixing the bit-rate and assuming that interpolated points provide a good approximation of the perceived quality, it seems that the performance of Octree is equivalent or better than the TriSoup, for the same color encoder. The Octree encoding module leads to sparser content representations with regular displacement, while the number of output points is increasing as the depth of the octree increases. The TriSoup geometry encoder leads to coarser triangular surface approximations, as the *level* is decreasing, without critically affecting the number of points. Missing regions in the form of triangles are typically introduced at higher degradation levels. Based on our results, despite the high number of output points when using the TriSoup module, it seems that the presence of holes is rated, at best, as equally annoying. Thus, this type of degradation doesn’t bring any clear advantages over sparser, but regularly sampled content approximations resulting from the Octree.

Regarding the efficiency of the color encoding approaches supported by G-PCC, the Lifting color encoding module is found to be marginally better than the RAHT module. The latter encoder is based on 3D Haar transform and introduces artifacts in the form of blockiness, due to the quantization of the DC color component of voxels at lower levels that is used to predict the color of voxels at higher levels. The former encoder is based on the prediction of a voxel’s color value based on neighborhood information, resulting in visual impairments in the form of blurriness. Supported by the fact that close bit-rate values were achieved by the two modules, a one-tailed Welch’s t-test is performed at 5% significance value to gauge how many times one color encoding module is found to be statistically better than the other, for Octree and TriSoup geometry encoders separately. Results are summarized in Table 9.3, and

show a slight preference for the Lifting module with respect to the RAHT module. In fact, in the Octree case, the Lifting model is either considered equivalent or better than the RAHT counterpart, the opposite being true only for the lowest degradation values R5 and R6 for 1 out of 8 contents. In the TriSoup case, the number of contents for which the Lifting module is considered better than RAHT either surpasses or matches the number of contents for which the opposite is true. Thus, we can generalize that a slight preference for the Lifting encoding scheme can be observed with respect to the RAHT counterpart. However, note that slightly higher color bit-rates are in principle required by the former approach following the MPEG Common Test Conditions (MPEG 3DG, 2017).

Limitations: The experiment described in this section provides a subjective evaluation of visual quality for point cloud contents under compression artifacts generated by the latest MPEG efforts on the matter. However, this study is not without its limitations.

To ensure a fair comparison, the MPEG Common Test Conditions (MPEG 3DG, 2017) were adopted in selecting the encoding parameters. However, the configurations stated in the document do not cover the range of possible distortions associated with point cloud compression. The fact that V-PCC fails to reach transparent quality is an illustration.

Moreover, for a given target bit-rate, different combinations of geometry and color parameters could be tested, resulting in very different artifacts. The encoding configurations defined in the MPEG Common Test Conditions (MPEG 3DG, 2017) focus on degrading both geometry and color simultaneously. Although the obtained settings are suitable for comparison purposes of updated versions of the encoders, there is no other obvious reason why this should be enforced. Thus, it would be beneficial to test whether a different rate allocation could lead to better visual quality.

Furthermore, the selection of the encoding parameters leads to large variations in file size, and consequently on achieved bit-rates; this makes comparing different encoding solutions particularly challenging, as they are not studied at the same conditions.

Finally, the choice of parameters results in some configurations not being evaluated. For example, the best configurations for TriSoup (R6) corresponds to the Octree encoding module; conversely, the approximation *level* 1 is never tested. Several intermediate solutions, arising from a more varied approach in selecting both *depth* and *level* parameters, are not tested.

9.3 Rate allocation for geometry encoding

Results from previous section showed that, while clear gains in compression efficiency could be seen when adopting V-PCC for point cloud encoding, drawing conclusions about the differences between Octree and TriSoup encoding in G-PCC is more challenging. In order to gain more insights on the impact of geometry encoding, a second experiment was conducted

Table 9.4 – Selected encoding parameters of G-PCC for experiment 2, for high and low target bit-rates. The depth parameter indicates the resolution of the Octree structure, whereas the level parameter indicates the TriSoup approximation.

		<i>amphoriskos</i>			<i>biplane</i>			<i>longdress</i>			<i>loot</i>			<i>the20smaria</i>		
		depth	level	bpp	depth	level	bpp	depth	level	bpp	depth	level	bpp	depth	level	bpp
high bit-rate	G0	512	0	2.01	544	0	1.88	768	0	2.94	576	0	0.82	608	0	1.27
	G1	416	1	2.05	480	1	1.82	608	1	2.90	448	1	0.88	512	1	1.27
	G2	576	2	1.96	608	2	1.84	864	2	2.94	672	2	0.83	736	2	1.26
	G3	736	3	1.97	736	3	1.88	992	3	2.87	928	3	0.85	928	3	1.28
low bit-rate	G0	192	0	0.45	288	0	0.49	384	0	1.00	320	0	0.32	256	0	0.26
	G1	160	1	0.47	256	1	0.51	320	1	0.99	256	1	0.33	224	1	0.28
	G2	224	2	0.48	320	2	0.48	416	2	0.92	384	2	0.32	320	2	0.28
	G3	256	3	0.45	416	3	0.50	480	3	1.00	480	3	0.34	384	3	0.26

to determine whether particular types of geometry artifacts are preferred against others.

9.3.1 Data set

Five contents are selected out from the data set described in section 9.1, to reduce the length and cost of the subjective assessment, while maintaining a wide range of variety. In particular, two models representing objects (*amphoriskos* and *biplane*) and three models representing human figures (*longdress*, *loot* and *the20smaria*) were recruited for the test. For each content, two target bit-rates were selected after expert viewing, to model high and low levels of quality degradations. At every bit-rate point, four geometry configurations of the G-PCC codec were evaluated. In particular, the highest *depth* value d that matched the targeted bit-rate without employing surface approximation was selected (TriSoup *level* value l set to 0). This would represent the pure Octree encoding module, labeled with G0. Subsequently, combinations of d and l were selected such that the final encoded point cloud would meet the bit-rate requirements, for $l = \{1, 2, 3\}$. For $l = 1$, that meant decreasing the value of d with respect to configuration G0, as the TriSoup configuration is expected to generate a higher number of points for *level* 1 with respect to the Octree (configuration G1). This is due to the fact that TriSoup creates a surface approximation of the occupied blocks, constrained to intersect each edge of the block at most once. For $l = 1$ (which signifies a block size of $2 \times 2 \times 2$ voxels), this results in an increase in the amount of points for the decoded point cloud. For $l = 2, 3$, the increase of number of points is mitigated by the progressively larger block sizes; thus, increasing values of d were chosen to match the bit-rate (configurations G2 and G3). This led to 8 configurations per content, for a total of 40 stimuli. For all configurations, the Lifting color module was used, as a slightly better performance was shown in the previous test with respect to RAHT. The color QP was always set to 4 to ensure no color degradations, which could have an effect on the rating.

A summary of the encoding parameters and achieved bit-rate for each content can be found in Table 9.4.

9.3.2 Methodology

Test method

A pairwise comparison methodology with ternary voting system was selected, due its high discriminatory power, in order to collect human preferences regarding the visual quality of two geometry encoded model versions. This protocol is advised from ITU-T Recommendations (ITU-T P.910, 2008), when stimuli are nearly equal in quality. Moreover, this test method is valuable to assess more abstract dimensions, which is in alignment with the scope of the experiment to decide what types of visual impairments are more annoying. In order to avoid forced choices in case of imperceptible differences among the two stimuli, a ternary voting system was adopted. The subjects were able to interact with the stimuli under evaluation through mouse movements, similarly to the test described in section 9.2, while a free viewing protocol was also adopted.

Rendering

The models were displayed using the exact same settings that were described in the sub-section rendering of section 9.2.2.

Testing environment

The test was performed in UNB, using the same conditions and the same room, as described in the sub-section testing environment of section 9.2.2.

Experimental design

Each subject was presented a pair of point cloud stimuli, displayed in a side-by-side manner, and was asked to declare which of the two models they preferred, with the option of no preference. The comparisons were only performed between the same content and within the same target bit-rate, for a total of 60 pairs to be assessed. Particular care was given to avoid displaying the same content consecutively, while the order of the stimuli was randomized per subject.

One training example was shown to the subjects to help them familiarize with the testbed and the task at hand; two identical stimuli of high quality that weren't part of the test were used for the purpose. Additionally, one dummy content was added at the beginning of the test to ease participants into the task, and the associated scores were discarded.

A total number of 25 subjects participated, involving 13 males and 12 females, with an average of 25 years of age.

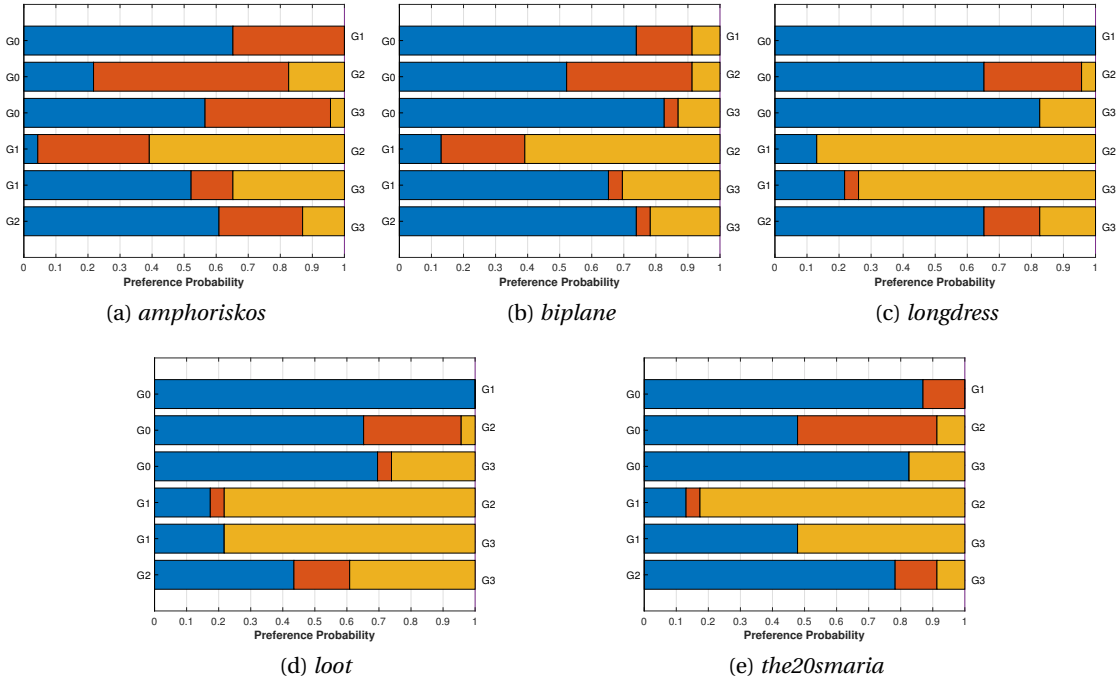


Figure 9.13 – Preference and tie probabilities for each pair of configurations under test in experiment 2, for the high bit-rate case. The color blue (yellow) of the bar indicates the probability of the configuration on the left (right) side being preferred over the one on the right (left) side. The orange bar indicates the tie probability.

Data processing

Outlier detection was performed on the data according to (Lee et al., 2013). One outlier was found, and the scores associated with it were subsequently discarded.

For each pair under assessment, the winning frequency w_{ij} of stimulus i against stimulus j was computed, along with the ties t_{ij} . In order to obtain the preference probabilities, the winning and tie frequencies were divided by the total number of subjects after outlier detection. The normalized MOS scores on a 0-100 scale were obtained from the winning frequencies by applying the Bradley-Terry-Luce (BTL) model, according to the Recommendation ITU-T J.149 (ITU-T J.149, 2004) and as described in annex A.1.2.

9.3.3 Results

Figures 9.13 and 9.14 depict the preference and tie probabilities for each pair of configurations i and j under test, for each content, for high and low target bit-rates, respectively.

Results show that the pure Octree configuration G0 is clearly the preferred approach. This conclusion can be drawn from the preference probabilities, which indicate that it is likely

9.3. Rate allocation for geometry encoding

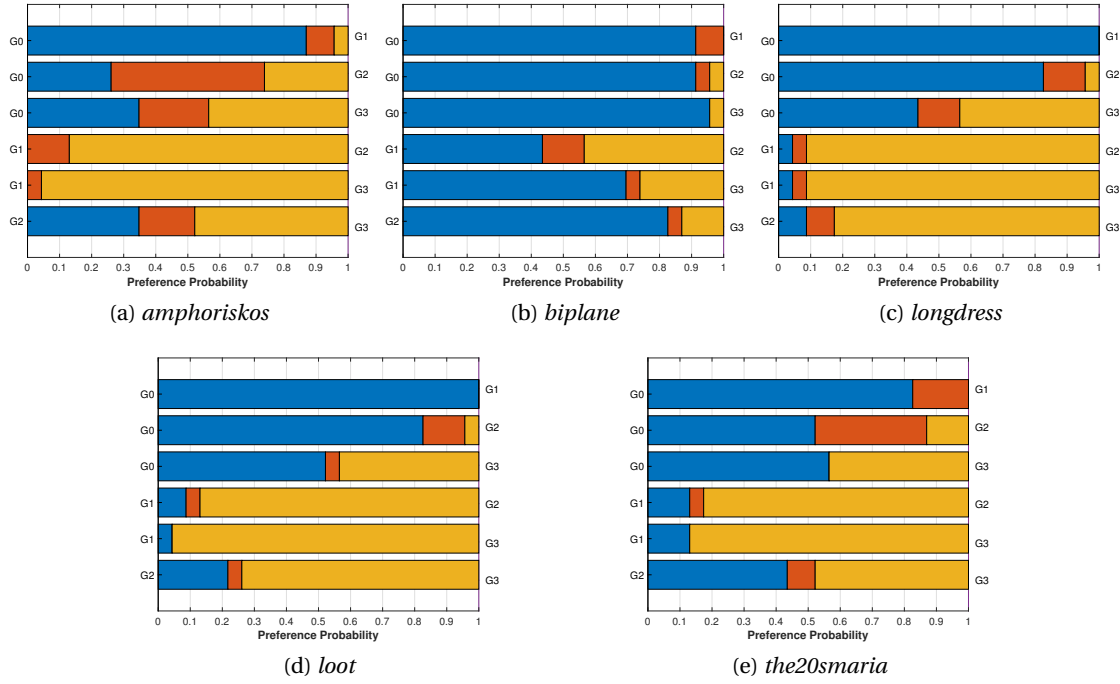


Figure 9.14 – Preference and tie probabilities for each pair of configurations under test in experiment 2, for the low bit-rate case. The color blue (yellow) of the bar indicates the probability of the configuration on the left (right) side being preferred over the one on the right (left) side. The orange bar indicates the tie probability.

for the Octree to be rated as either equal or better in perceived quality with respect to all the other configurations, for both low and high bit-rates. The sole exception in this trend is the comparison of G0 with G3 for *amphoriskos*, at low bit-rate, for which the latter configuration is preferred more frequently. In fact, when discarding the ties in the computation of the winning frequencies, configuration G0 is considered as worse than one of the other configurations in only 8.70% and 15.65% for high and low bit-rates, respectively. Ties account for 16% of the total number of ratings assigned to G0.

Conversely, configuration G1 seems to yield the worst performance, as it is considered better than other configurations in only 14% of the cases, and is rated as worse in 77.10% of the cases (71.30% and 88.70% for high and low bit-rates, respectively). In comparison, configuration G2 is rated as worse in 36.09% of the cases, and configuration G3 in 48.99% of the cases, while ties account for 18.26% and 8.12%, respectively.

Considering the high bit-rate case, besides configuration G0, which is likely to be either preferred or considered equal to all other configurations, G2 is the second-best configuration, as it is rated as better than configurations G1 and G3 in the majority of the cases, and is considered nearly equal to configuration G3 for content *loot*. It is worth mentioning that it is also considered nearly equivalent in quality with configuration G0 for content *amphoriskos*.

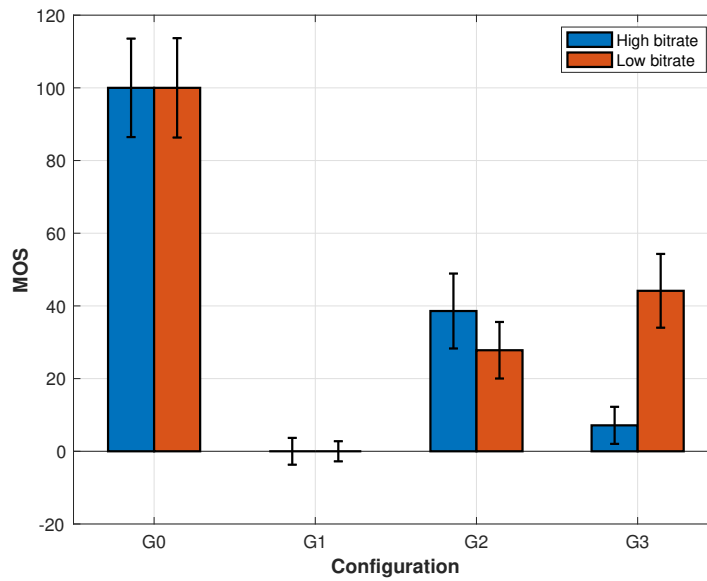


Figure 9.15 – Normalized MOS and relative CIs obtained from the winning frequencies gathered in experiment 2, for each configuration, averaged across the contents, separately for high and low bit-rates.

Finally, configuration G3 is rated as yielding better results with respect to configuration G1 for point clouds representing human figures, whereas for object models, it is rated as worse than G1. However, it is universally considered worse than configuration G0 and G2, with the aforementioned exception of content *loot*.

For the low bit-rate case, G0 is confirmed as the approach yielding the best results, as it is always outperforming G1 and is rated to be either better or equivalent to configurations G2 and G3. Configuration G1 is outperformed by all the other configurations, with the notable exception of content *biplane*, for which it is considered equivalent with respect to configuration G2, and for which it outperforms configuration G3. The outlier behavior of this content can be explained by the presence of noise in its reference version, which has an impact on the perception of geometry degradation. Between configurations G2 and G3, the latter is preferred, as it achieves either equivalent or better quality with respect to the former. Thus, it appears that a better depth resolution was preferred for low bit-rates, even if it came at the cost of a coarser surface approximation, at least when comparing TriSoup levels 2 and 3.

Figure 9.15 depicts the normalized MOS obtained from the winning frequencies, along with the respective CIs, as averaged across the contents. The blue bar represents the scores associated to the high bit-rate, whereas the orange bar represents the scores associated to the low bit-rate. Results clearly show the superiority of the Octree configuration with respect to the TriSoup ones; moreover, they confirm that configuration G1 seems to yield the worst performance in terms of visual quality. For high bit-rate, configuration G2 is preferred with respect to configuration G3, whereas for low bit-rate, the opposite was found to be true.

9.4. Rate allocation for geometry and color encoding

Table 9.5 – Selected encoding parameters of G-PCC for experiment 3, for high and low target bit-rates. The depth parameter indicates the resolution of the Octree structure, whereas the QP parameter indicates the quantization parameter for the Lifting encoding module.

		<i>amphoriskos</i>			<i>biplane</i>			<i>longdress</i>			<i>loot</i>			<i>the20smaria</i>		
		depth	QP	bpp	depth	QP	bpp	depth	QP	bpp	depth	QP	bpp	depth	QP	bpp
high bit-rate	B0	768	34	2.57	768	34	1.79	896	28	2.38	896	28	1.19	896	28	1.73
	B1	672	26	2.62	672	30	1.81	736	24	2.40	736	22	1.18	832	26	1.71
	B2	800	38	2.59	928	40	1.80	992	30	2.36	992	32	1.20	960	30	1.74
low bit-rate	B0	512	40	0.90	512	40	0.52	768	34	1.30	768	34	0.78	768	34	0.94
	B1	480	36	0.91	480	38	0.52	576	28	1.30	704	30	0.77	640	28	0.94
	B2	544	44	1.03	576	44	0.51	896	38	1.31	800	38	0.76	864	40	0.95

Results of the subjective experiment show that the surface approximation generated by the TriSoup module is rarely considered as superior than the regular Octree structure. This is especially true when the surface approximation is done at level $l = 1$, which, for the same bit-rate, demands a lower depth precision with respect to the Octree module. Increasing the depth precision by applying a coarser surface approximation ($l = 2, 3$) yields better results within the TriSoup module; however, the quality is still considered worse than what obtained at a lower depth precision by the Octree module.

9.4 Rate allocation for geometry and color encoding

One of the main limitations of the experiment conducted using the Common Test Conditions, can be pinpointed to its inability to analyze geometry and color degradations separately, or to identify the impact of different levels of impairment on the visual quality due to the simultaneous quality reduction in both texture and geometry. However, since several configurations of the geometry and texture encoding modules could lead to the same target bit-rate, it is not a given that choosing a medium level of degradation for both modules will lend the best possible results in terms of perceived quality. For instance, discarding some geometry information to be able to increase the quality of the texture encoding, or the opposite, could lead to more visually pleasing outcomes. Thus, it is critical to assess whether between the geometry and texture encoders and within a target bit-rate, which bit allocation is most efficient and visually pleasant. Thus, in this experiment, we test which combination of color and geometry encoding parameters would lead to the best results in terms of visual quality.

9.4.1 Data set

The same contents that were selected for the second experiment were also used in this test. For each content, two target bit-rates were chosen based on the results of the first experiment (section 9.2), to model medium-high and medium-low levels of quality degradation in terms

of both geometry and color. The Octree geometry in combination with the Lifting color encoding modules were adopted as the individually preferred alternatives from the previous experimentation. For contents *amphoriskos* and *biplane*, bit-rate R3 was selected for the low target bit-rate and R4 was selected for the high target bit-rate, whereas for contents *longdress*, *loot* and *the20smaria* bit-rates R4 and R5 were selected as low and high target bit-rate, respectively. Those encoded contents would form configuration B0. For every rate, the geometry and color quantization parameters were modified such that the same target bit-rate would be achieved. This is performed by either decreasing the parameter *depth*, which would allow allocation of more bits to the texture encoder (configuration B1), or by increasing the *depth* in the geometry encoder, which would lead in quality reduction of the texture encoder to match the target bit-rate (configuration B2). This way, the configuration of preference can be obtained in a rate allocation problem.

A summary of the encoding parameters and achieved bit-rates per content is reported in Table 9.5. We remind the readers that higher levels of QP correspond to a coarser color encoding, whereas lower levels of *depth* represent a decrease in geometry precision.

9.4.2 Methodology

Test method

The same test method described in the corresponding sub-section of 9.3.2 was employed in this experiment; that is, a pair comparison with the option of tie, realized in an interactive platform under a free viewing protocol.

Rendering

Identical configurations were employed, as detailed in the sub-section rendering of section 9.2.2.

Testing environment

This test was performed at EPFL under the same conditions and the same room described in the sub-section testing environment of section 9.2.2.

Experimental design

One training example was shown to the subjects to help them familiarize with the testbed and the task at hand; two identical contents with high quality that were excluded from the test were used for the purpose. Additionally, one dummy content was added at the beginning of the test to ease participants into the task, and the associated scores were discarded. The same guidelines were followed for the order and presentation of the stimuli under assessment,

9.4. Rate allocation for geometry and color encoding

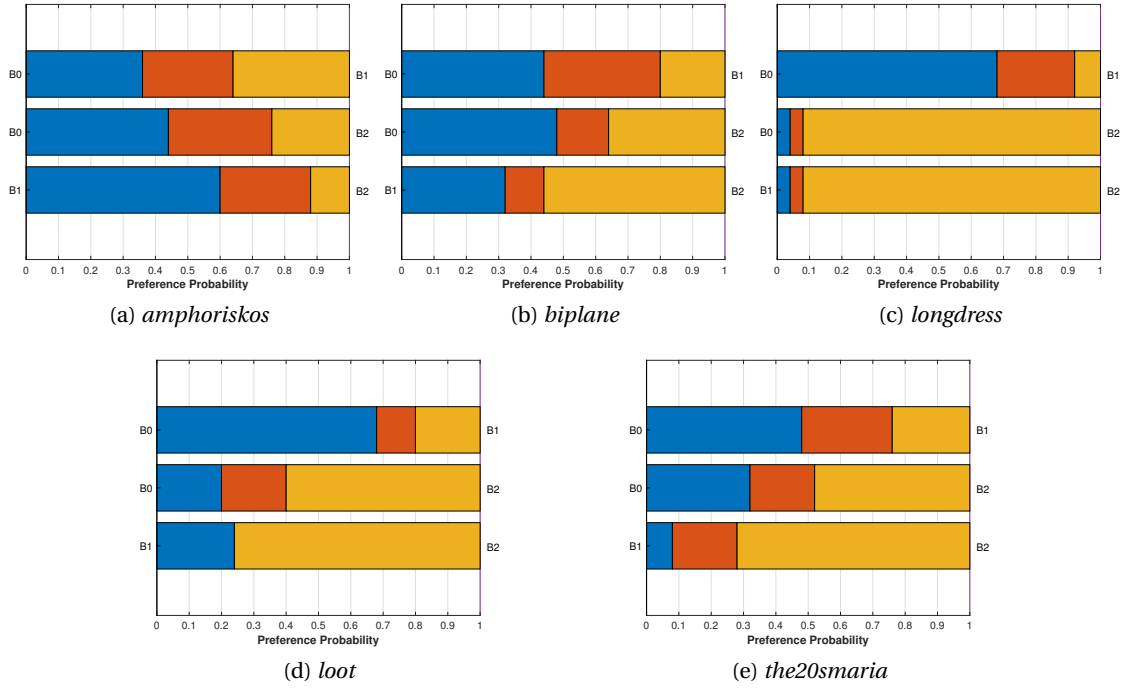


Figure 9.16 – Preference and tie probabilities for each pair of configurations under test in experiment 3, for the high bit-rate case. The color blue (yellow) of the bar indicates the probability of the configuration on the left (right) side being preferred over the one on the right (left) side. The orange bar indicates the tie probability.

as described in the sub-section experimental design of section 9.3.2. A total number of 25 subjects participated, involving 17 males and 8 females, with an average of 29.13 years of age.

Data processing

Outlier detection was performed on the data according to (Lee et al., 2013). No outlier was detected among the subjects. The same data processing as described in section 9.3.2.

9.4.3 Results

Figures 9.16 and 9.17 present the preference and tie probabilities for each pair of configurations i and j under test, for each content, for high and low target bit-rates, respectively. Results show that, depending on the content and its target bit-rate, different rate allocation for geometry and color can be preferred. For the high bit-rate case, configuration B2 seems to yield better results than its counterparts for contents *longdress*, *loot* and *the20smaria* (albeit marginally, for the latter, when compared to B0); for contents *biplane*, it outperforms configuration B1, but not B0, while for content *amphoriskos*, it is outperformed by both configurations. For all

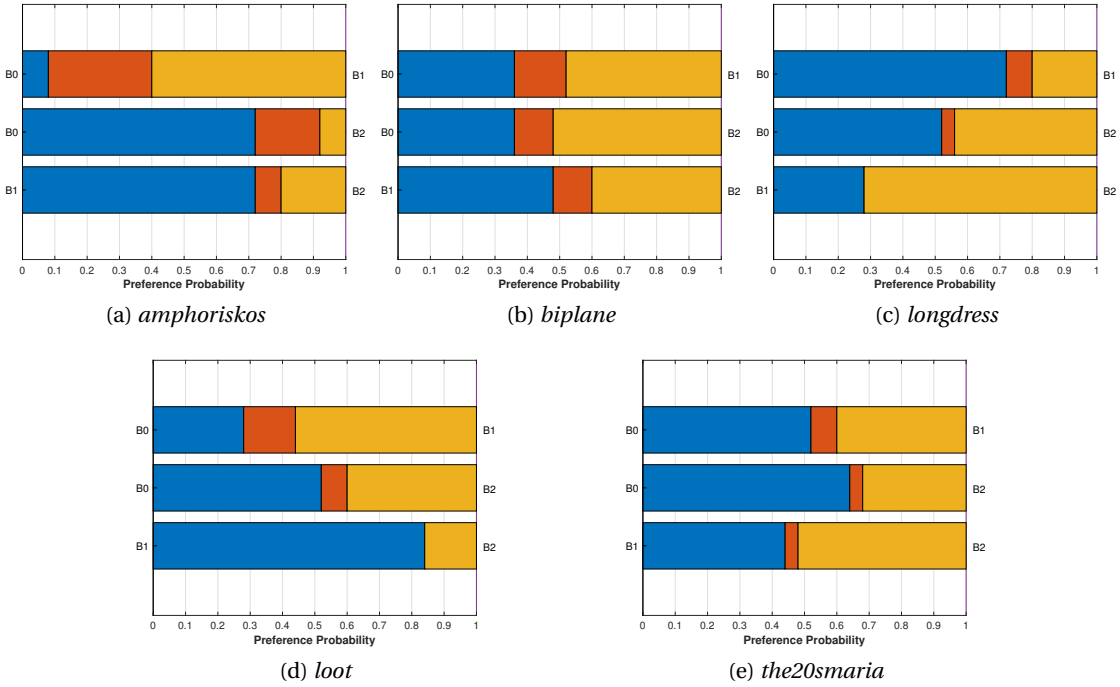


Figure 9.17 – Preference and tie probabilities for each pair of configurations under test in experiment 3, for the low bit-rate case. The color blue (yellow) of the bar indicates the probability of the configuration on the left (right) side being preferred over the one on the right (left) side. The orange bar indicates the tie probability.

contents, B0 seems to be slightly preferred or considered equal to configuration B1, indicating a general trend that favors better geometry accuracy than color fidelity.

For low bit-rates, results are more varied. For contents *amphoriskos*, *biplane* and *loot*, B1 seems to be the winning configuration, as it is rated to be either better or equal than the other two configurations. This indicates that color fidelity is preferred over geometry resolution. For contents *amphoriskos* and *loot*, configuration B0 is the second-best rated, confirming this trend; however, for content *biplane* B2 seems to be preferred with respect to B0. In the case of content *longdress* and *the20smaria*, however, B1 seems to be the least preferred solution, as both configurations B0 and B2 have a higher probability of being preferred with respect to B1. For both contents, B0 is considered as yielding a better visual quality with respect to B2, although marginally so for content *longdress*.

Figure 9.18 depicts the normalized MOS obtained from the winning frequencies using the BLT model. It can be observed that the relative CIs are quite large, probably due to the differences in performance between different contents. The general trend indicates that for high bit-rates, B2 is the best configuration, followed by B0, which points towards a preference for more level of details in geometry with respect to color. However, for low bit-rates B2 is the worst

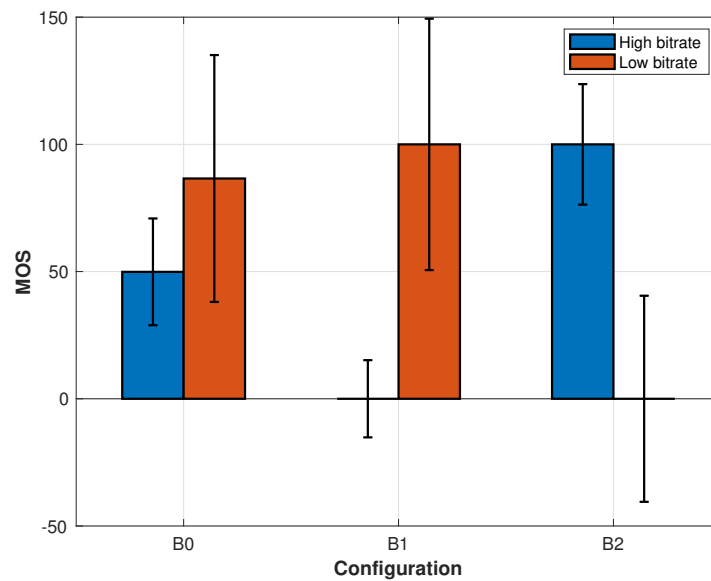


Figure 9.18 – Normalized MOS and relative CIs obtained from the winning frequencies gathered in experiment 3, for each configuration, averaged across the contents, separately for high and low bit-rates.

configuration, and B1 seems to be highly preferred. This suggests that for low bit-rates, better color fidelity might be more important than geometrical accuracy.

Results of the subjective experiment show that, depending on the targeted bit-rate, different configurations of geometry and color could be preferred. In particular, for high bit-rates, better geometry precision is preferred, whereas for low bit-rates, color fidelity seems to be the most important parameter. Yet, any decision for a rate allocation problem should be done on a content basis, as results vary significantly among them.

9.5 Conclusions

In this study, a comprehensive quality assessment and analysis of the emerging MPEG point cloud compression has been carried, through subjective evaluations. Our efforts were initially focused on quality assessment experiments using the Common Test Conditions, as defined by the MPEG committee, and experts' selection of encoding configurations. For this purpose, a diverse set of point cloud models was selected, prepared, encoded, and evaluated subjectively, using a point-based rendering software that was developed. The first experiment provided useful insights regarding the performance of the encoders and the types of degradation they introduce, yet, limitations were identified and described. Among them is the inability to draw solid conclusions about the efficiency of the G-PCC encoder. This could shed some light on the preference among the different visual artifacts introduced by the Octree and the TriSoup modules. Thus, a second experiment was conducted showing that human subjects prefer regular down-sampling over triangulated surface approximations, at both low and high bit-rates. We

have also addressed the restriction of the initial set of encoding configurations of downgrading both the geometry and color quantization parameters simultaneously, to investigate whether a better rate allocation scheme is possible. The results of the third experiment on this matter showed that, roughly, higher color quality is preferred at low bit-rates, while higher geometry precision is favoured at high bit-rates, even though results may vary among different contents.

The subjective scores obtained from the first experiment reported in this chapter have been made publicly available. Additional information is provided in annex E.

10 Learning-based encoding

Point cloud imaging has emerged as an efficient and popular solution to represent immersive visual information. However, the large volume of data generated in the acquisition process reveals the need of efficient compression solutions in order to store and transmit such contents. Several standardization committees are in the process of finalizing efficient compression schemes to cope with the large volume of information that point clouds require. At the same time, recent efforts on learning-based compression approaches have been shown to exhibit good performance in the coding of conventional image and video contents. It is currently an open question how learning-based coding performs when applied to point cloud data.

In static point cloud compression, there are different approaches aiming at reducing the data size of geometric or textural information. Notably, the most popular solutions employ tree data structures, graphs, or patches of projected views of a model. The first rely on data structures that can efficiently organize the spatial placement of the points, such as k -d trees and octrees; the second employ graph arrangements to represent a model with nodes indicating a point, or a neighborhood. The latter approaches are based on plane projections of a model that are typically obtained from different perspectives and can be encoded using conventional 2D imaging compression solutions. Lately, auto-encoding neural network architectures have been proposed to compress point cloud geometry, extending similar efforts that have preceded in 2D imaging. Despite the fact that this type of point cloud coding is still at its infancy, the results are very promising, with the current solutions competing, if not outperforming, state-of-the-art algorithms.

Inspired by the great potentials that neural networks show in learning transforms for compressing visual data representations, in this study we extend previous efforts by learning geometry and color attributes of point cloud models. In particular, we initiate by extending a publicly available geometry-only point cloud auto-encoding solution in learning transforms for a holistic data representation including both geometry and color. We analyse the performance of this unified network, using widely employed objective quality metrics that focus on geometric and color degradations. Moreover, we examine the impact of assigning various weights to geometry and color distortion terms in the loss function, to understand whether an optimal

weighting scheme can be found. The performance of this model is compared to a different architecture that is composed of two separately trained networks dedicated to geometry and color. Furthermore, the proposed model is benchmarked against a widely-used coding solution, which denotes the anchor in the recent point cloud compression-related efforts of the MPEG standardization body. A set of meta-analysis studies is also reported, carried to understanding the impact of data set, color space, and loss function selection, among others, in the network performance. Results demonstrate that the adopted architecture is able to perform competitively with respect to well-established solutions for point cloud compression, both in the geometry and color domain, especially at low bitrates.

To the best of our knowledge, there is only one study focused on compression of point cloud attributes, described in (Quach et al., 2020a), which is based on folding a 2D grid onto a point cloud and then mapping the attributes on top of it. An advantage of this approach is the application of highly efficient 2D imaging techniques for point cloud compression; yet, a bottleneck is the low accuracy of the folding in geometrically complex parts of a model. In our study, we handle geometry and/or color in the 3D domain by extracting features from regular grids making use of 3D convolutions, which enable capturing of spatial redundancies for both types of information. The study aims to provide useful insights for future references focused on the matter.

This chapter is based on material that has been published in (Alexiou et al., 2020a).

10.1 Network architecture

The network encodes a point cloud in a block-by-block basis, similarly to previous efforts on the field (Guarda et al., 2019b,a, 2020; Wang et al., 2019). Thus, every point cloud is initially partitioned into non-overlapping blocks of a specified dimension. Each block is sequentially fed into the network and encoded independently through an auto-encoding architecture. After decoding, a compressed variation of the initial block at the original dimension is exported. Remark that partitioning a point cloud into blocks has two main advantages; that is, lower computational demands in handling input units, and random access, provided that every block is interpreted as an independent sample. Yet, it comes with the limitation that spatial redundancies cannot be largely exploited when blocks of low resolution are selected.

10.1.1 Input

The geometry and texture of every input unit is provided in a typical format, which resembles a 6-tuple list, with each entry denoting a point that is defined by its x , y and z coordinates followed by the r , g and b color values. The input point cloud data are considered voxelized, thus, the original format can be easily converted to a 3D voxel grid. This data representation allows us to exploit 3D convolution kernels to capture spatial redundancies in the output feature maps. The 3D voxel grid is then partitioned into blocks of a specified resolution, and

each block is associated with a number of input channels that carry topological and potentially textural information, depending on the task. In particular, the blocks are of resolution $K \times K \times K \times \hat{C}$, with $\hat{C} = 1$ for geometry-only compression and $\hat{C} = 4$ for color-only or geometry-plus-color encoding. In all cases, the first channel contains values of 0 or 1 to indicate occupied voxels. The optionally-enabled, additional color channels contain values between 0 and 1, obtained after a scaling step.

10.1.2 Auto-encoder

The network architecture adopts as a baseline the model proposed in (Quach et al., 2019). As the majority of the current auto-encoding solutions, the processing pipeline can be decomposed in three main parts; that is, an analysis stage consisting of convolution layers, a synthesis stage that is composed of de-convolution layers, and a bottleneck in the middle that corresponds to the latent representation. Our selection for this baseline is motivated by the fact that it denotes a publicly available, efficient implementation of an end-to-end auto-encoder with good performance on geometry compression. Moreover, similar core architectures have been employed in 2D image-based paradigms, revealing high-performance in terms of compression efficiency.

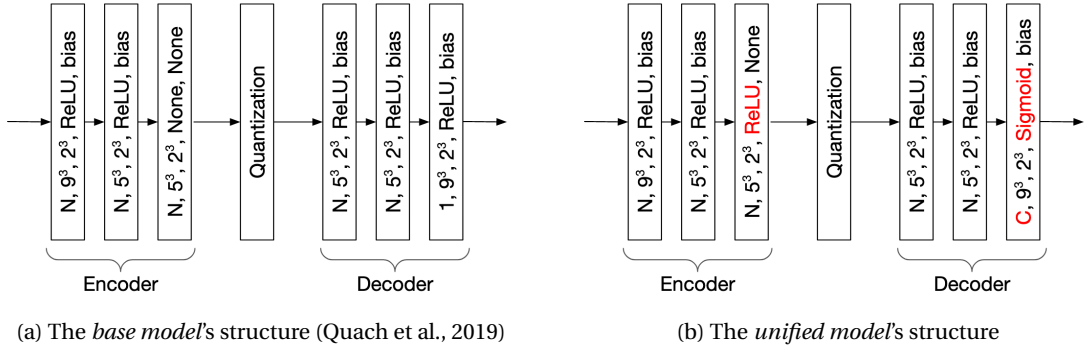


Figure 10.1 – Auto-encoding architecture.

Base model

In this model, only point cloud geometry is encoded (Quach et al., 2019). The architecture is composed of three 3D convolution layers at the encoder and their symmetric transposed convolution (i.e., de-convolution) counterparts at the decoder side, as illustrated in Figure 10.1a. The first term of each block of a diagram denotes the number of filters (i.e., N), the second term denotes the size of the filters (i.e., 9^3), the third term is the size of the stride (i.e., 2^3), the fourth term is the type of activation function, and the fifth term indicates if bias is applied.

At the encoding stage, a 3D point cloud block is given as an input. A selection of stride size higher than 1 implies down-sampling of the input representation. In our case, a stride

size of 2^3 , denotes down-sampling of the input unit by a factor of 0.5 in each dimension at the output of each layer. Quantization is applied on the latent representation, which is obtained at the output of the encoder. During training, the quantization is replaced by additive uniform noise (Ballé et al., 2016), in order to ensure that the gradient is defined for the back-propagation operation. Moreover, the rate is estimated using differential entropies (Ballé et al., 2016), provided that the values at the output of the quantization step are continuous. During testing, the floating point latent representation is quantized with trained probability tables, and the bitstream is obtained by entropy coding.

At the decoding stage, the bitstream is received and passes from a set of de-convolution layers with stride size equal to 2^3 , which implies up-sampling by a factor of 2. Through a series of symmetric de-convolution layers, the compact feature maps are decoded and the point cloud geometry can be recovered in the form of 3D blocks. A loss function is employed to quantify the reconstruction distortion and train the model in an end-to-end manner performing joint optimization of both rate and distortion. For this purpose, a multiplier is employed to steer the trade-off at will. In particular, the loss is composed of this multiplier (weight term) λ_g , a distortion term D_g , and a rate term R that represents bits per input occupied voxel (bpp) as follows:

$$L = R + \lambda_g D_g \quad (10.1)$$

Note that by modifying the λ_g term, the bitrates and the reconstructed quality can be tuned; that is, by setting a higher weight, the model will focus more on learning how to preserve geometry information and less on compressing, thus, resulting in higher reconstruction quality at the expense of higher bitrate. The distortion term is computed by comparing the original X with the recovered point cloud \tilde{X} . This task can be interpreted as a binary classification problem, hence, the focal loss is employed to assess the reconstruction error, defined as in (Lin et al., 2017) and given in Equation 10.2

$$\begin{aligned} \text{FL}(\dot{p}_x) &= -\alpha_x(1 - \dot{p}_x)^\gamma \log(\dot{p}_x), \\ \text{FL}(\tilde{X}) &= \sum_{x \in X} \text{FL}(\dot{p}_x), \end{aligned} \quad (10.2)$$

where \dot{p}_x is defined as p_x if the voxel x is occupied and $1 - p_x$ if the voxel is unoccupied, p_x is the output value of the voxel x indicating probability of whether the voxel is occupied or not. α_x is defined as α if the voxel x is occupied and $1 - \alpha$ otherwise.

Unified model

In this model, point cloud geometry and/or color attributes can be encoded. The same architecture as in the base model is employed, with some necessary modifications to support the enhanced functionality, illustrated in Figure 10.1b. In this diagram, red color is used to highlight differences with respect to the original version. Specifically, the number of channels for the last layer of the decoder, C , is set to either 1, 3 or 4 depending on the task. For geometry

compression, C is equal to 1, for color compression C equals 3, while for geometry-plus-color compression, C is equal to 4. Notice that a ReLU activation function is added at the final layer of the encoder, while at the final layer of the decoder, the activation function is switched to sigmoid in order to ensure that the output values lie in the range $[0, 1]$.

To train the network for point cloud geometry compression, we employ a slight variation of the loss function defined in (Quach et al., 2019) and provided in Equation 10.1. In particular, the distortion term is normalized by dividing with the total number of voxels, such that it represents a measurement of distortion per voxel.

To train the network for point cloud color compression, a similar formulation is adopted. In this case, the focal loss is replaced by a simple l_2 norm, which is computed between the original and the reconstructed color values across the occupied voxels of the input block. The color loss is normalized by dividing with the number of occupied voxels of a block to reflect the distortion per occupied voxel.

Note that both geometry and color distortion terms are normalized by the number of points that effectively contribute to the loss. For color degradation, a logarithmic function of the l_2 norm is computed to obtain scores in the same range with the geometry term. In Equation 10.3, the updated loss function used for color-only compression is provided.

$$L = R + \lambda_c D_c \quad (10.3)$$

To train the network for point cloud geometry-plus-color compression, both metrics are employed and both distortion terms are included in the loss function, as indicated in Equation 10.4. Notice that the overall quality of the restored model as well as a different quality preservation scheme can be enabled for the two attribute types by selecting different λ values. Note that subscripts g and c indicate geometry and color, respectively.

$$L = R + (\lambda_g D_g + \lambda_c D_c) \quad (10.4)$$

10.1.3 Output

For each input block, a bit-stream representing the encoded latent representation is received at the decoder side. After de-compression, an equally sized degraded version of the block is obtained. When geometry-only compression is required, the model outputs 1 channel that indicates occupancy. In color-only compression, 3 color channels are obtained. Notice that, in this case, the receiver knows the point cloud topology; thus, the compressed attributes per point are found at the corresponding voxel position at the output blocks. For geometry-plus-color compression, 4 channels are obtained combining occupancy and color information. In all cases, the output blocks that are extracted from the same point cloud are merged together following a particular order, to restore the de-compressed point cloud. Finally, the optionally compressed color values are converted back to the original range $[0, 255]$.

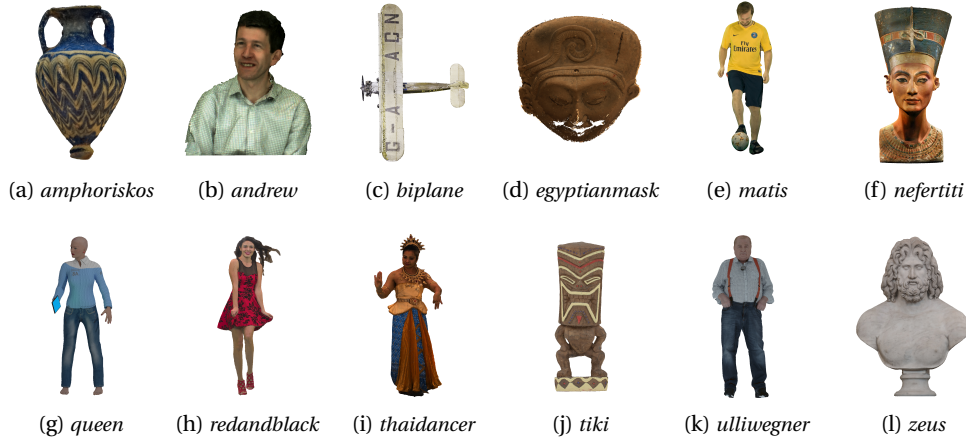


Figure 10.2 – Sample models used for training.

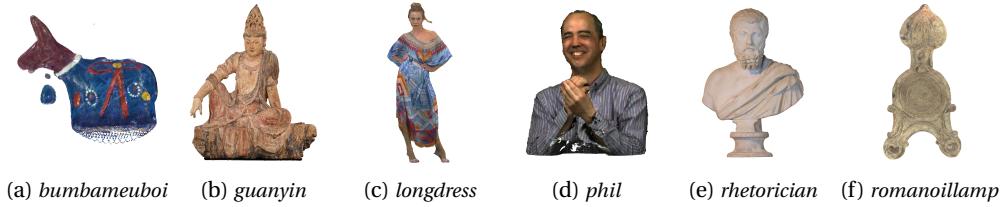


Figure 10.3 – Models used for testing.

10.2 Experimental setup

10.2.1 Data set

For the purposes of this study, a selection of high resolution point clouds from several repositories was pursued in order to form a collection of training and testing models with diverse characteristics in geometry and color. In particular, a total of 50 models were selected from the MPEG¹, JPEG Pleno², PointXR (Alexiou et al., 2020b), VSENSE (Zerman et al., 2019), and M-PCCD (Alexiou et al., 2019a) data sets, forming the so-called High Resolution Geometry and Color (HiResGC) data set. The JPEG Pleno and MPEG repositories consist of colored models that were assembled in the context of relevant standardization activities, containing representative sets of real-life acquired and synthetic point clouds that span across a variety of categories, such as, inanimate models, cultural heritage, human bodies, etc. The *PointXR dataset* (Alexiou et al., 2020b) consists of low-noise, high quality point clouds that represent cultural heritage models, obtained after conversion from their original mesh content representations. The VSENSE data set (Zerman et al., 2019) consists of two dynamic sequences of human bodies, thus, including several frames of the same figures at different poses. From

¹<http://mpegfs.int-evry.fr/MPEG/PCC/DataSets/pointCloud/CfP/>, last accessed 01/2020

²<https://jpeg.org/plenodb/>, last accessed 12/2020

this repository, only a representative, low-noise frame was selected per sequence. Finally, a content coming with the M-PCCD data set (Alexiou et al., 2019a) was recruited, to further enhance the data.

The majority of the models were voxelized at a bit depth of 10, independently of their original content representation (i.e., raw or voxelized at a higher grid resolution). Sparser point clouds were voxelized at a bit depth of 9, as in (Guarda et al., 2019b), while models with geometry originally lying at a grid of lower resolution (e.g., 9), remained as such (e.g., Microsoft Upper Bodies). Moreover, the color attributes were normalized in a range between 0 and 1. The collected point clouds were partitioned into blocks, with the latter denoting the input data that are fed into the network.

Training data

The training data consists of the entire set of point clouds that were collected, excluding 6 models that comprise our testing set. A part of the selected models is illustrated in Figure 10.2. The training models were partitioned into non-overlapping blocks of size K , with $K = 32$ or 64 depending on the task at hand, with each block being handled independently in our network. Following (Guarda et al., 2019b), blocks that contain less than 500 occupied points were discarded, as they carry limited relevant information. From the remaining blocks, a total of 10,000 samples were randomly picked to form our training set.

Testing data

The testing data consists of the models that have been specified in the Common Test Conditions document authored by the JPEG standardization committee as a result of its latest efforts (Perry, 2020). The employed models denote a representative set of inanimate objects and human figures with a relatively wide range of geometric arrangement and color distribution, as illustrated in Figure 10.3. For the testing data, block sizes of $K = 128$ are used, except if otherwise mentioned. Note that it is a rather common approach (Quach et al., 2019) to use different resolutions for training and testing blocks, whose influence is investigated in section 10.4.2.

10.2.2 Evaluation methodology

In this study, we opt two objective quality metrics that are largely employed in the literature in order to allow cross-comparisons, and we evaluate the quality of geometry and color information for the compressed models, separately. For evaluation of geometric distortions, we choose the symmetric point-to-plane metric with MSE using the PSNR variant, noted hereafter as D2-PSNR. The D2-PSNR captures topological distortions in a point cloud model by measuring the deviation of the coordinates of a distorted point from a linear approximation of the reference surface. To compute the PSNR variant, the resolution of the voxel grid that

the content lies in is employed at the numerator of the ratio. Two error values are obtained by setting both the compressed and the original model as a reference, and the symmetric error is obtained by choosing the maximum out of the two error values. For evaluation of color distortions, the symmetric color PSNR is adopted. The well-known formula from 2D imaging is employed, using the nearest neighbors algorithm to establish associations between the reference and the content under evaluation. To compute a quality score, the color values of the point cloud models are converted from the original RGB to the YCbCr colorspace using the ITU-R Recommendation BT.709-6 (ITU-R BT.709-6, 2015). This metric from now is referred to as YUV-PSNR. To compute a total score, a weighted average between the luma and the two chrominance channels is obtained using weights 6, 1 and 1, as in (Ohm et al., 2012). This procedure is repeated setting both the original and the distorted models as the reference and the maximum error is kept to account for the symmetric YUV-PSNR score.

To compute both metrics the MPEG software version 0.13.5 is used (Tian et al., 2017c). For the D2-PSNR, normal vectors are required to be associated with the coordinates of the testing models. In this case, we used a plane fitting algorithm with 10 nearest neighbors as implemented in MeshLab v2020.06.

10.2.3 Network configurations

To train the network, we select a number of filters $N = 32$ per layer, a batch size of 16, and a number of output channels $C = 4$, to involve both geometry and color information. The Adam optimizer (Kingma and Ba, 2014) is set with learning rate equal to 10^{-4} and $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The loss function given in Equation 10.4 is employed, using $\alpha = 0.9$ and $\gamma = 2.0$ for the focal loss computation given by Equation 10.2. The experiments are conducted using Python 3.6 and Tensorflow 1.13.1. As mentioned earlier, training blocks of size $K = 32$ and testing blocks of size $K = 128$ are in principle employed, except if otherwise declared.

10.3 Experimental results

When compressing both the geometric structure and the color attributes of point cloud contents using neural networks, two main approaches can be identified. The first approach relies on creating a holistic representation of both dimensions, feeding both geometry and color information to a network designed to compress both simultaneously. The second approach relies on designing two separate networks to be used sequentially: one that handles geometry, and another that deals with compressing the color attributes. The first approach is advantageous in terms of computational and time resources. Moreover, it allows for an holistic evaluation of point cloud distortions, given a loss function that can reliably detect artifacts in both geometry and color domains at the same time. In the second approach, networks dedicated on a particular type of information are employed and, thus, a better performance is expected provided the usage of the same network hyper-parameters (i.e., number and size of filters, size of strides, etc). Furthermore, the rate allocation for each component can be

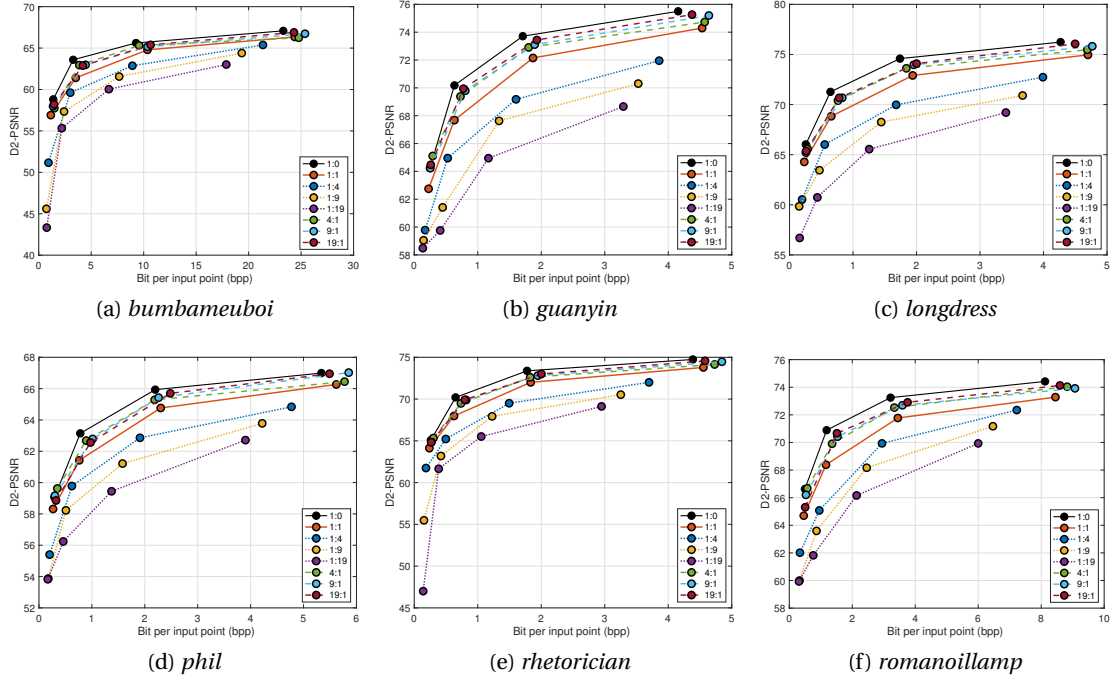


Figure 10.4 – Rate-distortion performance of the unified network architecture, according to geometry metric D2-PSNR, with different λ allocations to geometry and color ($\lambda_g : \lambda_c$). Solid black represents pure geometry compression ($\lambda_c = 0$), solid red represents 1:1 allocation. Dashed lines represent allocations for which $\lambda_g > \lambda_c$, whereas for dotted lines, $\lambda_g < \lambda_c$.

manipulated independently, thus leading to higher flexibility in the encoding process.

In this section, we describe and provide performance evaluation results for a series of experiments conducted using the unified model as a baseline, which compresses geometry and color attributes simultaneously. In particular, we analyse how the performance of the network is affected when different weights are given to either geometry or color distortions. Then, we compare the performance of our unified model with respect to using separate networks to encode geometry and color information. Finally, benchmarking results against the MPEG anchor are depicted to indicate the performance of the network against a well-established encoding solution.

10.3.1 Geometry against color impairments using the unified network

Figures 10.4 and 10.5 depict the performance evaluation of using the unified model to compress both geometry and color, according to geometry metric D2-PSNR and color metric YUV-PSNR, respectively, for all testing contents. To obtain the curves, parameters λ_g and λ_c in the loss function are weighted in order to obtain different allocation schemes, indicated by $\lambda_g : \lambda_c$. For these experiments, the unified model described in section 10.1.2 and illustrated in Figure 10.1b is employed.

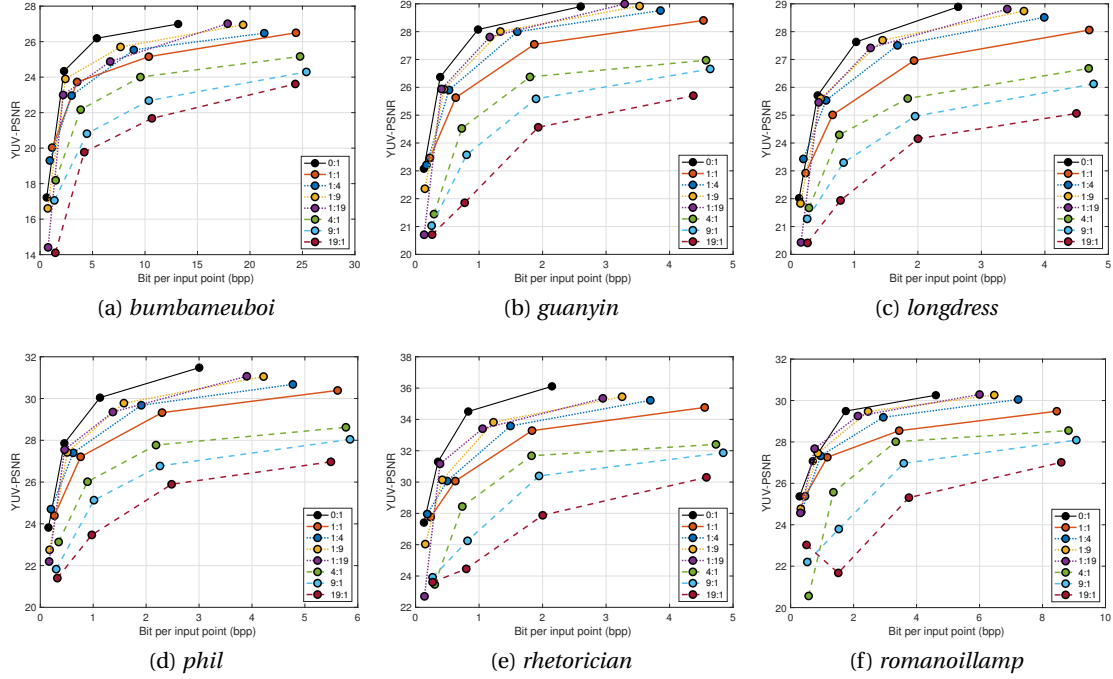


Figure 10.5 – Rate-distortion performance of the unified network architecture, according to geometry metric YUV-PSNR, with different λ allocations to geometry and color ($\lambda_g : \lambda_c$). Solid black represents pure color compression ($\lambda_g = 0$), solid red represents 1:1 allocation. Dashed lines represent allocations for which $\lambda_g > \lambda_c$, whereas for dotted lines, $\lambda_g < \lambda_c$.

In Table 10.1, the values of λ_g and λ_c that were selected to achieve the desired weighting for geometry and color distortions, respectively, are reported. Figure 10.4 indicates how different weighting schemes for geometry and color distortions affect the quality of the reconstructed point cloud in the geometry domain, expressed through the D2-PSNR metric. In particular, the solid black line shows the performance when the color distortion is not considered in the computation of the loss function ($\lambda_c = 0$). As such, it represents an upper limit on the performance in terms of geometrical distortions. The solid red line indicates the performance when equal weights are assigned to both color and geometry distortions, which we consider as the baseline. As expected, an increase in performance can be observed when more relative

Table 10.1 – Selected values of λ_g and λ_c for the computation of the loss function as in Equation 10.4, to achieve various distortion allocation schemes with ratios $\lambda_g : \lambda_c$, for different bitrate values (from smallest to largest, R1 to R4).

	1:1		0:1		1:4		1:9		1:19		1:0		4:1		9:1		19:1	
	λ_g	λ_c	λ_g	λ_c	λ_g	λ_c	λ_g	λ_c	λ_g	λ_c	λ_g	λ_c	λ_g	λ_c	λ_g	λ_c	λ_g	λ_c
R1	20	20	0	40	8	32	4	36	2	38	40	0	32	8	36	4	38	2
R2	100	100	0	200	40	160	20	180	10	190	200	0	160	40	180	20	190	10
R3	500	500	0	1000	200	800	100	900	50	950	1000	0	800	200	900	100	950	50
R4	2500	2500	0	5000	1000	4000	500	4500	150	4750	5000	0	4000	1000	4500	500	4750	150

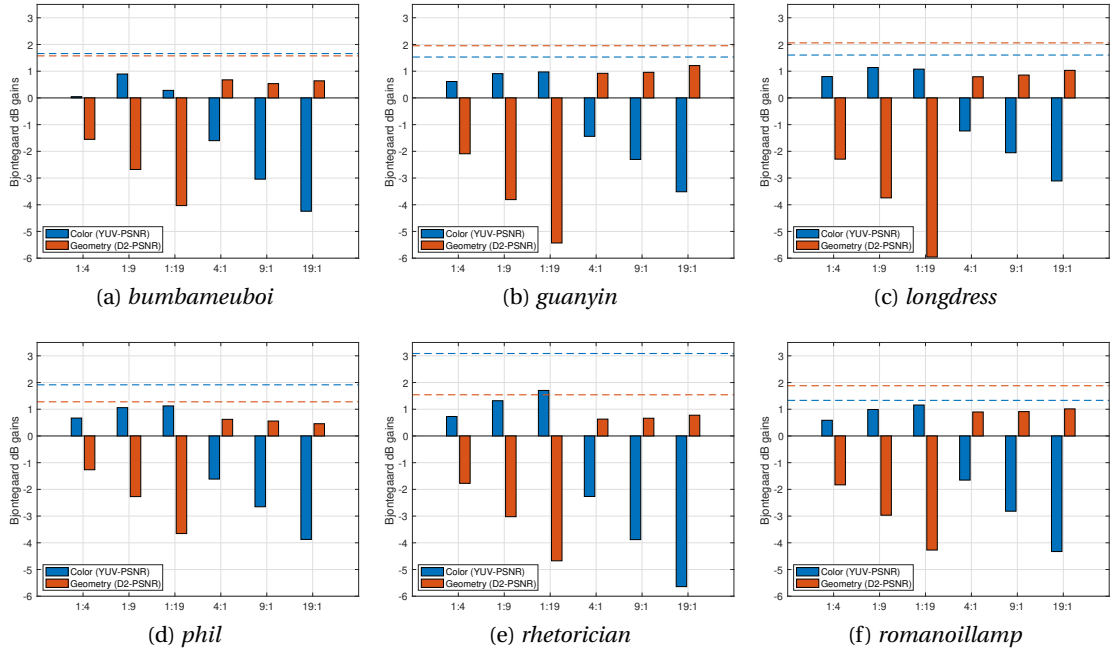


Figure 10.6 – Bjontegaard dB gains for each allocation $\lambda_g : \lambda_c$ with respect to allocation 1:1, for color metric YUV-PSNR (blue) and geometry metric D2-PSNR (red). Dashed lines represent dB gains when using pure color compression (blue) or pure geometry compression (red), with respect to 1:1 baseline.

weight is assigned to the geometry distortion in the loss function (dashed lines). However, the increase in performance is not as remarkable as the dB losses that are observed when more relative weight is assigned to the color distortion term (dotted lines). In fact, the performance for weight ratios 4:1, 9:1, and 19:1 is approximately equivalent for all contents.

A similar trend can be observed in Figure 10.5, which presents the performance of the same weighting schemes in terms of color distortion, represented by the YUV-PSNR metric. As in Figure 10.4, the solid black line indicates the performance when the color distortion is only considered in the loss function ($\lambda_g = 0$). It is noteworthy that, certain allocation schemes mark an increase in performance with respect to the theoretical upper limit 0:1 at low bitrates. This is due to the fact that the computation of the color metric depends on the underlying geometry. Thus, in a geometry-plus-color compression scheme, the reconstructed error is measured on a different than the input topology, which might lead to such behaviours, especially in such low color quality levels. As expected, allocation schemes which favor color distortions (dotted lines) achieve better performance with respect to the 1:1 baseline (depicted in solid red). However, sharp loss in performance can be observed when more weight is assigned to geometry distortions, at the expense of color information (dashed lines).

In order to better analyse the impact of varying the relative importance of color or geometry information in the loss function calculation, we computed the Bjontegaard dB gains obtained

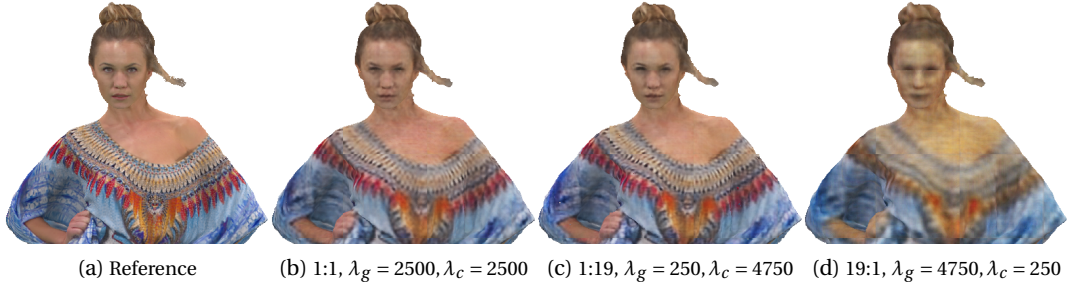


Figure 10.7 – Visual comparison for *longdress*, for different distortion allocation ratios.



Figure 10.8 – Visual comparison for *guanyin*, for different distortion allocation ratios.

by each allocation scheme under exam, with respect to the 1:1 baseline. Results are depicted in Figure 10.6, separately for each test content. Blue color indicates dB gains computed with respect to the color metric YUV-PSNR, whereas red color indicates gains with respect to geometry metric D2-PSNR. Dashed lines represent the theoretical upper limit, i.e., the gains obtained when using only geometry (1:0, red dashed) or only color (0:1, blue dashed) allocations.

As we observed before, the gains with respect to the baseline (bars above the 0 line) are quite modest, and tend to saturate between the 1:9 and 1:19 allocation schemes in the case of color gains, and between 9:1 and 19:1 in the case of geometry gains. However, steep losses in dB are observed as the distortion allocation schemes become more unbalanced. For content *longdress*, for instance, we observe a loss of -5.96 dB in the geometry domain when the 1:19 weighting ratio is selected, whereas the corresponding gains in terms of color distortions are limited to 1.08 dB (see Figure 10.6 (c)).

A visual comparison for the weight ratios 1:1, 1:19, and 19:1 at the highest bitrate under consideration is shown in Figure 10.7 for the content *longdress*. It can be observed that the geometry distortion introduced by changing λ_g from 2500 to 250, is not heavily influencing the visual perception of the content, despite the reported loss of 2.5 dB. However, in the case of distortion allocation of 19:1, the artifacts in the color domain heavily degrade its appearance, effectively masking any improvements brought in the geometry domain.

Figure 10.8 shows a visual comparison for the same allocation ratios, for content *guanyin*, at the second lowest bitrate under exam. It can be seen that for a weight ratio of 1:19, geometric artifacts in the form of holes appear (see Figure 10.8 (c)), whereas assigning larger weight to geometry distortion term brings a very poor performance in color compression. The 1:1 allocation, in this case, represents a compromise between geometry and color distortions.

Results show that, while performance gains can be achieved in either geometry or color domain by assigning larger weight to the corresponding type of distortion, they come at the cost of a loss in the other domain. Moreover, losses are generally more pronounced, whereas gains remain modest even when remarkably imbalanced allocation schemes are employed. The selection of the best allocation scheme must be conducted by examining which domain leads to perceptually more pleasant results, and by carefully considering whether the gains in one domain outweigh the costs in the other.

10.3.2 Unified network against separately trained networks

For the separately trained networks architecture, two models are employed, each dedicated to compress a particular type of attribute. In our context, we train a model on geometry-only compression and a second model on color-only compression. The testing point clouds are compressed by initially feeding the geometric information of the point cloud data into the geometry-only encoding network, in the form of individual blocks, as described in section 10.1.1 using $C = 1$. The de-compressed blocks are reassembled to restore the encoded point cloud topology. Then, a re-coloring step is applied by associating the original color values to the de-compressed coordinates using the nearest neighbor algorithm. The resulting point cloud is partitioned again into blocks (input channels $C = 4$) and fed to the color-only encoding network. The output blocks are eventually stitched together, forming the final decoded point cloud.

This implementation results in two bitstreams, each corresponding to a different type of attribute, which are both required at the received side in order to restore the encoded model. It should be noted that for the training of both networks, the same data and the same hyper-parameters adopted for the unified version and described in section 10.1.2 were applied. Moreover, a training and a testing block size of 32 and 128 were used, respectively.

Figures 10.9 and 10.10 report the performance evaluation results obtained with the unified network, with 1:1 allocation among geometry and color distortion terms, together with the results obtained from the separately trained networks on geometry and color. Performance is shown using the geometry metric D2-PSNR and the color metric YUV-PSNR, respectively. For the unified network, the parameters for distortion allocation 1:1 were used, according to Table 10.1. For the separately trained networks, parameter λ was set independently for geometry and color; curves are obtained by using (from smallest to highest bitrate), $\lambda_g = \lambda_c = 20, 100, 500, 2500$. Note that, to avoid redundancies, we only report the results for test contents *bumbameuboi*, *guanyin*, *longdress*, and *phil*, since for the rest of the models, very similar

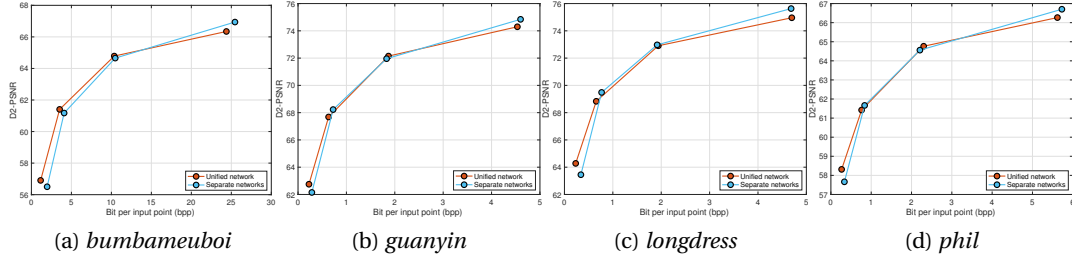


Figure 10.9 – Rate-distortion performance of the unified model and the separately trained networks, according to geometry metric D2-PSNR.

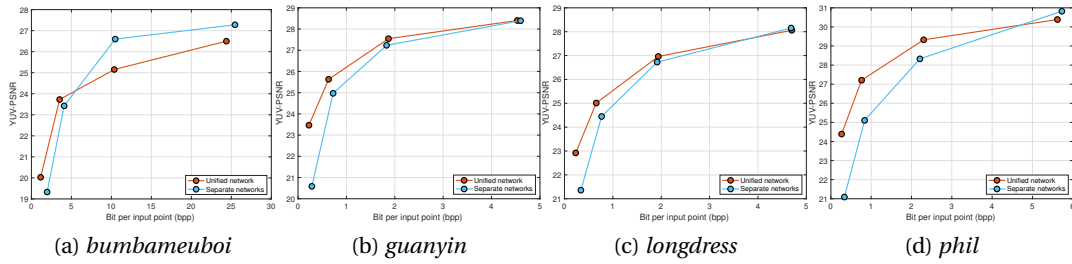


Figure 10.10 – Rate-distortion performance of the unified model and the separately trained networks, according to color metric YUV-PSNR.

behavior was observed.

Based on our results illustrated in Figure 10.9, similar performance is obtained when using the unified model to compress geometry information, with respect to employing an ad-hoc network which is trained on geometry-only data. The two solutions are interchangeable in terms of geometric distortions. In the color domain, however, a difference in performance can be observed between the two solutions, as shown in Figure 10.10. In particular, for three out of the four contents, i.e., *guanyin*, *longdress*, and *phil*, the two networks have similar performance for high bitrates, whereas for low bitrates, the unified model provides better performance. For *bumbameuboi*, though, notable gains can be observed for high bitrates, when a separate network is used to compress the color information. This might be due to the complexity of the model, both in the geometric and color domain, which might lead to diminished performance when the two types of information are considered simultaneously. Note that this constitutes a particularly sparse point cloud, which in general behaves as an outlier.

10.3.3 Benchmarking of unified network

In this section, we examine the performance of the unified network, which is selected as a superior approach based on the results of the previous section, against the anchor codec that was used in the MPEG point cloud compression-related activities. In Figure 10.11 and 10.12,

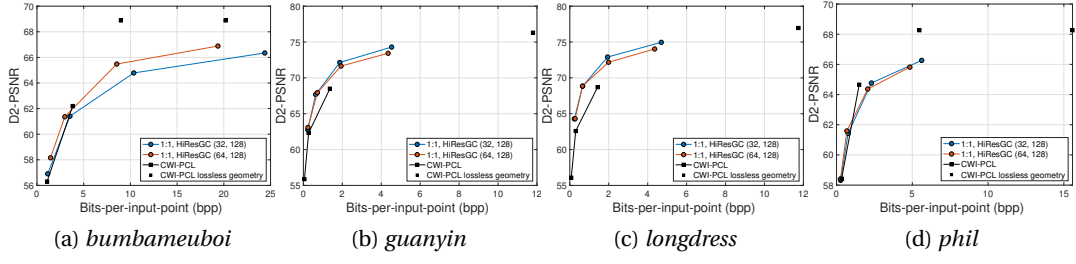


Figure 10.11 – Rate-distortion performance of the unified model, trained with block resolution of 32 and 64, against the MPEG anchor, according to geometry metric D2-PSNR.

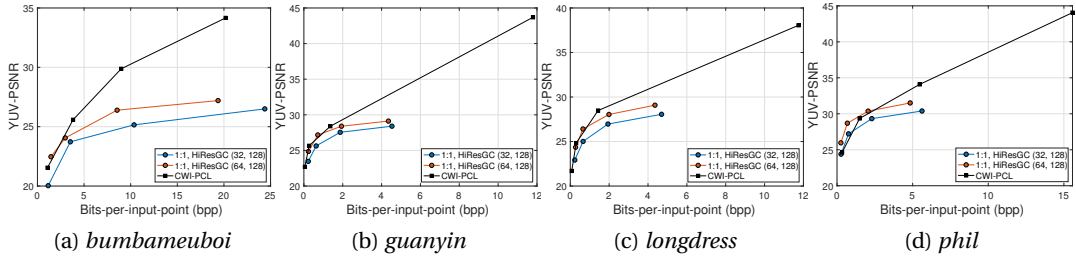


Figure 10.12 – Rate-distortion performance of the unified model, trained with block resolution of 32 and 64, against the MPEG anchor, according to color metric YUV-PSNR.

rate-distortion curves indicate the performance of the network using a training block size of 32 and 64, which is found to better exploit spatial redundancies (Wang et al., 2019) (see also section 10.4.1) and, thus, leading to lower bit rates for the same visual quality. For block resolution of 32, the λ values for geometry and color distortion were chosen according to the 1:1 ratio in Table 10.1, whereas for block resolution of 64, $\lambda_g = \lambda_c = \{80, 400, 2000, 10000\}$.

For the MPEG anchor, namely, CWI-PCL (Mekuria et al., 2017a), we opt for geometry compression octree bit-depths of 7, 8, 9 and 10 and for color compression JPEG Quality Parameter (QP) of 10, 50, 80 and 100, respectively, to obtain scalable visual quality levels by degrading both attributes simultaneously. Note that when the octree bit-depth is equal or higher than the corresponding voxel resolution of a content, lossless geometry compression is essentially applied; thus, leading to a PSNR value of infinity for geometric distortion. These cases are noted with simple markers on the figures to allow indicating the corresponding achieved bit-rates (see Figure 10.11, black squares).

It can be observed that for low bit-rates, the network achieves comparable or higher performance with respect to the CWI-PCL in terms of geometric distortions. Similar performance can be observed when considering color distortions, as depicted in Figure 10.12. In particular, training the network with blocks of resolution 64 leads to better performance with respect to resolution 32, and achieves comparable performance with respect to the CWI-PCL for low bit-rates. A quality saturation is shown for the network performance as the bit-rate is

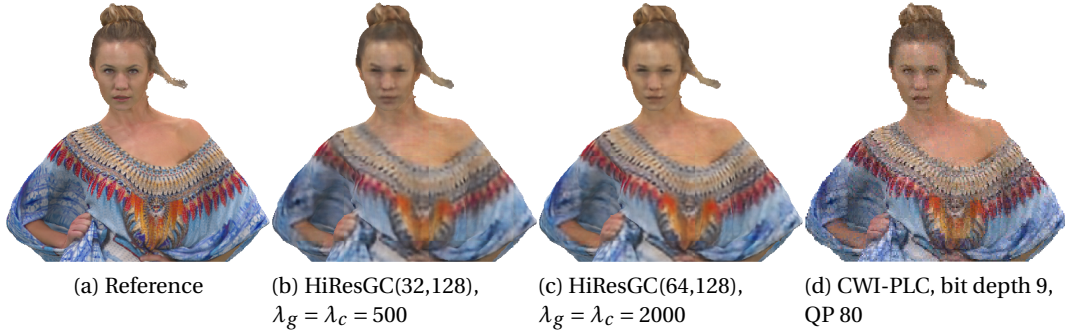


Figure 10.13 – Visual comparison for *longdress*, compressed using the proposed network and the MPEG anchor.

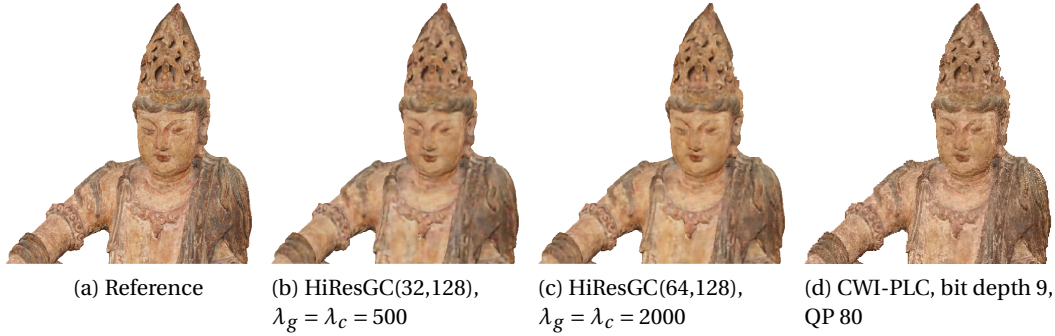


Figure 10.14 – Visual comparison for *guanyin*, compressed using the proposed network and the MPEG anchor.

increasing, indicating the need for more efficient architectures for compression at high fidelity.

Despite the similar quality values that are observed when considering the quality metric YUV-PSNR, visual comparison between the results obtained with the proposed model and the CWI-PCL show that markedly different distortions are introduced by the two compression solutions. Figure 10.13 shows a zoomed-in region of the content *longdress*, for the second-highest bit-rate. It can be observed that, whereas the CWI-PCL codec contains artifacts in the form of high frequency noise in the color domain, the network tends to have a smoother appearance, at the cost of a loss of detail. It can also be observed that increasing the block resolution from 32 to 64 leads to sharper results and more preserved details. A similar behavior can be seen for the content *guanyin*, as depicted in Figure 10.14. In particular, smoother texture is obtained when encoding with the network architecture with respect to CWI-PCL, as the former introduces artifacts in form of low-pass filtering, whereas false contours are present using the latter.

10.4 Meta-analysis

Neural networks represent a powerful tool to learn a compact representation of given data. As such, they have been largely employed to tackle compression for 2D visual data representations, and have recently been extended in point cloud data formats. However, a number of issues remains to be faced when considering compression of point clouds through neural network architectures, both when considering the distribution of the points in 3D space, and when trying to encode the accompanying attributes. In this section, we aim to shed some light regarding the influence and the selection for a number of hyper-parameters that affect the learning efficiency of a given network architecture. Note that the same network parameters and configurations specified in section 10.2.3.

10.4.1 Selection of training data for geometry compression

Inspired from the different approaches (Quach et al., 2019; Guarda et al., 2019b) in the generation of training data for point cloud geometry compression, in this experiment, we aim to evaluate the impact of using different data sets and grid resolutions. In general, there are two main lines that have been reported in the literature for the generation of relevant training data. In the first approach (Quach et al., 2019), a mesh repository is employed and point cloud models are generated through sampling, and potentially voxelizing at a desirable grid resolution. Typically, the original mesh models are artificially generated, and represent full-shaped colorless objects. In the second approach (Guarda et al., 2019b), which is adopted in our experimental set-up, high-resolution point cloud contents are collected from available repositories. Such contents typically consist of either real-life acquired and synthetic point clouds that span across a variety of categories.

Provided that point clouds are generally comprised of a considerable amount of points, whose sheer size and irregular structure make them unsuitable for being directly handled by neural networks, a common choice is to apply voxelization and block partitioning at a low resolution. Nonetheless, setting a specific block size against another influences the performance of the network, as has been shown in previous studies (Wang et al., 2019). Adding attribute encoding increases the complexity, as they will necessarily depend on the underlying 3D structure to be encoded.

In this experiment, to account for the first approach, we use point clouds extracted from the ModelNet data set, as described in (Quach et al., 2019). The models are scaled and regularly sampled, before being voxelized at a specific geometric resolution. To analyse the impact of the geometric resolution on the performance efficiency, voxel grid resolutions of 32 and 64 are employed for every model. To account for the second approach, we use the HiResGC data set that has been defined for our experimental set-up (see section 10.2), using block resolutions of 32 and 64. In both cases, point cloud units that contain less than 500 occupied voxels are discarded, and from the remaining data, a number of 10,000 is randomly sampled. In summary, we use four different training sets of 10,000 colorless samples: two are extracted

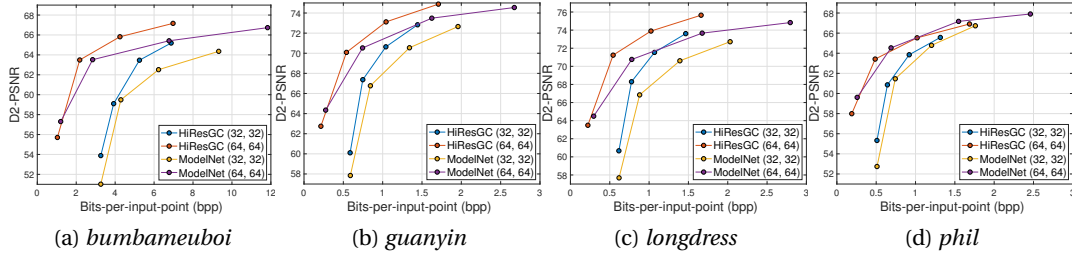


Figure 10.15 – Rate-distortion performance of the geometry-only network, using different data sets and training data resolutions, according to geometry metric D2-PSNR.

from the ModelNet data set and the other two from our generated data set, with grid resolution of 32 and 64 each. In this experiment, the testing models are partitioned in blocks of the same resolution as the one that was used for the training data (32 and 64).

In Figure 10.15, performance evaluation for 4 out of the 6 testing models is illustrated, using both data sets for learning, at both grid resolutions. It can be observed that better compression efficiency is achieved by the network when trained with the HiResGC, when compared to the ModelNet counterpart. Moreover, there is a clear trend of reaching higher performance when using a block resolution of 64, under both training sets. It is worth noting that the gains in compression efficiency come at the cost of higher demands in terms of resources and time, as blocks of resolution 64 require more computational power.

10.4.2 Resolution of testing data

The choice of a given block resolution for training data does not imply that the same grid size must be selected for the testing data. In fact, larger testing blocks can be chosen for compression, denoting another parameter that can potentially affect the reconstruction quality of point clouds. In this experiment, as a first step, we quantify the performance of our network in geometry compression by using different grid resolutions for the testing data. For this purpose, we use 4 different variations of the network, trained with the HiResGC and the ModelNet data sets and training blocks of size 32 and 64. The selected resolutions for the testing blocks under evaluation were set to: 32, 64, 128, and 256. In a second step, the HiResGC data set and a training block size of 32 is employed to examine the quality levels of the reconstructed color using testing block sizes of 64, 128, and 256.

In Figure 10.16, performance evaluation results for the geometry-only network are illustrated, showing rate-distortion curves for 4 out of the 6 testing models; very similar results are obtained for the rest of the contents. First row represents results obtained with networks trained with a block resolution of 32, whereas the second row depicts results with training block resolution of 64. As can be observed, in both cases testing grid resolutions of 64 and 128 achieve the best results. Similar conclusions are obtained when using the ModelNet data set to train the networks, at a generally more modest overall performance.

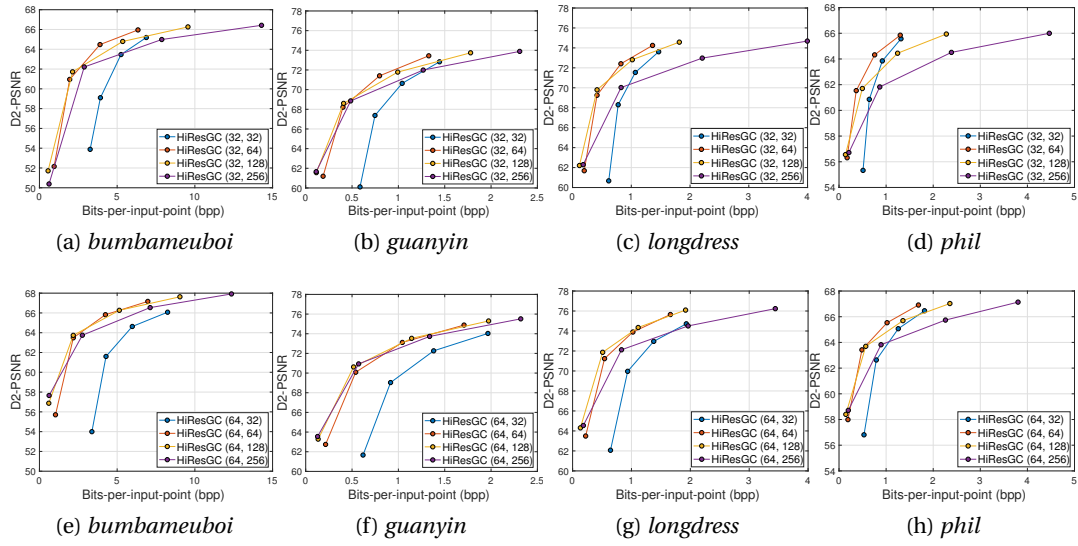


Figure 10.16 – Rate-distortion performance of the geometry-only network, for different testing grid resolutions, according to geometry metric D2-PSNR. First row represents results obtained with a training block resolution of 32, whereas the second row depicts results with training block resolution of 64.

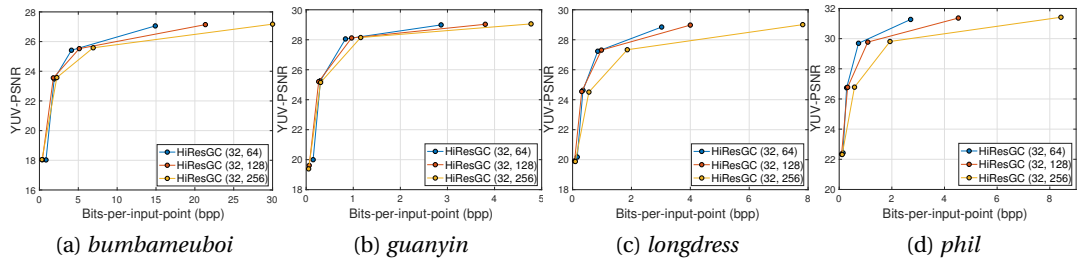


Figure 10.17 – Rate-distortion performance of the color-only network, for different testing grid resolutions, according to color metric YUV-PSNR. In parenthesis, the training data resolution that was used for the learned model.

In Figure 10.17, we present the performance evaluation results for the color-only network, for the same 4 contents. As can be seen, increasing the testing resolution leads to performance saturation, as equivalent quality the influence of border artifacts, which appear due to the block partitioning step, is not necessarily captured by the objective quality metrics. Moreover, the independent encoding/decoding of blocks might lead to different color distributions exhibiting among neighboring regions, which is a quite visible and annoying visual degradation for colored point clouds. Naturally, smaller block resolutions would lead to a more evident appearance of this effect, despite the fact that identical quality scores are obtained at the different testing resolutions.

10.4.3 Color space

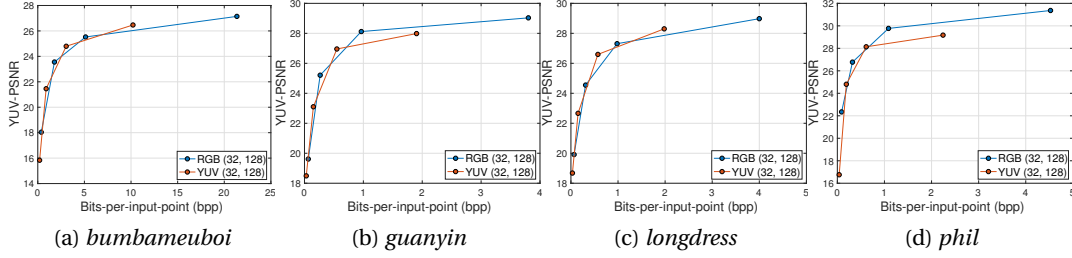


Figure 10.18 – Rate-distortion performance of the color-only network, for different input color spaces, according to color metric YUV-PSNR.

Another parameter that could potentially affect the results of color learning is the type of representation the textural information is provided to the network. Convolution neural networks typically learn local features and optimize filter weights in order to achieve data-driven compact representations. However, it is unclear whether using different bases in the network can effectively influence the results. In this experiment, we opt to examine the performance of the network when using the RGB and the YCbCr/YUV color spaces. The latter has effectively been used in classical image and video compression, while the first depicts the most widely-used color format that has been used in machine learning applications.

For this experiment, we used the ITU-Recommendation BT.709-6 (ITU-R BT.709-6, 2015) for conversion between RGB and YUV. The RGB color values for both training and testing data sets were converted to YUV, and then normalized between 0 and 1. Note that no color conversion is applied at any layer of the network. Thus, the loss function is always computed in the corresponding input color space. Results of the comparison between RGB and YUV are depicted, for 4 out of the 6 contents, in Figure 10.18. It can be observed that in general, both color spaces have similar performance. Slight gains can be observed at high bitrates when the RGB color space is employed, for the contents *guanyin* and, more remarkably, *phil*. Thus, it appears that color space selection does not have a large impact on the compression performance of color attributes.

10.4.4 Loss function

The performance of neural network architectures is affected by the choice of the loss function that is used to train a model. In order to assess whether performance gains could be obtained by using a different loss function for computing distortions in the color domain, we tested three different objective quality metrics, namely, l_1 , l_2 , and SSIM, with the former two denoting the most popular approaches that are used in similar network tasks. To obtain the loss value, the corresponding distance (l_1 , or l_2) is computed between the color channels of the original point cloud and the recovered point cloud across the input point coordinates. For the computation of the SSIM, which denotes a more perceptually-relevant metric, the same

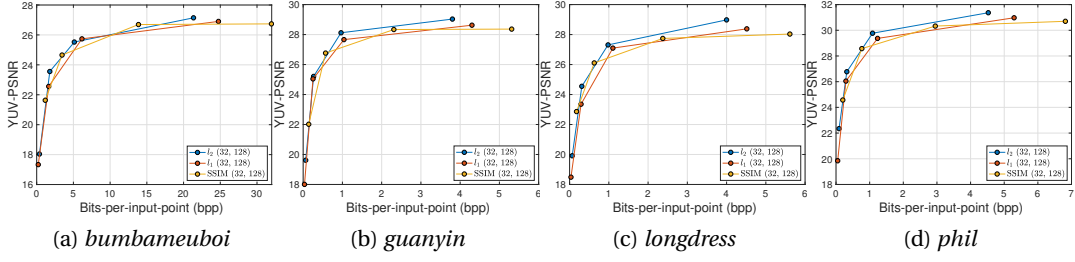


Figure 10.19 – Rate-distortion performance of the color-only network, for different loss functions, according to color metric YUV-PSNR.

equation as in (Wang et al., 2004) is used. However, instead of computing the metric on the Y channel, we decided to use it for all RGB channels. In (Zhao et al., 2016), it was shown that using SSIM on RGB channels could reflect the quality of the recovered images. Moreover, a filter size of 6, instead of the default 11, to reduce computational costs. To be consistent with the other losses, we applied some simple manipulations to make the range of the loss be within 0, indicating no error, and 1, indicating the largest possible error. As a result, the SSIM loss is defined as follows:

$$L_{SSIM} = \frac{1 - SSIM}{2} \quad (10.5)$$

For all loss functions under exam, the logarithm function is applied at the output value.

Results of the evaluation of different loss functions for color attributes are depicted in Figure 10.19, for 4 out of 6 contents in exam. It can be observed that all loss functions show similar performances. Slight gains can be observed when using l_2 at high bitrates. Thus, it can be concluded that in our set-up, the choice of the loss function does not seem to have a significant impact on the performance of the network under exam. However, the l_1 or l_2 would be the most compelling choices, considering the reduced costs with respect to SSIM.

10.5 Conclusions

In this study, we present a proposed neural network architecture to simultaneously handle the encoding of geometry and color attributes of point cloud contents. In principle, our efforts can be interpreted as a first attempt to compress both geometry and texture of point clouds using convolutional neural networks. Several parameters are examined, and conclusions are drawn regarding their efficiency, paving the way for next attempts. Our network competes with the anchor encoder that was employed in the MPEG activities; however, there is a large, unexplored space that can lead to further improvements. For instance, provided that a point cloud model is split into a series of blocks that is handled independently, due to memory and computational limitations, there is no effort in learning redundancies between neighboring blocks, by enabling for instance intra or inter prediction techniques. Moreover, it has been seen that variational auto-encoders applied on the feature space can remarkably assist by

improving the learning efficiency for the entropy model; such an addition is not tested in our network. Finally, it is well-known that high-quality training data are required for better performance; the availability of well-established training data sets with a representative range of geometric and textural complexities are of crucial importance, and would facilitate future efforts.

11 Conclusions

In this dissertation we address challenges in perceptual quality assessment and compression of point cloud imaging. Our focus lies in how people perceive distortions that commonly exhibit in this type of content representation. Starting from geometry-only models and approximations of degradations from acquisition and compression artifacts, we proceed to richer, colored models of higher practical use subject to distortions introduced from state-of-the-art codecs. Subjective and objective quality assessment methodologies are proposed and underlying influencing factors are examined and discussed. Finally, after reviewing and evaluating the state-of-the-art in point cloud compression, we propose our solution based on a deep-learning approach suitable to encode both geometry and color information.

Below is a summary of contributions that is accompanied by respective conclusions and potential limitations in the corresponding experimental set-ups that were employed. Following the manuscript partitions, they are clustered in three main parts.

Measuring perceptual quality: In this part we focus on subjective quality evaluation methodologies for geometry-only and geometry-plus-color point cloud data.

- We propose the use of interactive subjective evaluation methodologies, extending corresponding test methods to accommodate the richer nature of 3D models. This approach leads to different visual experiences among users, provided that each subject was free to interact at will. To compensate this uncertainty, we allowed interactions without imposing any time limitation before users submitting their judgement. Despite this uncontrolled factor that is introduced in our tests, we advocate that such methodologies are better adjusted to the interactive nature that comes with such richer contents. For example, by fixing the distance by which a model is inspected, artifacts that appear in closer views are not evaluated. By allowing the users to choose their preferred viewpoint, such views are not excluded, but neither imposed. Thus, the final judgement will be an overall score that inherently contains the preferred type of navigation. In all cases, the level of uncertainty can be estimated by the CIs, which in our tests never extended to

unreasonable ranges.

- The impact of different test methods in subjective quality assessment of point cloud topology in a desktop set-up was examined. Results suggest good correlation between two of the most widely employed protocols, namely ACR and DSIS. This comes despite the challenging nature of the set-up in terms of perceiving the underlying shape of the models using raw points for rendering purposes. In the latter case, subjects were found to base their opinions in differences between the number of points that exhibit between the two stimuli that were displayed simultaneously. In the former case, the impairments introduced in the model approximation were solely assessed, however, common cross-content biases were identified, leading to less consistency and higher uncertainty in the obtained quality scores. Our observations lead to the conclusion that both test methods can be employed for testing purposes, each having its own practical use. Yet, as in other imaging modalities, it is confirmed that the DSIS provides a more consistent approach to identify impairments.
- The impact of different display equipment was evaluated in subjective quality assessment of point cloud topology. In particular, noise and compression artifacts were assessed in a desktop set-up and in AR with 6DoF, using point primitives of minimum size and always showing the reference content. Results suggest that different types of degradation might be rated differently when consumed by different devices. For instance, strong correlation was observed in the case of noise, whereas in the presence of structural loss, the correlation worsened. This can be explained by the fact that certain types of degradations are easier to perceive when compared to others. In this case, there is higher tolerance for making predictions about the perceptual quality in different inspection scenarios. However, to ensure applicability of the results, subjective evaluation should be performed on the targeted equipment.
- Experimentation with surface reconstruction as a rendering mechanism for point cloud data shows different rating trends when compared to the usage of point primitives for display purposes. Considering that reconstruction steps are commonly lossy, while the final visual outcome depends on the selected algorithm and the configuration that is employed, uncertainty is introduced. Our results show low correlation when comparing scores obtained from evaluation of the same stimuli under point cloud degradations with point-based and mesh-based rendering. Converting point clouds to mesh representations denotes a viable approach that could be incorporated in the rendering pipeline. However, this approach could be simply translated to a problem of identifying optimal solutions for reconstruction, and applying corresponding quality assessment methodologies on the obtained mesh. This would mean to transfer the problem from the point cloud domain to the mesh domain, which is a well-studied field by the computer graphics community. However, this is out of our scope for our thesis, which is focused on explicit point cloud data representations.
- Simple implementations of point-based rendering solutions making use of geometrical

shapes that replace point samples were examined for colored models. The size of the shapes was adjusted dynamically based on local densities, thus, leading to perception of watertight models. Results show that solutions leading to perception of sharper details are preferred. In particular, spherical primitives were identified as a versatile solution that outperforms disks and cubes. This comes as a result of being efficient in filling holes due to their volumetric size, with reduced visual artifacts when compared to the alternatives, due to their geometric nature. Moreover, their usage doesn't require the presence of normal vectors. However, it denotes an expensive solution in terms of computational costs. Based on human opinions, sharper details are clearly preferred to more refined curves when displaying human figures. For objects, the advantages are mitigated making less expensive solutions, such as disks, plausible alternatives. On this matter, however, it should be noted that the topology and the texture of the selected models might have assisted in obtaining these results. Finally, it was concluded that the resolution of the contents did not affect the splat shape preferences.

- Statistically equivalent results were obtained from the usage of two different point-based rendering schemes that were employed to display point clouds under compression artifacts in the context of subjective quality assessment. The two rendering methods, namely, splat-based and voxel-based lead to very different visual artifacts. In particular, using cubic rendering primitives of adaptive size in the former, results in rougher surface approximations for sparser contents. In the latter, the mapping between voxels and image pixels that are displayed onto the screen, leads to the perception of holes. The experiment was conducted using a single codec, however, a combination of geometry and color distortions were selected to encode the models. Results show strong correlation between the subjective scores that were collected from both experiments. This suggests that human opinions on the level of impairment introduced from this particular encoder are not be substantially affected by the selected point-based rendering schemes. It should be noted, though, that the types of artifacts occurring in the point cloud topology from the selected codec (octree decomposition), denote a simple, less-demanding set-up to draw safe conclusions. Evidently, further experimentation is required using more encoding engines that lead to different visual distortions to evaluate the generalizability of these conclusions. Nonetheless, octree decomposition denotes one of the most popular baselines for state-of-the-art point cloud compression.
- We experimented with VR to enable reproducible and fully-controlled environments that can record unconstrained interactions of users in 6DoF with 3D models. In particular, we designed virtual scenes that served the purpose of the experiment, enabling high sense of realism (i.e., illumination, shadows, appearance of point cloud model) and intuitive controls. Rendering tools to adjust the appearance of point cloud models through a set of configurable parameters were developed and released. The developed software was employed in proof-of-concept subjective quality experiments for the evaluation of the color encoding modules that are integrated in G-PCC. Two subjective evaluation protocols were adjusted in the nature of the experiments, and compared. Based on our

results, the two encoding modules were found to be statistically equivalent. Moreover, a proposed variation of the sequential DSIS test method that allows re-visiting the reference and the distorted model at will, was found to be more reliable, faster and it was generally preferred by subjects. Finally, analysis of the users behavior showed that subjects tend to prefer close-range, frontal views of the models.

- An earlier version of the same VR framework was employed to conduct an eye-tracking experiment where human subjects were able to inspect point cloud models in a task-dependent protocol with 6DoF interactions. This is the first study on the field that considers point cloud models, addressing several unforeseen challenges. The experimental set-up consisted of a VR headset which was equipped with an eye-tracking device. The users were asked to inspect the models and order them in a criterion of their preference, thus promoting engagement with the content. The head and gaze data were recorded in real time, as users were navigating in the virtual scene. A main drawback of our set-up was the inability of the eye-tracking device to adjust to headset slippage caused by head movements. Thus, one of the main contributions lies in a methodology that was developed based on error profiling, which was effectively applied to improve the accuracy of our results by exploiting high-quality gaze measurements. Moreover, to overcome the limitations of using point samples for rendering, which do not allow colliding in the 3D space, a method was devised to decide frontal regions and exclude occluded parts, provided the position and orientation of the camera, and the position of the model. Taking under consideration the error of the gaze measurements, fixation density maps were generated in the form of importance weights. Results confirm trends that have been observed in other imaging modalities; that is, users are attracted by edges, contrast, regions that pop-up and, in the case of human figures, faces. As a limitation of this approach is noted the absence of evaluation of the heuristic algorithm that was developed in order to decide on the angular error of each gaze measurement, based on the error profiling results. Relevant tools were later developed along with further improvements that were integrated in an application paradigm for inspection of point cloud models in a VR museum.

Predicting perceptual quality: In this part we focus on objective quality evaluation methodologies for geometry-only and geometry-plus-color point cloud data.

- A new objective quality metric that operates on the point cloud domain was proposed based on the angular similarity of unoriented normal vectors between points that belong to a reference and model under evaluation. The metric, namely plane-to-plane, essentially measures the angular distance between local linear approximations of corresponding underlying surfaces. By pooling across individual similarity values from each pair of associated points that belong to the two models, a total quality score is predicted, capturing geometric degradations that exhibit in the distorted model. The limitation of the metric lies in its dependence on normal vectors that should be associated with the

coordinates of the point cloud models. Considering the ill-posed nature of this problem, the method was found sensitive to the selection and configuration of the normal estimation algorithm. To better understand the effect, we chose 3 widely-used algorithms, which were evaluated in terms of angular error against ground-truth normals. The same algorithms were additionally employed in our benchmarking analysis, in order to estimate normals based on which the metric was computed. Results show that the metric performs better when the normals are estimated at larger neighborhoods with respect to the ones that lead to low normal estimation errors. In fact, by enlarging the neighborhood sizes, smoother surface approximations are obtained, which can be interpreted as a low-pass filtering operation that removes high-frequency geometric components that may not be perceptually relevant. Moreover, it was found critical for the normals under comparison to reflect the same region of the content, hence, the range search variants were identified as a suitable solution for neighborhood formulation. Finally, adjustments of the neighborhood size per content appear to be beneficial in order to improve generalization capabilities. That is, larger regions should be used with higher resolution contents, and models with more irregular topology; from this point of view, the neighborhood size can be interpreted as a regularizer. Our results show that the performance of the proposed metric competes or outperforms current geometry-only algorithms in several data sets that were recruited, under proper configuration of the selected normal estimation algorithm.

- A second point-based metric was introduced, namely PointSSIM. It is based on statistical dispersion estimation of the distribution of quantities that are defined per point cloud attribute and reflect corresponding local properties. Specifically, the metric relies on the extraction of local features that capture structural similarity of location, normals, curvatures, and color attributes. This way, the operational logic of the well-known SSIM is extended to a higher dimensional, irregular space of a volumetric content, incorporating not only color, but also topological coherence. As part of the metric, a voxelization step is proposed that precedes feature extraction and simulates distant inspection. Our results show that depending on the data set, the application of the metric on certain attributes might be more efficient than others. The color-based features, which essentially consist of luminance-based local statistics, were found to be the most consistent across the examined data sets, achieving high performance. To our view, the main reason for this result is twofold: (a) luminance-based measurements are well-correlated with degradations that appear in color information, and (b) the formulation of local neighborhoods for feature extraction enables an implicit integration of the model's topology and corresponding geometric impairments in the obtained values. Moreover, the activation of the voxelization module can eliminate cross-content density variations, denoting a powerful tool. Combinations of the target voxel resolution and the neighborhood size can be exploited in order to obtain measurements that capture distortions of the model at different scales. Our benchmarking results using several subjectively annotated data sets show that PointSSIM achieves state-of-the-art performance, under proper configurations.

- Image-based metrics allow to capture both topology and texture distortions, as reflected by the corresponding rendering application in a holistic way. Making use of highly-sophisticated conventional 2D metrics, they denote a candidate solution that can provide quality predictions on model views that are experienced by users. To assess the performance of image-based metrics, an objective quality evaluation framework was defined and two subjectively annotated data sets that contained the same stimuli under both geometry and color degradations were recruited to assess generalization capabilities. The impact of removing the background information from the captured model views and the effect of enabling additional viewpoints for the computation of a predicted quality score was also examined. Our results show that image-based metrics achieve good correlation in both data sets under examination. Moreover, it was found that applying the computations on the foreground improves the performance, while also, we concluded that image-based metrics may attain accurate predictions even when employing a single view.
- We experimented with the integration of interactivity information recorded from users during subjective quality assessment in the computation of image-based quality predictions. In a first attempt, we simply pooled individual objective scores across all frames that were inspected by subjects. The large fluctuations that were observed in the objective scores between close and further views within the same session, though, led to high uncertainty and sub-optimal performance. Thus, we devised a strategy to translate the interactivity information as importance weights assigned to model views from a given camera layout, associating higher weights to viewpoints that were more frequently visited by users. This way, we fixate the distance between the camera and the model, which regularizes the scale of the objective quality scores, and the number of model views over which a total degradation score is obtained for the model under evaluation. The latter approach was found to bring substantial gains in terms of performance and computational resources.
- As a last contribution, in this category falls a benchmarking study that was carried out to evaluate the performance of the state-of-the-art objective quality metrics. For this purpose, the subjectively annotated M-PCCD data set was recruited, which consists of a rich set of diverse models subject to MPEG compression distortions. The analysis was issued over the entire data set, and repeated after clustering the stimuli per codec, per type of content, and per content, in order to obtain further insights. Our results show that, in every testing case, the newly introduced PointSSIM and the PCQM achieve highly accurate predictions with marginal differences, outperforming the other algorithms under examination. The local pooling in the luminance component that both make use, is assumed to be the main reasoning behind this result. The majority of the alternative metrics was found to be limited by their generalization capabilities across different contents, and across different codecs. In the latter case, it is evident the weakness to capture artifacts introduced by V-PCC. Regarding the former case, point-to-point attribute comparisons and image-based approaches were the most vulnerable. In

less challenging set-ups that consider variations of the same content, or stimuli that fall in the same type of content, remarkable improvements were observed across all metrics. Noteworthy are the performance gains of PSNR_Y, which achieves the best results when the stimuli are clustered per content, implying the efficiency of luminance-based measurements, while emphasizing the inability of point-to-point associations to generalize to different topologies.

Towards efficient compression: In this part we focus on compression of point cloud data.

- A large-scale quality evaluation study was carried out in two inter-continental laboratories to benchmark the state-of-the-art MPEG test models. A wide set of point cloud contents with diverse characteristics was recruited. The V-PCC and all variants of G-PCC were employed and configured based on the Common Test Conditions specified by the MPEG experts. A web-based interactive rendering solution was developed for the purposes of the experiment, which was released. The models were displayed using screen-faced points of adaptive size, ensuring the perception of watertight models. Results show the superiority of the V-PCC at low bit-rates. It is also remarked that using this codec, transparency is not achieved. To further analyse the performance of the G-PCC geometry modules, a second experiment is conducted using pairwise comparison. Results show that the Octree encoding module is preferred to TriSoup configurations at both low and high quality levels. Finally, to address rate-allocation aspects between geometry and color information, a third experiment with a similar set-up was conducted. Results show that for high bit-rates, geometry is considered more important, whereas at low bit-rates, color enhancements are preferred.
- A deep learning-based convolutional neural network is proposed to encode geometry and/or color of point cloud data. The architecture is rather generic and essentially extends current developments on the field. The encoder operates on the 3D domain making use of 3D convolutions to extract features from point clouds in a block-by-block basis. The contribution lies on the ability of the proposed scheme to incorporate color information, which leads to generation of feature maps that express a point cloud in a more holistic way. The latter denotes the so-called unified network that is trained on colored models, and can be adjusted to better preserve geometry or color attributes. The unified network is compared to two separately trained networks, one dedicated to geometry and the other to color, to understand if one architecture brings more benefits with respect to the other. Results show that the unified architecture achieves better results in color information, mainly, at low bit-rates. The latter is then compared to the MPEG anchor, showing competitive results, with better performance at low bit-rates. The use of convolutional layers for color encoding limits the performance at higher bit-rates by the presence of blurriness artifacts. A number of parameters that affect the network performance are also examined as part of the study. The most noteworthy results indicate that, as expected, the training data set affects the performance of the

network, better performance is observed as the block size is increasing, whereas adopting a different color space, or the SSIM over a simple l_2 norm in the loss function does not bring any advantage.

Future aspects: Point cloud quality assessment can still be considered at its infancy. The same is true for network architectures dedicated in point cloud compression. Despite recent developments, there are several aspects on how research that was conducted can be extended in the future.

One future objective is to involve more sophisticated point-based rendering schemes for subjective quality assessment of point clouds, that allow more realistic content representations. It is in our aims to experiment with visual attention in VR in order to better understand how people consume 3D models in more interactive scenes. The developed VR museum, could offer a starting point. In the same line, it would be valuable to experiment with the design of more complex virtual scenes. For instance, quantifying and predicting differences in subjective behavior (i.e., quality assessment or other tasks in VR) in the presence of distractions from a high-quality scenery, could be envisioned. Moreover, the performance of objective quality metrics can be further improved by incorporating more sophisticated multi-scale approaches that better combine geometry and color. The PointSSIM offers a good basis. Further developments can be finally viewed for our baseline compression scheme with more sophisticated components for a high-performing auto-encoding architecture.

Annexes

A Statistical analysis tools

Measurements of perceived quality are fundamental in the context of multimedia services and applications. Quality scores can be obtained by either subjective or objective means. The first provide ground truth information, whereas the latter provide predictions regarding the visual quality of the multimedia content. Regarding subjective data, they are typically collected in experiments with the participation of human observers. Rating distributions are formed and need to be analysed in order to decide on the quality level of each stimulus. Moreover, the same testing material might be assessed in different sessions, or experiments, thus, making it valuable to understand whether the obtained results agree. Finally, objective algorithms need to be benchmarked against ground truth subjective scores, in order to decide on their prediction accuracy and robustness.

The aforementioned reasons make clear the necessity for tools that allow rigorous scrutiny of quality assessment results. In this chapter, we describe methodologies and performance indexes inspired by (ITU-T P.1401, 2012; ITU-T J.149, 2004; Hanhart, 2016), and employed for analysis purposes in the context of this thesis. In particular, we initially detail the metrics that are used to characterize visual quality based on human ratings. Then, we describe the methodology that is followed to compare the results of two experiments, completing with the procedure adopted to benchmark objective quality metrics.

A.1 Processing of subjective scores

Human scores from subjective testing need to be analysed in order to draw conclusions regarding the validity of the experiment and the quality of the multimedia content. For instance, provided that a limited number of people usually participates in subjective experiments, statistical analysis tools can be employed in order to identify whether the conclusions that are drawn can be generalized. Moreover, we can understand the impact of the degradations on the perceived quality, as well as how two stimuli are compared to each other. In this section, we describe the procedures and the metrics that are employed to process and analyse subjective scores.

A.1.1 Category rating

The methodology detailed below is adopted in experiments where a single or a double stimulus category rating test method is adopted. This is, when one, or two stimuli are presented to the subjects, which are asked to provide a score based on a selected rating scale. As an outcome of such experiments, a set of subjective scores is obtained, which are statistically analysed to characterize the quality of the testing material.

Outlier detection

For every experiment, outlier detection and removal is initially performed based on the subjective scores in order to exclude observers whose ratings deviate substantially from the rest of the participants. In our analysis, we follow the procedure that is described in Recommendation (ITU-R BT.500-13, 2012).

Based on this methodology, it is firstly determined whether the distribution of scores for a particular content is normal or not. Specifically, for each test content, if the kurtosis coefficient of the scores is between 2 and 4, the distribution can be considered as normal. Then, a confidence interval is defined and based on the number of occurrences of scores being outside of this range, a subject is rejected or not. In particular, if the scores are distributed normally, for each score larger than $2 \cdot \sigma$ from the mean of the scores (upper limit) of a stimulus i , a counter U_i is incremented. For each score smaller than $2 \cdot \sigma$ from the mean of the scores (lower limit) of a stimulus i , a counter L_i is incremented. In case of non-normal distributions, the upper and lower limits are set as $\sqrt{20} \cdot \sigma$ from the mean of the scores of a stimulus. Assuming a total of K number of stimuli, the scores of a subject are removed if the conditions of Equation A.1 are met.

$$\frac{\sum_{i=1}^K (U_i + L_i)}{K} > 0.05 \text{ and } \left| \frac{\sum_{i=1}^K (U_i - L_i)}{\sum_{i=1}^K (U_i + L_i)} \right| < 0.3. \quad (\text{A.1})$$

Mean opinion scores and confidence intervals

After outlier removal, the remaining ratings are employed to compute the mean opinion score (MOS) for a testing stimulus, based on Equation A.2

$$\text{MOS}_i = \frac{\sum_{j=1}^N m_{ij}}{N} \quad (\text{A.2})$$

where m_{ij} denotes the score given to stimulus i from a subject j , and N indicates the number of subjects.

For every stimulus, the 95% confidence interval (CI) of the estimated mean is also computed

assuming a Student's t -distribution, based on Equation A.3

$$CI_i = t(1 - \alpha/2, N - 1) \cdot \frac{\sigma_i}{\sqrt{N}} \quad (A.3)$$

where $t(1 - \alpha/2, N - 1)$ is the t -value corresponding to a two-tailed Student's t -distribution with $N - 1$ degrees of freedom, and σ_i denotes the standard deviation of the scores for stimulus i . The variable α indicates the significance level, which is typically set equal to 0.05 for subjective evaluations. The interpretation for the CI measurement is that if the same experiment is repeated a large number of times in the future using a random sample of the population, there is 95% probability that the CI that will be computed will contain the true value.

In principle, the MOS describes the total quality of a testing stimulus, whereas the CI describes the level of uncertainty of the corresponding MOS value.

A.1.2 Pair comparison

The methodology detailed below is adopted in experiments where a pair comparison test method is adopted. This is, when two stimuli are presented to the subjects, which are asked to provide their preference. As an outcome of such experiments, a preference matrix is obtained, which are statistically analysed to characterize the quality of the testing material.

For a comparison set containing n different types of classes under comparison, C_1, C_2, \dots, C_n , there are $\binom{n}{2}$ pairs to be compared. The comparison results for the set can be summarized by a matrix of winning frequencies $\{w_{ij}\}$. By equally splitting a tie in half between the two preference options, $\{w_{ij}\}$ is computed based on Equation A.4

$$w_{ij} = p_{ij} + t_{ij}/2 \quad (A.4)$$

where p_{ij} is the number of subjects who preferred C_i over C_j and t_{ij} is the number of subjects who rated them the same.

In order to obtain continuous scale quality score values for C_i 's from the matrix of winning frequencies, we use the Bradley-Terry-Luce (BTL) model that is frequently applied for analysis of pair comparison data. In this model, the probability of choosing C_i against C_j , is expressed by \mathbb{P}_{ij} and is represented as:

$$\mathbb{P}_{ij} = \frac{w_{ij}}{w_{ij} + w_{ji}} = \frac{\pi_i}{\pi_i + \pi_j} \quad (A.5)$$

where π_i is the quality score of C_i , which is referred to as the true rating in the literature, and provides an estimation for the MOS. Every $\pi_i \geq 0$ and $\sum_i \pi_i = 1$. The π_i 's can be estimated by the maximum likelihood estimation based on the empirical probability values \mathbb{P}_{ij} . As CIs, we use the standard deviations from the covariance matrix of the parameter estimates, and assume a Gaussian distribution at the 95% confidence level.

For analysis purposes, we normalize the quality scores of the instances in a comparison set so that the maximum MOS value becomes 100.

A.2 Comparison of subjective scores from different experiments

It is frequently required to compare the results derived from two individual experiments in order to draw conclusions regarding their statistical equivalence. There are several sources of bias that can affect the scoring distributions, even when the same set of stimuli is evaluated. For instance, different test methods, experimental set-ups, or viewing conditions, might lead to different conclusions regarding the visual quality of the testing material. Thus, it is essential to examine the influence of such factors and quantify the statistical relevance of the tests. In this section, the procedures and the performance indexes that are adopted to compare subjective scores from two experiments are described.

A.2.1 Data mapping

A regression model is suggested to be used before comparing quality scores between two experiments, in order to account for different rating behaviors that might be observed. In particular, according to Recommendation (ITU-T P.1401, 2012), even in the case of repeating the same experiment with the same observers and identical testing material, the results are expected to be slightly different. These deviations can be considered as noise that is associated with the obtained scores. Moreover, the subjective ratings are additionally affected by a series of contextual effects. The short-term contextual effect reflects the tendency of a human subject rating higher or lower the current stimulus, if the previous samples were of lower or higher quality, respectively. The so-called medium and long-term contextual effect, arises from the average quality of the entire sets of stimuli that are evaluated in two experiments. This can lead to different rating distributions and, specifically, to different quality scores for the same stimulus that might be assessed in both tests. For instance, in experiments that mainly contain testing material of poor quality, observers tend to score them higher, and vice versa. Finally, rating deviations can occur from long-term dependencies that account for different cultural behavior and multimedia exposure background. Despite the adoption of best-practices in experimental designs to avoid the influence of such effects (e.g., randomized presentation order, stimuli of quality that spans from lowest to highest quality, etc.), they can only be minimized to a certain extent without being entirely eliminated.

It is clear that the aforementioned issues lead to some degree of uncertainty regarding human ratings. Moreover, it is frequent to observe additional discrepancies between the score distributions from two experiments, which can be classified as:

- Bias or offset: is observed when a constant offset exists between the MOS values. This can occur when the overall quality of the stimuli under evaluation is rather high, or low, which might lead observers to rate more pessimistically or more optimistically.

A.2. Comparison of subjective scores from different experiments

- Gradient difference: is observed when the scores tend to become more pessimistic faster in one experiment than in another. This effect is usually caused when the test does not have quality samples that cover the entire quality range.
- Ranking difference: is observed when the ranking of the scores for the same set of stimuli is different from one experiment to the other, which denotes the most serious problem.

The impact of the aforementioned effects on the score distributions can be identified through scatter plots that depict the MOS values from one experiment against the MOS values from the second. Substantial discrepancies will lead to a large spread between the data points, while a narrow monotonic relationship shows strong correlation. Several performance indexes are additionally employed in order to quantify the observed disagreements. Yet, in order to reduce their influence, without altering the ranking order which is considered the most important property, it is suggested to apply fitting functions before computing the statistics. For instance, in order to remove the bias and the gradient difference, a linear mapping can be applied in order to align the scores of one experiment to that of another. A monotonic third order polynomial mapping can be used to additionally linearize a “banana shape” that might be observed in the relationship between the MOS values from the scatter plots.

In our analysis, to map subjective scores before comparing two experiments, we use both linear and cubic fitting functions, according to (ITU-T P.1401, 2012) and based on the Equations A.6 and A.7

$$P(\text{MOS}) = a \cdot \text{MOS} + b \quad (\text{A.6})$$

$$P(\text{MOS}) = a \cdot \text{MOS}^3 + b \cdot \text{MOS}^2 + c \cdot \text{MOS} + d \quad (\text{A.7})$$

where $P(\bullet)$ symbolizes prediction, while a , b , c , and d denote the parameters of the functions that are determined using a least squares method and are constraint to ensure monotonicity. Moreover, we set the subjective scores from both experiments as ground truth.

To facilitate clarity, let us define the sets A and B to refer to the ratings collected from the corresponding experiments under comparison. Let us assume that the scores from set A are set as the ground truth, with the MOS of the stimulus i being denoted as MOS_i^A , while MOS_i^B is used to indicate the MOS of the same stimulus in set B . A predicted MOS for stimulus i , indicated as $P(\text{MOS}_i^B)$, is estimated after issuing a regression model to each pair $[\text{MOS}_i^A, \text{MOS}_i^B]$, $\forall j \in \{1, 2, \dots, K\}$, where K denotes the number of stimuli. Then, the performance indexes are computed using MOS^A and $P(\text{MOS}^B)$ in order to examine the statistical relevance of the two sets.

A.2.2 Statistical evaluation metrics

Relevant properties characterizing the relationship between the distributions of the two rating sets are investigated in order to draw conclusions regarding the outcomes of two experiments. In particular, we examine the linearity, monotonicity, accuracy and consistency of the results, using the Pearson linear correlation coefficient, the Spearman rank order correlation coefficient, the root-mean-square error and the outlier ratio, respectively. Note that these indexes employ the MOS values from the two experiments, and a mapping might precede the computations, as mentioned earlier. Hereafter, we refer to the potentially mapped MOS as predicted MOS.

Pearson linear correlation coefficient

The Pearson linear correlation coefficient (PLCC) quantifies linear relationships between two variables X and Y . The PLCC ranges in the interval $[-1, 1]$, where a value of 1 (-1) indicates the strongest positive (negative) correlation, whilst a value of 0 indicates no correlation. The formula to compute PLCC index is given in Equation A.8,

$$\text{PLCC} = \frac{K \cdot \sum_{i=1}^K x_i \cdot y_i - \sum_{i=1}^K x_i \cdot \sum_{i=1}^K y_i}{\sqrt{K \cdot \sum_{i=1}^K x_i^2 - (\sum_{i=1}^K x_i)^2} \cdot \sqrt{K \cdot \sum_{i=1}^K y_i^2 - (\sum_{i=1}^K y_i)^2}} \quad (\text{A.8})$$

where x_i and y_i denote the MOS and the predicted MOS values for stimulus i from the two experiments, while K indicates the number of stimuli.

The PLCC is a measure of linearity between the MOS values obtained from the two experiments.

Spearman rank order correlation coefficient

The Spearman's rank order correlation coefficient (SROCC) quantifies monotonic relationships between two variables X and Y . Monotonicity measures if an increase (decrease) in one variable is associated with an increase (decrease) in the other variable, independently of the magnitude. Intuitively, the SROCC between two variables equals the PLCC between the ranking order of those two variables. The SROCC ranges in the interval $[-1, 1]$, with ± 1 indicating absolute monotonicity. The formula to compute the SROCC index is given in Equation A.9,

$$\text{SROCC} = 1 - \frac{6 \cdot \sum_{i=1}^K (R(x_i) - R(y_i))^2}{K \cdot (K^2 - 1)}, \quad (\text{A.9})$$

where x_i and y_i denote the MOS and the predicted MOS values for stimulus i from the two experiments, K indicates the number of stimuli, and $R(\bullet)$ is a ranking relation (sorting) that is applied to the argument.

A.2. Comparison of subjective scores from different experiments

The SROCC is a measure of monotonicity between the MOS values obtained from the two experiments.

Root-mean-square error

The root-mean-square error (RMSE) quantifies the difference between two variables X and Y . The formula to compute the RMSE index is given in Equation A.10,

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^K (x_i - y_i)^2}{K - 1}} \quad (\text{A.10})$$

where x_i and y_i denote the MOS and the predicted MOS values for stimulus i from the two experiments, and K indicates the number of stimuli. The absolute difference between MOS and predicted MOS is also referred to as absolute prediction error.

The RMSE is a measure of accuracy between the MOS values obtained from the two experiments.

Outlier ratio

The outlier ratio (OR) quantifies the consistency between two variables X and Y , as the ratio between the number of outliers divided by the number of stimuli. An outlier is identified when the absolute prediction error $|\text{MOS}_i^A - P(\text{MOS}_i^B)|$ exceeds the 95% CI for stimulus i from the scores of experiment A . The formula to compute the OR index is given in Equation A.11,

$$\text{OR} = \frac{L}{K} \quad (\text{A.11})$$

where L is the number of outliers and K is the total number of stimuli evaluated in each experiment.

The OR is a measure of consistency between the MOS values obtained from the two experiments.

A.2.3 Estimation errors

To decide whether statistically distinguishable scores are obtained for the stimuli under assessment from two experiments, the correct estimation, the under-estimation, and the over-estimation errors can be computed. In particular, let us assume that the scores of set A are the ground truth. For every stimulus, the difference $P(\text{MOS}_i^B) - \text{MOS}_i^A$ for every stimulus i is estimated with a 95% CI, after a multiple comparison test at a 5% significance level. There are 3 possibilities:

Annex A. Statistical analysis tools

- If the CI contains 0, correct estimation is observed, indicating that the visual quality of stimulus i is rated statistically equivalently from both populations.
- If the CI is above 0, over-estimation is observed, indicating that the visual quality of stimulus i is rated higher in set B .
- If the CI is below 0, under-estimation is observed, indicating that the visual quality of stimulus i is rated lower in set B .

These computations are repeated for every stimulus. The results are aggregated and divided by the total number of stimuli, to account for the correct estimation (CE), under-estimation (UE), and over-estimation (OE) percentages.

A.2.4 Classification errors

To examine whether the ratings obtained from two experiments lead to different conclusions regarding the visual quality of a pair of samples, the correct decision, the false tie, the false differentiation, and the false ranking errors are computed. In particular, let us assume that the scores of set A are the ground truth. The true difference between the scores of stimuli i and j from set A is calculated as $MOS_i^A - MOS_j^A$ with a 95% CI. Depending on whether 0 lies below, in-between, or above the CI, there are three possibilities: (a) i is better than j , (b) i is the same as j , and (c) i is worse than j . This is repeated $\forall i, j \in \{1, 2, \dots, K\}$ with $i \neq j$, and K the number of stimuli. Similarly, the quantity $P(MOS_i^B) - P(MOS_j^B)$ is computed with a 95% CI, using the predicted scores from set B .

- When the outcome of the three-way classification from sets A and B agree for a pair of stimuli i and j , a correct decision is observed.
- When the outcome of the three-way classification from set A (i.e., ground truth) say that i is better than j , or i is worse than j , and the result from set B advise that i is the same as j , a false tie occurs. This is the least offensive error.
- When the outcome of the three-way classification from set A indicate that i is the same as j , whereas the result from set B dictate that i is better than j , or i is worse than j , a false differentiation occurs. This is a more offensive error.
- When the outcome of the three-way classification from set A suggest that i is better than j , or i is worse than j and the result from set B state the opposite, a false ranking occurs. This is the most offensive error.

These computations are repeated for every combination of pairs of stimuli (i, j) , with $i \neq j$. The results are aggregated and divided by the total number of combinations to account for the correct decision (CD), the false tie (FT), the false differentiation (FD), and the false ranking (FR) percentages.

A.2.5 Standard deviation of Opinion Score

The Standard deviation of Opinion Score (SOS) coefficient a parametrizes the relationship between MOS and the standard deviation associated with it. It is derived by considering how the standard deviation of subjective scores varies in relation to the given MOS. Assuming a MOS range of [1, 5], the minimum SOS coefficient will be found at the extremes of the scales, whereas the maximum variation will be observed for a MOS score of 3. From that, a square relationship can be assumed between the SOS and the MOS, as given in Equation A.12

$$\text{SOS}(x)^2 = a \cdot (-x^2 + 6x - 5) \quad (\text{A.12})$$

The parameter a indicates the level of dispersion of MOS scores, or, alternatively, to the amount of user diversity in the rating. Consequently, close values of a obtained in separate tests denote similarity among the distribution of the scores.

A.2.6 Inferential statistical methods

One of the main objectives of polling users in order to gather subjective scores is to gain insights on how the general population would rate the contents under exam. Inferential statistical methods are commonly used to analyse the data, serving the purpose of deducing properties of the underlying distributions of probability. As such, statistical models describing the data, and the population it is drawn from, are needed.

One main distinction can be asserted depending on whether the data distribution can be considered normal or not; in the former case, parametric tests (which use concepts such as comparison of means) are employed, whereas in the latter case, non-parametric tests are used. Commonly used tests to assess the normality of a distribution include Kolmogorov-Smirnov, Anderson-Darling, and Shapiro-Wilk.

Inferential statistical methods can be particularly useful to test the effect of one or more groups on the dependent variable, through hypothesis testing. In the case of MOS scores, for example, they can be used to determine whether one condition under test had a significant effect on the final score distribution. A null hypothesis and an alternative hypothesis are formulated, and a significance level α is defined. The result of the test is commonly expressed through a p-value, which indicates how likely the observed data is, according to the null hypothesis. If the p-values falls under the significance level, then the null hypothesis is rejected at α significance level; otherwise, the null hypothesis cannot be rejected. The effect size is usually reported as well, to give an idea of the magnitude of the effect on the scores. Larger effect sizes correspond to a greater impact on the distribution of the data, whereas small effect sizes indicate that the effect, if present, might not be considerable.

In case of comparisons among groups, one important distinction has to be made in terms of

hypothesis testing, depending on how the scores were collected:

- If the same participants assigned the scores to all the groups under consideration, the groups are considered paired. In this case, the analysis must take into consideration the fact that an implicit relationship exists between the groups, because the same participants contributed the data points. This is the case when examining factors in a subjective quality assessment campaign in which all users scored all stimuli, for example.
- If different participants assigned the scores to the groups under consideration, then the groups are deemed unpaired. In this case, it cannot be assumed that differences in scores among groups are directly due to the influence of the factor under test. A common consequence of this is the need of more users in order to draw meaningful results. This is the case when comparing subjective scores obtained in two different laboratories, for example.

In this manuscript we are mainly interested in comparing two groups, either paired or unpaired. When group means are compared, which is more suitable for normal distributed populations, paired or unpaired Student's t-tests are used. On the other hand, when group medians are being compared, which is more suitable for ordinal data, we use Wilcoxon signed-rank and rank-sum (also called Mann-Whitney's U) tests for paired and unpaired groups, respectively.

Student's t-test

The student's t-test is a statistical hypothesis test aiming at understanding whether there is a difference in the distribution model of the two groups. The data is assumed to be drawn from a normally distributed population.

If an unpaired t-test is performed, models are built for each group, and they are then compared to assess their difference. In this case, and considering a two-tailed test, the null hypothesis that is being tested is that the two models come from distribution with equal means, whereas the alternative hypothesis states that the two means are different.

In the occasion of a paired t-test, the difference among the two groups is first computed, and then a one-sample t-test is performed. In this circumstance, the null hypothesis assumes that the difference has a mean of 0, whereas the alternative hypothesis states that the mean is different from 0.

Wilcoxon rank test

The Wilcoxon family of tests include the rank sum and the signed rank tests, depending on whether the groups are paired or unpaired. They both compare the medians of the groups; as

such, they are suitable for ordinal data and for small sample sizes, for which the normality of the population cannot be assumed.

Similarly to what has been described for the Student's *t* test, the rank sum test ranks the data from each group, and then performs a comparison of rank means, whereas the signed rank test first computes the difference among the two groups, and then calculates the ranks for the differences. In the first case, the null hypothesis states that the distributions of both groups are equal, while the alternative hypothesis states that they are not equal. In the second case, the null hypothesis assumes that the difference among the ranks has a mean of 0, whereas the alternative hypothesis states that it is different from 0.

A.3 Comparison of objective against subjective scores

Objective quality metrics are essential for providing predictions of visual quality for content representations in several applications related to information and communication systems. The performance of a metric is characterized from its prediction power. That is, how accurate are the output scores with respect to subjective opinions for the quality of multimedia content. To be able to decide on the accuracy and reliability of an objective algorithm, benchmarking is required against ground truth subjective data. This evaluation procedure also allows performance comparisons between different objective quality metrics. In this section, the procedures and the performance indexes that are adopted to benchmark objective quality metrics are described.

A.3.1 Data mapping

To evaluate how well an objective metric is able to estimate perceptual quality, MOS computed from ratings of subjects that participate in an experiment are required and serve as ground truth. The metrics are typically benchmarked after applying a regression model in order to map the objective scores to the subjective quality range, while also to account for biases, non-linearities and saturations that might appear in subjective testing. In particular, let us define the result of execution of a particular objective metric indicates a Predicted Quality Score (PQS). A predicted MOS, denoted as $P(\text{MOS})$, is estimated by applying a fitting function on the $[\text{PQS}, \text{MOS}]$ data-set. In our analysis, we use the linear and the monotonic cubic fitting functions given in Equations A.6 and A.7, according to Recommendation (ITU-T P.1401, 2012), and the logistic fitting function given in Equation A.13, following Recommendation (ITU-T J.149, 2004)

$$P(\text{MOS}) = a + \frac{b}{1 + \exp[-c \cdot (\text{MOS} - d)]} \quad (\text{A.13})$$

where a , b , c , and d denote parameters of the function that are determined using a least squares method, after ensuring monotonicity in order to maintain the ranking order.

A.3.2 Performance indexes

To investigate the linearity, monotonicity, accuracy and consistency of an objective quality metric, the PLCC, SROCC, RMSE and OR performance indexes are computed, respectively. As described in A.2.2 and briefly recapped below:

- The PLCC is a measure of linearity between the MOS and the predicted MOS. The PLCC ranges in the interval $[-1, 1]$, where a value of 1 (-1) indicates the strongest positive (negative) correlation, whilst a value of 0 indicates no correlation. The formula to compute PLCC index is given in Equation A.8.
- The SROCC is a measure of monotonicity between the MOS and the predicted MOS. The SROCC ranges in the interval $[-1, 1]$, with ± 1 indicating absolute monotonicity. The formula to compute the SROCC index is given in Equation A.9.
- The RMSE is a measure of accuracy between the MOS and the predicted MOS. The formula to compute the RMSE index is given in Equation A.10.
- The outlier ratio (OR) is a measure of consistency between the MOS and the predicted MOS. The formula to compute the OR index is given in Equation A.11.

A.4 Comparison of rate-distortion curves

Objective quality metrics provide, for each stimulus under evaluation, an estimation of its impairment. In the case of compression (or other generic rate-optimization solutions), the stimuli are traditionally engineered to cover a sufficient range in terms of bit-rate, in order to assess a variety of corresponding distortions. Rate-distortion curves provide a useful interpolation of discrete tuples of distortion for a given bit-stream size, and allow to generalize the behavior observed on the single stimuli.

In such cases, it is desirable to obtain, from the curves under exam, a numerical value that quantifies the difference between the curves. In particular, we want to ascertain the bit-rate savings or the quality gains that we observe when we select one rate-optimization solution with respect to another. For the former case, this corresponds to quantifying, for a predefined level of distortion, the amount of rate that can be saved by using one solution; in the latter case, it provides the distortion decrease (or quality gain) that can be observed at a given bit-rate, when using one algorithm or the other. The procedure was formally defined in 2001 by Gisle Bjontegaard for PSNR values (Bjontegaard, 2001).

B Point cloud data structures

A point cloud can be defined as a set of points aiming to represent a 3D model. Each point is defined by coordinate positions in 3D space. Additional attributes, such as color, normal vectors, curvature and reflectivity can be associated, among others, to reflect measured properties of the underlying 3D surface.

Point clouds denote a visual modality that can efficiently acquire, encode and render advanced content representations. Thus, it has recently drawn a significant amount of interest by the scientific community and industrial market. There are several different ways to acquire point cloud data, such as depth sensors and photogrammetry. Independently of the acquisition technique that is employed, though, the geometric structure of a captured or extracted point cloud is, in principle, irregular. This means that the coordinates of a point cloud are real numbers of any precision that can span any range, depending on the acquisition technology and the size of the scene. This creates difficulties in the manipulation of the data, given that a point cloud usually consists of a vast amount of points with coordinates in floating-point format, which results in excessive storage requirements.

A simple, yet efficient way to reduce the size and ease the manipulation processes, is to convert the original model to a regular data structure. Obviously, such an operation leads to information loss in the general case; however, the error is bounded and the effects can be mitigated by increasing the resolution of the regular representation.

B.1 Octree structure

Octree data structures are extensively employed in point cloud compression, as they enable an efficient way for a regular representation of the model's geometry. To generate an octree, the minimum bounding cube enclosing the point cloud is initially computed. A recursive partition is applied on the 3D space defined by the bounding cube until a desired level-of-detail (LoD), or a tree-depth is achieved. The former specifies the size of the leaf nodes, whereas the latter defines the number of partitions that are recursively applied on the octree data structure. At

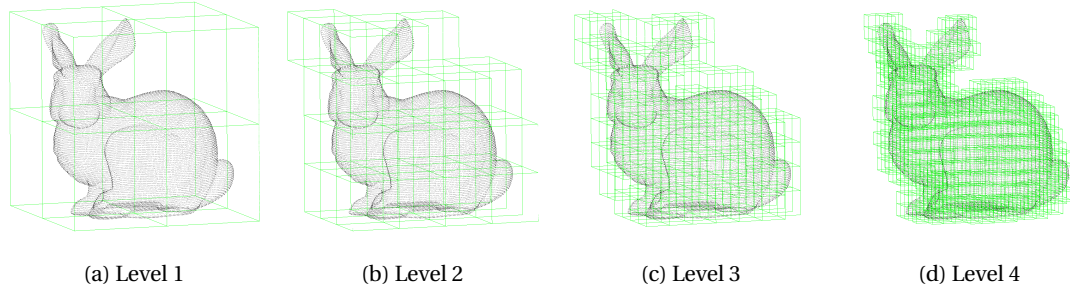


Figure B.1 – Octree data structure decomposition.

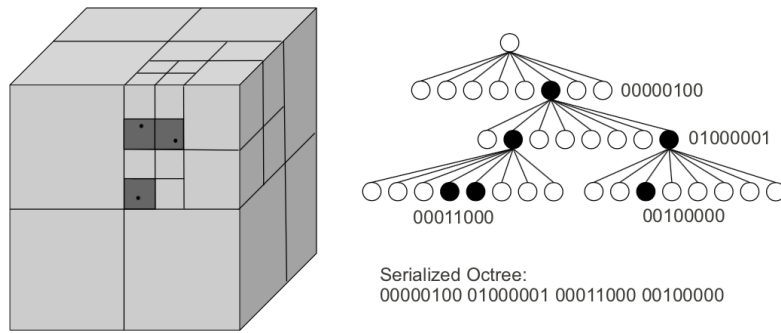


Figure B.2 – Octree-based compression. Illustration from (Kammerl et al., 2012).

each level, every bounding cube is sub-divided into 8 smaller and equally sized sub-cubes as illustrated in Figure B.1. This procedure is repeated until the specified LoD or tree-depth is reached. Then, all the points that are enclosed in a leaf node are collectively represented by the center of that node.

Octree decomposition can be used to efficiently encode point cloud data. In the simplest case of octree-based compression, a byte can be used to represent the occupancy of a branch node, provided a fixed ordering for its eight children. Then, by traversing the tree and storing the occupancy maps of the children at each level, the topology of a point cloud is encoded, as depicted in Figure B.2.

B.2 Voxel grids

Voxelization is a commonly used approach to convert irregular point clouds to regular data structures. A voxel, v , can be defined as a sample in a regularly spaced 3D grid. It consists of a volumetric element of size 1, $[-0.5 + i, 0.5 + i) \times [-0.5 + j, 0.5 + j) \times [-0.5 + k, 0.5 + k)$, which is represented by the center of the voxel with coordinates $(i, j, k) \in [0, 2^{N-1}]^3$, where N is the voxel bit depth. Voxelization can be defined as the process of mapping the coordinates of each

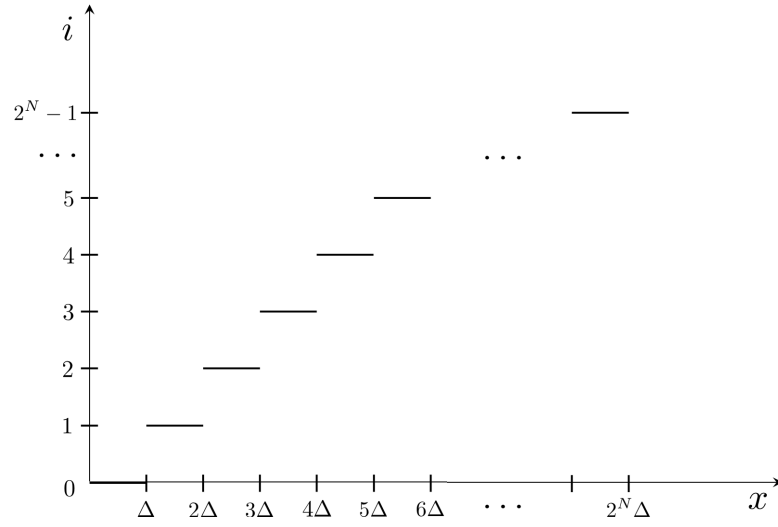


Figure B.3 – Quantization steps.

point spanning in a continuous space, $p \in \mathbb{R}^3$, to a discrete set of values, $v \in \mathbb{Z}_{\geq 0}^3$. This is very similar to quantization; however, in the case of voxelization the reconstruction step is typically skipped. Moreover, duplicated entries of voxel centers are typically discarded. Information from additional attributes (e.g. color) present in the original model is associated to the voxels. This information is obtained after attribute-dependent processing.

Provided that the geometry of a point cloud can have an arbitrary span, it is rather common to apply normalization prior to voxelization. Let us assume that the coordinates of a point p in the original point cloud are expressed as $(\hat{x}, \hat{y}, \hat{z})$, while after normalization are denoted as (x, y, z) , with the model placed in a bounding box of range $[0, 1]^3$. This can be performed through the following homogeneous transformation:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} s & 0 & 0 & t_x \\ 0 & s & 0 & t_y \\ 0 & 0 & s & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ 1 \end{bmatrix} \quad (\text{B.1})$$

where $s = (\max[\hat{x}, \hat{y}, \hat{z}] - \min[\hat{x}, \hat{y}, \hat{z}])^{-1}$, and $t_x = -\min \hat{x}$, $t_y = -\min \hat{y}$, $t_z = -\min \hat{z}$. The normalized geometry is then quantized. Considering the simplest case, a value i_n is obtained from forward quantization of the coordinate x_n , as: $i_n = \lfloor x_n / \Delta \rfloor$

The value i_n is an index to the interval that x_n is falling across the corresponding axis x . The boundaries of an interval are called decision levels. Every x_n that is falling within the same boundaries is represented by the same index (quantization level). The length of the interval equals the quantization step, Δ , which depends on the number of quantization levels that are required. Assuming a number of 2^N quantization levels, as shown in Figure B.3, then $\Delta = 2^{-N}$, with N indicating the number of bits that are required to represent the entire set of

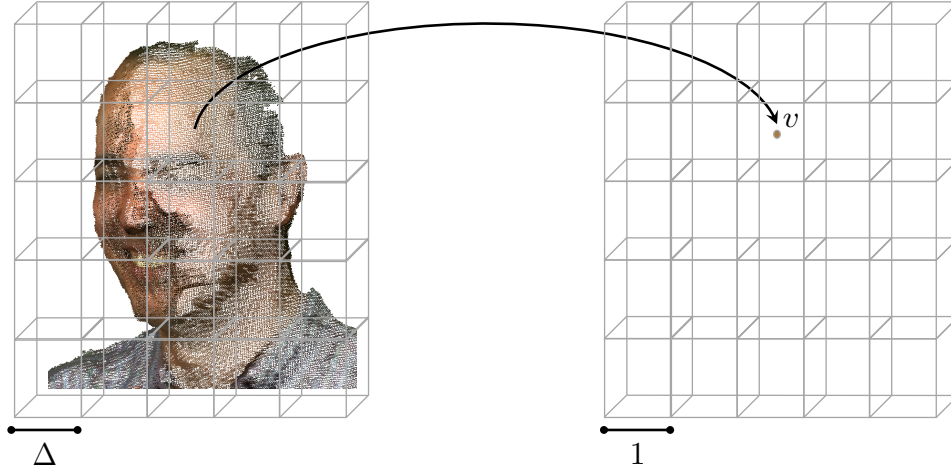


Figure B.4 – Voxelization.

quantization levels i .

Note that the voxel v_n corresponds to a cubic sub-space contained in the original point cloud with length of edge equal to the quantization step Δ . See Figure B.4.

Naturally, a voxel exists or not, depending on whether a point lies in the corresponding sub-space of the original model or not. It is possible that several points are falling in the same sub-space, thus, the same voxel is registered several times. Considering the geometry of a model, registering the same entry of the voxel grid multiple times is redundant. However, this is not the case when it comes to other attributes. In particular for color, in case a given voxel represents a single point, the color of the latter is assigned to the former. In case a given voxel represents more than one points, color blending (i.e. averaging), or sampling can be employed, to obtain the color of the voxel.

Summarizing, voxelization results in a lattice grid of volumetric elements of size 1 represented by their centers, and approximates a regularly spaced sub-sampled version of the original point cloud. The positions of the voxels are typically obtained through quantization of geometry and after duplication removal, while different attribute-dependent approaches can be used to assign attribute information to the voxels that is present in the original model.

Finally, it is noteworthy that an octree structure can be employed in order to represent, or encode a voxel grid. Provided a proper LoD, or tree-depth for the former, the voxel grid can be represented without any loss.

B.2.1 Implementations

For purposes of experimentation, different approaches for point cloud voxelization have been implemented. For geometry, two different alternatives have been realized: (a) a mid-treat,

and (b) a mid-riser quantizer. Note that this naming is used by convention as it resembles, but doesn't accurately reflect the implementation of the corresponding quantization schemes. The mid-treat quantizer, is implemented through Equation B.2, while the mid-riser quantizer is given by Equation B.3:

$$i = \lfloor x/\Delta + 1/2 \rfloor \quad (\text{B.2})$$

$$i = \lfloor x/\Delta \rfloor \quad (\text{B.3})$$

If the original point cloud contains color information, and each voxel corresponds to a single point, the color value of the latter is attributed to the former. In case a voxel corresponds to more than one points, three different alternatives have been examined: (a) random color sampling, by randomly picking a color value from the points that correspond to the same voxel, (b) first-sorted color sampling, by selecting the color value of the first among the sorted points that correspond to the same voxel, and (c) color blending, by averaging the color values of the points that corresponds to the same voxel. The latter approach can be viewed as a low-pass filtering and leads to color smoothing with visually more pleasing results, as can be seen in Figure B.5. Note that the original model was represented by 106,199,111 whereas the voxelized point cloud (at 10-bit resolution) is comprised of 1,081,025 points.



(a) Original model



(b) First-sorted



(c) Randomly-picked



(d) Blended

Figure B.5 – Illustration of visual effects for different color mapping techniques in voxelization.

C Accuracy of normal estimation algorithms

Normal vectors in point cloud imaging are crucial for a number of use-cases including rendering, surface reconstruction, segmentation, and feature extraction, among others. In essence, these attributes indicate the shape of the 3D model, which is represented through point samples. Normal vectors are not natively exported during point cloud acquisition in most cases, and when this happens, it is not necessary that they accurately reflect the underlying surfaces. Thus, it is rather common to (re-)estimate them from a point cloud model in an off-line, post-processing stage, where various normal estimation algorithms and configurations might be tested.

In principle, the problem of normal estimation on point clouds has been extensively studied from different communities, such as computer graphics, signal processing and mathematics, while lately, deep-learning solutions have been also proposed. In this context, a point cloud is interpreted as a collection of discrete samples that are drawn from continuous surfaces, and the objective is to infer the underlying shape from this set of unorganized points. Normal estimation can be considered as an ill-posed problem, in the sense that there is no unique solution for a given topology. Moreover, the performance of normal estimation algorithms is highly affected by the density of samples and surface curvature irregularities across a model, as well as the presence of noise that might be introduced during acquisition, or other processing stages (e.g., compression) (Mitra and Nguyen, 2003). The accuracy, robustness and efficiency of widely-used normal estimation algorithms has been evaluated and reported by a series of studies in the literature (Klasing et al., 2009; Jordan and Mordohai, 2014).

In this study, the objective is to evaluate the accuracy of normal estimation schemes that are extensively exploited in our work, in order to obtain insights regarding their operating point. In particular, we choose 3 widely-used algorithms as implemented by well-established software and we test several neighborhood sizes to estimate the normal vector of a queried point: (a) plane fitting using k -nearest neighbors, (b) plane fitting using range search, and (c) quadric fitting using range search. The ground-truth normal vectors are generated by sub-sampling a set of reference mesh models. Then, normal estimation is applied on the sampled geometry for a given algorithm and configuration. Lastly, the average angular error

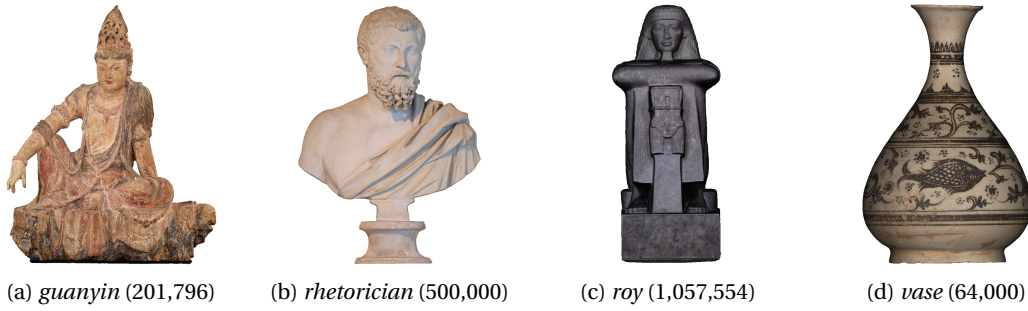


Figure C.1 – Reference mesh contents (in parenthesis the number of faces).

is computed between the estimated and the ground-truth vectors, in order to quantify the performance of the corresponding testing condition.

C.1 Data set

We pick 4 mesh models with different shapes and curvature distributions from the references of the *PointXR dataset* (Alexiou and Ebrahimi, 2020). The models can be seen in Figure C.1, namely, *guanyin*, *roy*, *rhetorician*, and *vase*, from left to right.

C.2 Computation of normal vectors

C.2.1 Ground-truth normal vectors

For every content we repeat the following procedure:

1. Load the mesh in CloudCompare and use a build-in functionality to estimate the normal vectors per vertex i.e., the normal at a vertex is obtained as the mean normal vector across all the triangles (i.e., faces) connected to this vertex.
2. Generate a point cloud by sub-sampling at different sparsity levels, using a target number of output points; in this study, we use 250K, 500K, 1M and 2M. Each point cloud is obtained by drawing a constant number of samples at random positions, from each triangle of the original mesh. The normal vector of a point sample is computed via spatial interpolation applied on the normal vectors of the vertices that surround it in the original mesh.
3. Remove neighboring coordinates of the sampled geometry with identical normal vectors.
4. Scale geometry in the range $[0, 1]$.

The point cloud models carrying the ground-truth normal vectors are obtained from the last step.

C.2.2 Estimated normal vectors

Normal vectors are estimated based on the point coordinates of the models that carry the ground-truth normals, which were generated previously. The algorithms and configurations that were selected for evaluation purposes, are summarized below.

Normal estimation methods are typically split in two steps: (a) identification of a neighborhood, and (b) fitting of a curve. For the former, there are two main approaches, namely, the k -nearest neighbors and the range search using a fixed radius R . For the latter, different order polynomials are employed with linear and quadratic being the most common. Notice that there are no indications in the literature which algorithm provides more accurate results. In previous studies, one method is typically selected, and different configurations are evaluated (Klasing et al., 2009; Jordan and Mordohai, 2014). In this analysis, we recruit several algorithms as implemented from different software.

Plane fitting using k -nearest neighbors For this algorithm, the classic approach described in (Hoppe et al., 1992) is employed. In particular, the k -nearest neighbors (k -nn) are used to identify a local region around a queried point, and in every neighborhood a planar surface is fitted. In this experiment, we set $k = \{8, 16, 32, 64, 128, 256, 512\}$, and the Meshlab (Cignoni et al., 2008) built-in implementation is used.

Plane fitting using range search This algorithm is identical to the one described earlier, with the only difference that range search using a radius R is employed for neighborhood identification. In this experiment, we use R from 0.001 to 0.01 with a step of 0.001 (each model's geometry is limited to a bounding box of size 1). Moreover, the CloudCompare built-in function is used. This implementation is fast, while also it tackles a typical bottleneck of this algorithm arising when no neighboring points are identified at a specific radius. In the latter case, the search range progressively increases up until a minimum number of neighbors is reached; thus, normal vectors are estimated for all point samples.

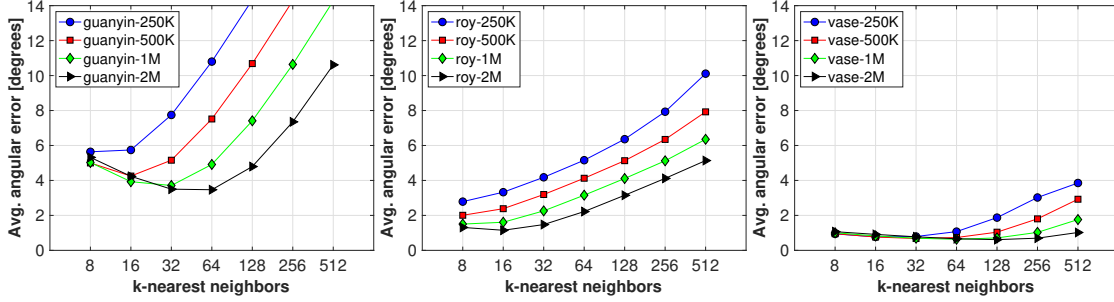
Quadric fitting using range search This algorithm is based on fitting a quadratic polynomial in a neighborhood defined by range search. Again, the radius R is spanning from 0.001 to 0.01 with a step of 0.001, and the CloudCompare built-in function is used. As mentioned earlier, there are several advantages that are offered from this implementation.

C.3 Performance evaluation

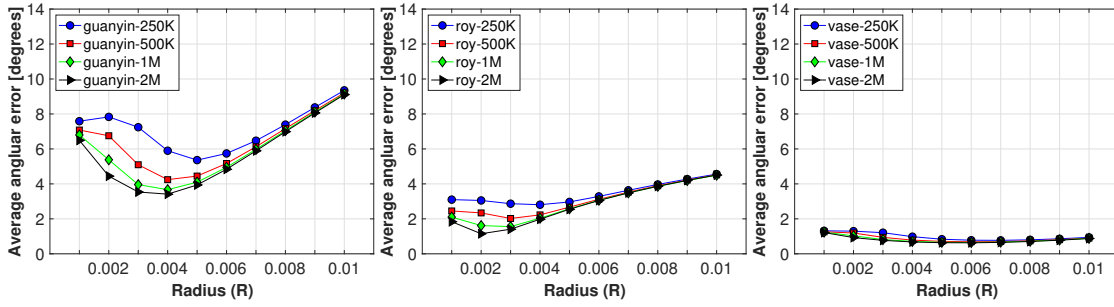
After obtaining both the ground-truth and the estimated normal attributes, the average angular error (per point) is computed across a model. The angular error is measured in degrees. Note

Annex C. Accuracy of normal estimation algorithms

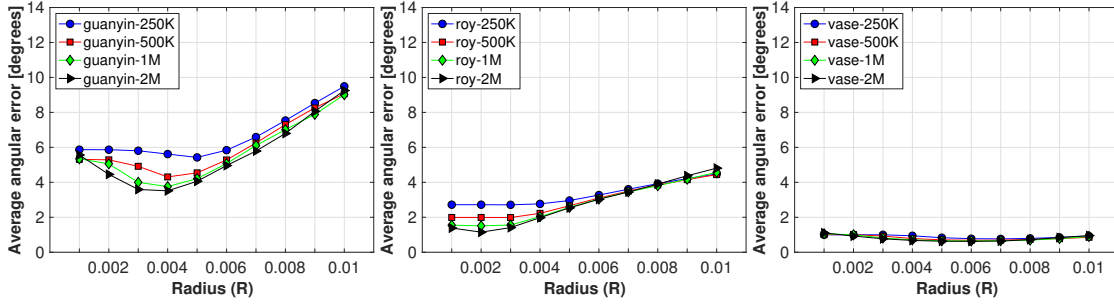
that in our computations we do not consider the orientation of the vectors. This is due to the fact that wrongly oriented normal vectors (flipped) are rather common after normal estimation. To address this issue, several algorithms have been proposed (e.g., Minimum Spanning Tree), however, their performance and impact is outside of our scope.



(a) k -nn with plane fitting



(b) Range search with plane fitting



(c) Range search with quadric fitting

Figure C.2 – Average normal estimation error in degrees. A different normal estimation algorithm is displayed in each row, while the same model is presented across a column.

C.4 Results

In Figure C.2, the average angular error per point is presented and expressed in degrees, in order to indicate the error between the estimated and ground-truth unoriented normals. Note that all the models are displayed excluding *rheticorian*, which was found to undergo a very similar behavior to *guanyin*. In each column a different model is depicted, while in each row a

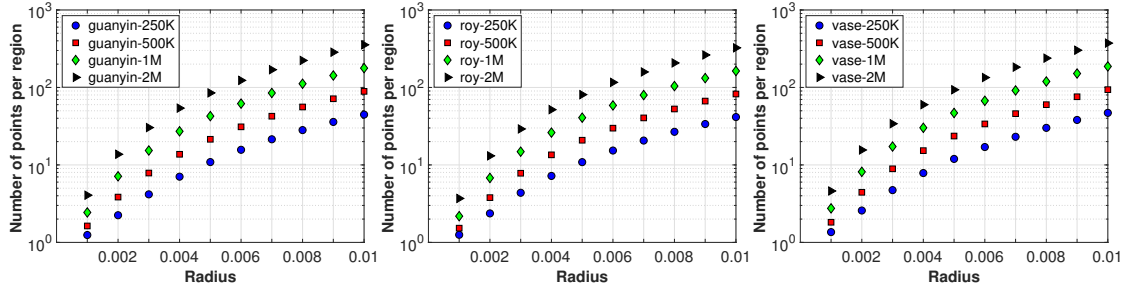


Figure C.3 – Average number of points per neighborhood, for *guanyin*, *roy* and *vase*.

different normal estimation algorithm is arranged. In every figure, each curve corresponds to a certain variation of a model (i.e., sparsity level) and reports the angular error at different neighborhood sizes.

As could be expected, the normal estimation error depends on the algorithm as well as the model's topology and sparsity. A general observation is that the normal estimation error is constantly lower at denser models. Moreover, it is evident that the performance of the plane and quadric fitting using range search is similar, with the latter showing improvements at low-radii, especially for sparser versions and more complex geometries (e.g., *guanyin* at 250K). Moreover, it seems that the error trends for the plane fitting using k -nn behave differently with respect to range search-based variants.

Using the k -nn approach, the minimum error is achieved at larger k 's for denser models, whereas for sparser models, the minimum is achieved at smaller neighborhoods; this is clearer for point clouds with more complex topology (i.e., *guanyin* and *rhetorician*). Notice that approximately the same angular error is obtained when simultaneously doubling the target number of points of a model (i.e., sparsity level) and the number of neighbors over which a neighborhood is identified. This is reasonable if we consider that in both cases, the region over which the plane is fitted covers approximately the same volume. On the contrary, using the same k as the sparsity level is increasing, indicates that larger volumes are employed.

When using range search, the error is generally decreasing as the radius is increasing up to a threshold. Moreover, the minimum error is achieved at adjacent low- to mid-range R values for all versions of a content, while at high-radii the error convergences and increases at the same pace independently of the sparsity level. These phenomena are related to the fact that using the same radius, the same space is employed to fit a curve.

An interesting outcome is that approximately the same minimum angular error is achieved per model, across all tested algorithms. To obtain a better view, in Figure C.3 the average number of points of a neighborhood for the selected radii in range search is reported. Combining this plot with the results shown in Figure C.2, we can extract that this minimum angular error is achieved over a similar neighborhood population, when compared to the number of neighbors of the k -nn approach for which the minimum error is reported. In general, it can be

Annex C. Accuracy of normal estimation algorithms

seen that using an R that leads to an average number of points that approximates the number of neighbors in k -nn, a similar error is obtained.

These results indicate that the normal estimation error is not substantially affected by the fitted curve or the working principle of the neighborhood identification approach (i.e., k -nn, or range search). Rather, it is the configuration of the algorithm that is critical; that is, the volumes over which the surfaces are approximated. These configurations should be adjusted considering the intrinsic characteristics of a model, in order to grant high accuracy.

Finally, it needs to be emphasized that these conclusions reflect the outcome of our experimentation using a limited set of models; thus, although indicate reasonable trends, further investigation is needed to draw safe conclusions. It should be clear that the scope of this study is not to benchmark the selected algorithms, rather to understand whether they are able to accurately estimate the normal vectors from a set of points, and to identify their operating point.

D Renderers

Nowadays, point clouds are most commonly consumed through devices with conventional, flat screens (i.e., phones, laptops, desktops) or through HMDs (i.e., HTC VIVE, Faro, Bridge) by means of software for 3D computer graphics (i.e., OpenGL, Direct3D, WebGL). Some of the most popular libraries for point cloud processing and rendering are Point Cloud Library (PCL) (Rusu and Cousins, 2011)¹, CloudCompare (CloudCompare, 2020)², MeshLab (Cignoni et al., 2008)³. These frameworks are able to handle and render a vast amount of data, providing robust and high-performance solutions. However, they support only fixed point size rendering, while the point sizes don't adjust with virtual camera distance changes, which is more evident at closer views (zoom in). CGAL⁴ denotes a classical suite for computational geometry algorithms. Recent libraries such as Open3D⁵, polyscope⁶ and the "Point cloud visualizer" add-on in Blender⁷, have also received attention. Among the web-based alternatives is Potree⁸, which is suitable for large point clouds, offering a wide range of features and fast interactivity and Meshlab JS⁹. Finally, Unity¹⁰ and Unreal¹¹, denote popular game-engines for XR experiences.

The above solutions indicate the large pool of options that are available today. In this section we report custom rendering implementations that were developed for our experimentation purposes, making use some of these frameworks.

¹<http://pointclouds.org/>

²<https://www.cloudcompare.org/>

³<http://www.meshlab.net/>

⁴<https://www.cgal.org/>

⁵<https://github.com/intel-isl/Open3D>

⁶<https://github.com/nmwsharp/polyscope>

⁷<https://github.com/uhlik/bpy>

⁸<https://github.com/potree/potree>

⁹<http://www.meshlabjs.net/>

¹⁰<https://unity3d.com/>

¹¹<https://www.unrealengine.com/>

D.1 Voxel-based renderer

The voxel-based rendering software is developed in C++ in the context of (Torlig et al., 2018a). In this implementation, the rendering-related computations are performed in run-time. Initially, each point cloud is read point by point. The range across each axis is recorded and the average position of all the points (i.e., the centroid) is kept. By subtracting the centroid from the coordinate data, the 3D model is effectively centered in the viewing volume.

In the next step, a point cloud is scaled by multiplying the coordinates of each point by the largest power of two that would still let the model fit in a cubic volume of side 1024. In other terms, the spatial coordinates are multiplied by a scaling component, s , given by Equation D.1

$$s = z_f \times 2^{(10 - \lceil \log_2(w_{max} - w_{min}) \rceil)} \quad (D.1)$$

where z_f is a zoom factor that is inversely proportional to the virtual distance between the user and the content, and is updated based on the user's scrolling of the mouse wheel. The terms w_{max} and w_{min} correspond to the largest and smallest coordinate values of the point cloud. Essentially, this procedure scales the content appropriately as a function of the current virtual distance, and prepares the content to be projected in a pixel grid of a selected resolution.

Subsequently, the points go through a rigid rotation, as a function of the viewing angle. This is done by multiplying the spatial coordinates with a rotation matrix. This rotation matrix is calculated using angles in two axes, which are determined dynamically by the user through clicking and dragging using the right mouse button across the screen in the X and Y directions. Rotations in these angles are equivalent to incremental changes to the yaw and pitch of the rendered model, respectively. Note that each stimulus is rotated identically in order to attain in-sync visualization from the same viewpoint, in case more than one models are displayed simultaneously.

In a following step, the spatial coordinates of the points obtained from the aforementioned transformations are traversed and quantized to integer values. During this iteration, the spatial position of each point might be modified by updating the x and y coordinates, in case translation has been issues by the user's commands. In particular, this translation, or panning is determined through dragging the mouse while holding the left mouse button. Again, identical panning is applied on the views of each displayed model.

The color value of every point with spatial coordinates (x, y, z) , as resulted from the procedure described above, is associated with a single image pixel (\hat{x}, \hat{y}) in the respective projected image. During the iteration, if another point with identical (x, y) coordinates and a smaller distance from the projection plane is identified, the first point is ignored, and the color value of the pixel (\hat{x}, \hat{y}) is given by the second point. In the special case where multiple points have coinciding coordinates after rotation, quantization and panning, the associated pixel value is derived as the average of the color values of the points. This procedure is repeated for every point of both the reference and the distorted point clouds, providing content projections that are finally

rendered to the viewer. Unoccupied pixels in the rendered images are given a default value of (127, 127, 127) in the RGB color space, which corresponds to neutral gray.

A screen-shot of the renderer as part of the subjective evaluation testbed is illustrated in Figure 4.12. Visual examples of models under compression artifacts are provided in Figures 4.14 and 4.15.

D.2 Splat-based renderer in VTK

In this renderer, each point is replaced by a splat, which is represented by a primitive geometrical 2D or 3D object from the built-in options provided in the VTK¹² library. A geometrical object is defined by a set of vertices and corresponding connectivity information. Thus, in this implementation, a point cloud is essentially converted to a format with mesh-like properties, in a pre-processing stage. During run-time, the latter is loaded into the visualizer to represent the point cloud content.

At the pre-processing stage, we first define and construct the primitive geometrical objects. The source elements that are natively integrated in our solution are disks, cubes and spheres; however, extensions to other objects are straightforward. Implementation-wise, for cubes and spheres, the `vtkGlyph3D` filter with the `vtkCubeSource` and the `vtkSphereSource` are employed, respectively. The latter demands a number of vertices across the spherical coordinates ϕ and θ , which are both set equal to 7 in our case. For disks, the `vtkTensorGlyph` filter is employed using the `vtkRegularPolygonSource` with 16 sides in order to obtain a good approximation of a disk. The default radius (i.e., size) of every source element is adjusted at will.

After establishing the geometrical shape, the position and the color of the splats are determined by the corresponding points' coordinates and color values. Regarding the splat orientation when using disks, it is defined as perpendicular to the direction of the normal vector associated with the corresponding point sample. Thus, in this case, the latter attributes should be associated with the coordinate data. Note that visible artifacts might be observed in the form of holes, when the normal vectors do not accurately reflect the underlying surface (i.e., mis-oriented). For cubes and spheres, the orientation is fixed across the z -axis. Each splat is oriented towards this particular direction in the world coordinate system the model is lying. In particular, when manipulating a displayed model (e.g., rotation, translation, etc.), the camera position and direction is correspondingly adjusted, while the splat orientation remains identical within the coordinate space. This approach may lead to the perception of different splat sizes, as the camera is rotating. For instance, when the camera is perpendicular to the frontal face of a cube, the same point will be projected on a smaller area onto the screen with respect to the case the camera is aligned with the space diagonal of the cube. This is true for disks and cubes, whereas such effects are not observed for spheres, due to the nature of this geometrical shape.

¹²<https://vtk.org/>

The renderer supports both fixed and adaptive splat sizes. In the first case, the region over which every splat of a model is extended is constant, and can be manually specified by the user. In the latter case, the size of the splats is adaptively scaled based on the sparsity of local neighborhoods across a model. The first approach eliminates magnification of sparser regions and is better suited for point clouds with regular geometry and uniform point density distribution. Yet, in the second approach, the visual quality is assumed to be better for models with fluctuating sparsity levels, as the splat sizes can be correspondingly adjusted. To enable the latter option in our software, during pre-processing, all points of a model are traversed and the distances to their k nearest neighbors are identified. Then, for every point p , the corresponding splat size s_p is set equal the local mean distance μ_p that is computed considering its k nearest neighbors. To avoid amplification of the splat size of isolated points that deviate from surfaces (e.g., acquisition errors), we assume that μ_p is a random variable that follows a Gaussian distribution $N(\mu, \sigma)$, and every point p with local mean outside of a specified range, is classified as an outlier. In our case, this range is defined by the global mean $\mu = \bar{\mu}_p$ and standard deviation $\sigma = \bar{\sigma}_p$. For every point p , if $\mu_p \geq \mu + 3 \cdot \sigma$, or $\mu_p \leq \mu - 3 \cdot \sigma$, then p is considered an outlier, and s_p is set equal to the global mean μ . Otherwise, the splat size s_p is set equal to the local mean μ_p .

A screen-shot of the renderer as part of the subjective evaluation testbed is illustrated in Figure 4.11. Visual examples of models under different splat shapes, are provided in Figure 4.6. Visual examples of models under compression artifacts using cubic splats are provided in Figures 4.14 and 4.15.

D.3 Splat-based renderer in JS

An interactive renderer has been developed in a web application on top of the Three.js¹³ library. The software supports point cloud data stored in both PLY and PCD formats, which are displayed using square primitive elements (splats) of either fixed or adaptive sizes. The primitives are always perpendicular to the camera view direction by default, thus, the rendering scheme is independent of any information other than the coordinates, the color and the size of the points. Note that the latter type of information is not always provided by popular point cloud formats, thus, there is a necessity for additional metadata for our renderer (see below).

To develop an interactive 3D rendering platform in Three.js, the following components are essential: a camera with trackball control, a virtual scene, and a renderer with an associated canvas. A virtual scene is initialized and a point cloud model is placed in the middle. The color of the scene can be adjusted to serve the purpose of a user. To capture the scene, an orthographic camera is employed, whose field of view is defined by setting the camera frustum. The users are able to control the camera position and zoom through mouse movements, handling their viewpoint; thus, interactivity is enabled. A `WebGLRenderer` object is used to draw the current view from the current camera position onto a canvas. The dimensions of the

¹³<https://threejs.org/>

canvas can be manually specified. It is worth mentioning that the update rate of the trackball control and the canvas is handled by the `requestAnimationFrame()` method, ensuring fast response (i.e., 60 fps) in high-end devices.

After a point cloud has been loaded into the scene, its shape is scaled according to the camera's frustum dimensions to be visualized in its entirety. The selected view of the model is projected onto the canvas, with each point being mapped to a single pixel. To enable visualization of watertight surfaces from different viewpoints, each point is represented by a square splat mapped to a corresponding number of pixels on the canvas. The size of the splat is adjusted according to the model's intrinsic resolution and camera's position. For the latter, given the orthographic camera, the splat size is adjusted based on the camera zoom parameter; thus, the size is increasing or decreasing, depending on whether the model is visualized from a close or a far distance. For the former, metadata is loaded by the application, carrying the information for the size of each point, or the entire model, depending on whether an adaptive or a fixed rendering approach is adopted. Besides the point size, an additional factor is provided that might be used to scale the splats of a model. This constant may be interpreted as a global compensating quantity that can be adjusted depending on the sparsity of the model, for visually pleasing results. Although such calculations can be performed in real-time, using off-line generated metadata has the advantage of reducing the computational overhead of the rendering software.

To enable fixed point size rendering, a single value is used to scale the splats of a model. This is achieved by multiplying the default value of the size in the class `material`, which is responsible for the appearance of a point. For adaptive splat rendering, a custom WebGL shader/fragment program was developed, allowing access to the attributes and adjustments of the size of each point individually. In particular, a new `BufferGeometry` object is initialized adding as attributes the points' position, color and size; the former two can be directly retrieved from the content. A new `Points` object is then instantiated using the object's structure, as defined in `BufferGeometry`, and the object's material, as defined using the shader function. The information about the size of the entire set of points, or each point individually, is stored in form of metadata in a JSON file, for fixed or adaptive rendering respectively. In the first case, a single value is required, whereas in the second case, a value per point is stored following the same order as the points that belong to the model.

Auxiliary functionalities and tools that were integrated in the developed software and can be optionally enabled, consist of recording user's interactivity information and allowing taking screen-shots of the rendered models.

A screen-shot of the renderer as part of the subjective evaluation testbed is illustrated in Figure 9.6. Visual examples of models under compression artifacts are provided in Figure 9.5. The software has been uploaded and released on GitHub (see annex E).

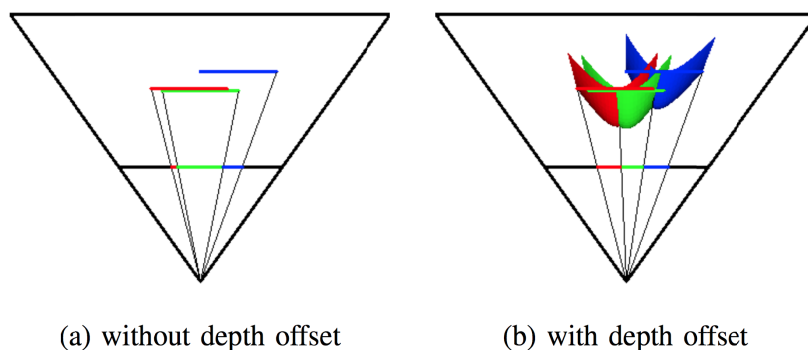


Figure D.1 – Illustration of shader interpolation. Figure from (Schütz and Wimmer, 2015).

D.4 Splat-based renderer in Unity: PointXR toolbox

The Unity platform is used to develop our toolbox. The Pcx Point Cloud Importer¹⁴ is a software dependency responsible for loading the point clouds and converting them to objects with mesh-like properties. By default, a point cloud is rendered either by using raw points, or by associating disks of fixed size across an entire model through a dedicated shader script. The size of the disks can be manually configured at per user's preferences and is automatically adjusted to the virtual camera distance.

Implementation-wise, our application enhances the natively supported rendering capabilities by integrating the following features: (a) Quad shader, which is less computationally expensive as it requires a smaller number of vertices to represent a point. At the same time, no added visual distortions are noticeable, especially when is combined with the shader interpolation rendering mode (see further). (b) Adaptive point size, which can be beneficial with models of irregular structure and varying point density, as it allows the regulation of the size of each point individually. The deployed algorithm is inspired by (Javaheri et al., 2017b; Alexiou and Ebrahimi, 2019), with the size of each point depending on the k -nearest neighbors' distances as implemented by the KDTree¹⁵ into Unity. In fact, the same approach described in section D.3 is employed, assuming that the average distance of a point from its neighbors is a random variable that follows a Gaussian distribution. If the this distance exceeds a threshold, then the global mean is employed, thus, magnification of outlier points is avoided. The number k can be manually specified. (c) Shader interpolation (Schütz and Wimmer, 2015), which is integrated into the rendering pipeline (i.e., shader scripts) for higher visual quality. In this mode, surface discontinuities are reduced, and flickering artifacts perceived due to changes of viewing position are decreased. In particular, a depth offset is added on the view space to each pixel of the primitive element that represents a point, and pixels with lower depth values (i.e., frontal parts of the primitives) are shown on the screen. This essentially leads to a screen faced paraboloid, as illustrated in Figure D.1. Thus, larger point sizes can be used to avoid

¹⁴<https://github.com/keijiro/Pcx>

¹⁵<https://github.com/viliwonka/KDTree>

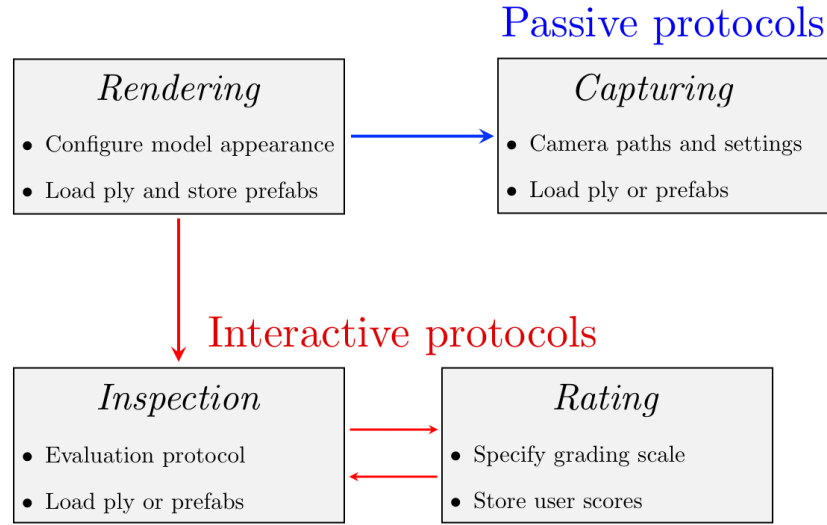


Figure D.2 – Color coded block diagram with scene dependencies to enable corresponding evaluation protocols.

hollows, leading to the perception of higher quality and smoother surfaces.

Our toolbox consists of four Unity scenes, namely, *Rendering*, *Inspection*, *Rating*, and *Capturing*. In Figure D.2, a block diagram is provided illustrating their main functionalities and dependencies. The *Rendering* scene can be used to configure the visual appearance of a model. Through a GUI, fixed or adaptive point size can be set and related parameters (i.e., point size in the former and number of nearest neighbors in the latter) can be specified, the shader interpolation mode can be enabled or disabled, and a shader of preference can be submitted. Moreover, a point scaling factor that is globally applied on the entire model can be manually adjusted for higher fidelity. Furthermore, the position, the rotation, and the size of a model can be specified at will. The user can apply the selected rendering features and visualize the resulting model in real-time either in a typical monitor, or through a headset in VR mode. Upon saving, a pre-fabricated (prefab) object, corresponding assets, and a configuration file with the selected rendering options are generated. The latter file can be optionally used as an input to the scene, in order to automatically apply the corresponding rendering configurations. An example of the *Rendering* scene is illustrated in Figure D.3.

The *Inspection* scene, when used as a standalone application, consists of the default Unity viewer where point clouds and prefabs can be loaded. Yet, when used in combination with the *Rating* scene, they establish the interactive quality assessment testbed. In particular, in the *Inspection* scene the experimenter can specify whether to use a single-stimulus or a double-stimulus visualization protocol. The latter option currently provides three variants: (a) simultaneous, where the models are presented side-by-side, (b) sequential, where one model is presented after the other, and (c) alternating, where the subject is able to switch between the two models at will. Moreover, the experimenter can choose to log interactivity information



Figure D.3 – Example of the PointXR toolbox scene for adjusting the rendering configurations of a model.

of a subject during evaluation. On the other hand, the *Rating* scene is responsible to capture the scores of a subject. In particular, a grading panel with a question and a list of scores in the form of buttons appears in front of the observer, upon request (see Figure 5.3c). Note that the question and the answers can be manually specified, thus covering a wide range of subjective evaluation methodologies when combined with the visualization protocols from the *Inspection* scene.

Finally, the *Capturing* scene is part of our toolbox to carry passive evaluation experiments. The user can define a set of camera parameters (i.e., position, rotation) and corresponding time intervals that will be used to capture views of a model as rendered in our virtual environment. Simple camera paths can be enabled through a GUI, such as circular or spiral rotations with adjustable angular speed. The manual selection of camera settings (i.e., position, rotation, time interval) is another option. For higher precision and control, though, a configuration file can be loaded to explicitly specify the camera settings at every time instance. To produce video sequences from the selected viewpoints, our scene makes use of the Unity Recorder tool.

A screen-shot indicating a model rendered in the virtual world is illustrated in Figure 5.2. Visual examples of a user interacting in the *Inspection* and the *Rating* scene are provided in Figure 5.3. The software has been uploaded and released on GitHub (see annex E).

E Open access material

Below is a list of open access contributions to to facilitate and promote research on the field.

Data sets

- **G-PCD:** Data set that contains reference point cloud models, degraded stimuli, and subjective quality scores. The generation of the degraded stimuli is described in section 3.1. The subjective scores are obtained from the experiments described in sections 3.2 and 3.3. This work has been published in (Alexiou and Ebrahimi, 2017b) and (Alexiou et al., 2017). The URL link is:
<https://www.epfl.ch/labs/mmspg/geometry-point-cloud-dataset/>.
- **RG-PCD:** Data set that contains reference point cloud models, degraded stimuli, corresponding reconstructed meshes, and subjective quality scores. The generation of the degraded stimuli is described in section 3.1. The subjective scores are obtained from the experiment described in section 3.4, where the reconstruction methodology is also described. This work has been published in (Alexiou et al., 2018). The URL link is:
<https://www.epfl.ch/labs/mmspg/reconstructed-point-clouds-results/>.
- **ViAtPCVR:** Data set that contains tracked behavioral data, post-processing results, saliency maps in form of importance weights, re-distribution of a sub-set of contents and scripts to prepare the stimuli of the study. The experiment and the data processing is described in section 5.2. This work has been published in (Alexiou et al., 2019b). The URL link is:
<https://www.epfl.ch/labs/mmspg/visual-attention-point-clouds/>.
- **M-PCCD:** Data set that contains subjective scores, instructions to retrieve models and scripts to prepare the stimuli of the study. The generation of the point cloud stimuli is described in section 9.1. The subjective scores are obtained from the experiment described in section 9.2. This work has been published in (Alexiou et al., 2019a). The

URL link is:

<https://www.epfl.ch/labs/mmosp/quality-assessment-for-point-cloud-compression/>.

- **PointXR:** The material is organized as follows:
 - *PointXR dataset:* Repository consisting of 20 high-quality point clouds representing cultural heritage, generated from publicly available mesh models. The point clouds were generated after texel sampling applied on the mesh models in Meshlab using a texture resolution of 4096x4096 (WxH).
 - *PointXR experimental data:* Data set that contains reference point cloud models, degraded stimuli, and subjective quality scores. The subjective scores are obtained from the experiment described in section 5.1, where the generation of the degraded stimuli is described.

This work has been published in (Alexiou et al., 2020b). The URL link is:

<https://www.epfl.ch/labs/mmosp/downloads/pointxr/>.

Software

- **Point cloud angular similarity (plane-to-plane):** Prototype MATLAB implementation of the angular similarity metric (also called as plane-to-plane). The quality metric is described in section 6.1. This work has been published in (Alexiou and Ebrahimi, 2018c). The URL link is:
<https://github.com/mmosp/point-cloud-angular-similarity-metric>.
- **Point cloud structural similarity (PointSSIM):** Prototype MATLAB implementation for the computation of structural similarity scores. The quality metric is described in section 6.2. This work has been published in (Alexiou and Ebrahimi, 2020). The URL link is:
<https://github.com/mmosp/pointssim>.
- **Point cloud web renderer:** Open source web-based point cloud renderer in Three.js library. Implementation details are provided in section D.3. The renderer is used in the context of the experiments described in chapter 9. This work has been published in (Alexiou et al., 2019a). The URL link is:
<https://github.com/mmosp/point-cloud-web-renderer>.
- **PointXR toolbox:** Unity implementation for rendering and visualization of 3D point clouds in virtual environments. Implementation details are provided in section D.4. The software is used in the context of the experiment described in section 5.1. This work has been published in (Alexiou et al., 2020b). The URL link is:
<https://github.com/mmosp/point-cloud-web-renderer>.

Technical reports

- **Benchmarking of plane-to-plane:** Performance evaluation analysis of the plane-to-plane metric. This work has served as an input document to the JPEG committee. The URL link is:
<https://infoscience.epfl.ch/record/278961>.

Bibliography

- Abreu, A. D., Ozcinar, C., and Smolic, A. (2017). Look around you: Saliency maps for omnidirectional images in vr applications. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. (2018). Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR.
- Akman, O. and Jonker, P. (2010). Computing saliency map from spatial information in point cloud data. In Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., and Scheunders, P., editors, *Advanced Concepts for Intelligent Vision Systems*, pages 290–299, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. T. (2001). Point set surfaces. In *Proceedings of the Conference on Visualization '01, VIS '01*, pages 21–28, USA. IEEE Computer Society.
- Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. T. (2003). Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15.
- Alexiadis, D. S., Zarpalas, D., and Daras, P. (2013). Real-time, full 3-d reconstruction of moving foreground objects from multiple consumer depth cameras. *IEEE Transactions on Multimedia*, 15(2):339–358.
- Alexiou, E. and Ebrahimi, T. (2017a). On subjective and objective quality evaluation of point cloud geometry. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3.
- Alexiou, E. and Ebrahimi, T. (2017b). On the performance of metrics to predict quality in point cloud representations. In Tescher, A. G., editor, *Applications of Digital Image Processing XL*, volume 10396, pages 282 – 297. International Society for Optics and Photonics, SPIE.
- Alexiou, E. and Ebrahimi, T. (2018a). Benchmarking of objective quality metrics for colorless point clouds. In *2018 Picture Coding Symposium (PCS)*, pages 51–55.

Bibliography

- Alexiou, E. and Ebrahimi, T. (2018b). Impact of visualisation strategy for subjective quality assessment of point clouds. In *2018 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–6.
- Alexiou, E. and Ebrahimi, T. (2018c). Point cloud quality assessment metric based on angular similarity. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.
- Alexiou, E. and Ebrahimi, T. (2019). Exploiting user interactivity in quality assessment of point cloud imaging. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Alexiou, E. and Ebrahimi, T. (2020). Towards a point cloud structural similarity metric. In *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–6.
- Alexiou, E., Ebrahimi, T., Bernardo, M. V., Pereira, M., Pinheiro, A., Da Silva Cruz, L. A., Duarte, C., Dmitrovic, L. G., Dumić, E., Matkovič, D., and Skodras, A. (2018). Point cloud subjective evaluation methodology based on 2d rendering. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Alexiou, E., Tung, K., and Ebrahimi, T. (2020a). Towards neural network approaches for point cloud compression. In Tescher, A. G. and Ebrahimi, T., editors, *Applications of Digital Image Processing XLIII*, volume 11510, pages 18 – 37. International Society for Optics and Photonics, SPIE.
- Alexiou, E., Upenik, E., and Ebrahimi, T. (2017). Towards subjective quality assessment of point cloud imaging in augmented reality. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6.
- Alexiou, E., Viola, I., Borges, T. M., Fonseca, T. A., de Queiroz, R. L., and Ebrahimi, T. (2019a). A comprehensive study of the rate-distortion performance in mpeg point cloud compression. *APSIPA Transactions on Signal and Information Processing*, 8:e27.
- Alexiou, E., Xu, P., and Ebrahimi, T. (2019b). Towards modelling of visual saliency in point clouds for immersive applications. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4325–4329.
- Alexiou, E., Yang, N., and Ebrahimi, T. (2020b). PointXR: A toolbox for visualization and subjective evaluation of point clouds in virtual reality. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Alexious, E., Pinheiro, A. M. G., Duarte, C., Matković, D., Dumić, E., da Silva Cruz, L. A., Dmitrović, L. G., Bernardo, M. V., Pereira, M., and Ebrahimi, T. (2018). Point cloud subjective evaluation methodology based on reconstructed surfaces. In Tescher, A. G., editor, *Applications of Digital Image Processing XLI*, volume 10752, pages 160 – 173. International Society for Optics and Photonics, SPIE.

- Ballé, J., Laparra, V., and Simoncelli, E. P. (2016). End-to-end optimized image compression.
- Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G., Levine, J. A., Sharf, A., and Silva, C. T. (2017). A survey of surface reconstruction from point clouds. *Comput. Graph. Forum*, 36(1):301–329.
- Bian, Z., Hu, S.-M., and Martin, R. R. (2009). Evaluation for small visual difference between conforming meshes on strain field. *J. Comput. Sci. Technol.*, 24(1):65–75.
- Bjontegaard, G. (2001). Calculation of average psnr differences between rd-curves. *VCEG-M33*.
- Bletterer, A., Payan, F., Antonini, M., and Meftah, A. (2016). Point cloud compression using depth maps. *Electronic Imaging*, 2016:1–6.
- Bogdanova, I., Bur, A., and Hugli, H. (2008). Visual attention on the sphere. *IEEE Transactions on Image Processing*, 17(11):2000–2014.
- Bonaventura, X., Feixas, M., Sbert, M., Chuang, L., and Wallraven, C. (2018). A survey of viewpoint selection methods for polygonal models. *Entropy*, 20(5):370.
- Botsch, M. and Kobbelt, L. (2003). High-quality point-based rendering on modern gpus. In *11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings.*, pages 335–343.
- Botsch, M., Spornat, M., and Kobbelt, L. (2004). Phong splatting. In *Proceedings of the First Eurographics Conference on Point-Based Graphics*, SPBG'04, pages 25–32, Goslar, DEU. Eurographics Association.
- Botsch, M., Wiratanaya, A., and Kobbelt, L. (2002). Efficient high quality rendering of point sampled geometry. In *Proceedings of the 13th Eurographics Workshop on Rendering*, EGRW '02, pages 53–64, Goslar, DEU. Eurographics Association.
- Bross, B., Han, W. J., Sullivan, G. J., Ohm, J. R., and Wiegand, T. (2012). High efficiency video coding (HEVC) text specification draft 9. Document jctvc-k1003. Joint Collaborative Team on Video Coding (JCT-VC).
- Bulbul, A., Capin, T., Lavoué, G., and Preda, M. (2011). Assessing visual quality of 3-d polygonal models. *IEEE Signal Processing Magazine*, 28(6):80–90.
- Cao, C., Preda, M., and Zaharia, T. (2019). 3d point cloud compression: A survey. In *The 24th International Conference on 3D Web Technology*, Web3D '19, pages 1–9, New York, NY, USA. Association for Computing Machinery.
- Cao, K., Xu, Y., and Cosman, P. (2020). Visual quality of compressed mesh and point cloud sequences. *IEEE Access*, 8:171203–171217.
- Chen, F., Brown, G. M., and Song, M. (2000). Overview of 3-D shape measurement using optical methods. *Optical Engineering*, 39(1):10 – 22.

Bibliography

- Chen, X., Saparov, A., Pang, B., and Funkhouser, T. (2012). Schelling points on 3d surface meshes. *ACM Trans. Graph.*, 31(4):29:1–29:12.
- Chou, P. A., Koroteev, M., and Krivokuća, M. (2020). A volumetric approach to point cloud compression—part i: Attribute compression. *IEEE Transactions on Image Processing*, 29:2203–2216.
- Christaki, K., Christakis, E., Drakoulis, P., Doumanoglou, A., Zioulis, N., Zarpalas, D., and Daras, P. (2019). Subjective visual quality assessment of immersive 3d media compressed by open-source static 3d mesh codecs. In *International Conference on Multimedia Modeling*, pages 80–91. Springer.
- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. (2008). MeshLab: an Open-Source Mesh Processing Tool. In Scarano, V., Chiara, R. D., and Erra, U., editors, *Eurographics Italian Chapter Conference*. The Eurographics Association.
- CloudCompare (2020). (version 2.11) [GPL software].
- Cohen, R. A., Tian, D., and Vetro, A. (2016a). Attribute compression for sparse point clouds using graph transforms. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1374–1378.
- Cohen, R. A., Tian, D., and Vetro, A. (2016b). Point cloud attribute compression using 3-D intra prediction and shape-adaptive transforms. In *2016 Data Compression Conference (DCC)*, pages 141–150.
- Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., and Sullivan, S. (2015). High-quality streamable free-viewpoint video. *ACM Trans. Graph.*, 34(4).
- Corbillon, X., De Simone, F., and Simon, G. (2017). 360degree video head movement dataset. In *Proceedings of the 8th ACM on Multimedia Systems Conference, MMSys’17*, pages 199–204, New York, NY, USA. ACM.
- Corsini, M., Gelasca, E. D., Ebrahimi, T., and Barni, M. (2007). Watermarked 3-d mesh quality assessment. *Trans. Multi.*, 9(2):247–256.
- Corsini, M., Larabi, M. C., Lavoué, G., Petřík, O., Váša, L., and Wang, K. (2013). Perceptual metrics for static and dynamic triangle meshes. *Computer Graphics Forum*, 32(1):101–125.
- da Silva Cruz, L. A., Dumić, E., Alexiou, E., Prazeres, J., Duarte, R., Pereira, M., Pinheiro, A., and Ebrahimi, T. (2019). Point cloud quality evaluation: Towards a definition for test conditions. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Dachsbacher, C., Vogelgsang, C., and Stamminger, M. (2003). Sequential point trees. *ACM Trans. Graph.*, 22(3):657–662.

- David, E. J., Gutiérrez, J., Coutrot, A., Silva, M. P. D., and Callet, P. L. (2018). A dataset of head and eye movements for 360° videos. In *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, pages 432–437, New York, NY, USA. ACM.
- de Oliveira Rente, P., Brites, C., Ascenso, J., and Pereira, F. (2019). Graph-based static 3d point clouds geometry coding. *IEEE Transactions on Multimedia*, 21(2):284–299.
- de Queiroz, R. L. and Chou, P. A. (2016). Compression of 3D point clouds using a Region-Adaptive Hierarchical Transform. *IEEE Transactions on Image Processing*, 25(8):3947–3956.
- de Queiroz, R. L. and Chou, P. A. (2017a). Motion-compensated compression of dynamic voxelized point clouds. *IEEE Transactions on Image Processing*, 26(8):3886–3895.
- de Queiroz, R. L. and Chou, P. A. (2017b). Transform coding for point clouds using a Gaussian process model. *IEEE Transactions on Image Processing*, 26(7):3507–3517.
- Dhond, U. R. and Aggarwal, J. K. (1989). Structure from stereo-a review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510.
- Diniz, R., Freitas, P. G., and Farias, M. C. Q. (2020a). Multi-distance point cloud quality assessment. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3443–3447.
- Diniz, R., Freitas, P. G., and Farias, M. C. Q. (2020b). Towards a point cloud quality assessment model using local binary patterns. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Dricot, A. and Ascenso, J. (2019). Hybrid octree-plane point cloud geometry coding. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5.
- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110.
- Dutagaci, H., Cheung, C. P., and Godil, A. (2010). A Benchmark for Best View Selection of 3D Objects. In *Proceedings of the ACM Workshop on 3D Object Retrieval*, 3DOR '10, pages 45–50, New York, NY, USA. ACM.
- Dutagaci, H., Cheung, C. P., and Godil, A. (2012). Evaluation of 3d interest point detection techniques via human-generated ground truth. *The Visual Computer*, 28(9):901–917.
- Ebner, T. (2018). HHI Point cloud dataset of moving actress. ISO/IEC JTC1/SC29/WG11 Doc. M42152.
- Ebrahimi, T., Foessel, S., Pereira, F., and Schelkens, P. (2016). Jpeg pleno: Toward an efficient representation of visual reality. *IEEE MultiMedia*, 23(4):14–20.
- Eon, E., Harrison, B., Myers, T., and Chou, P. A. (2017). 8i voxelized full bodies, version 2 — a voxelized point cloud dataset. ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) Doc. m40059/M74006.

- Gandoin, P.-M. and Devillers, O. (2002). Progressive lossless compression of arbitrary simplicial complexes. *ACM Trans. Graph.*, 21(3):372–379.
- Garcia, D. C. and de Queiroz, R. L. (2017). Context-based octree coding for point-cloud video. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1412–1416.
- Garcia, D. C. and de Queiroz, R. L. (2018). Intra-frame context-based octree coding for point-cloud geometry. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1807–1811.
- Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., and Yuan, J. (2019). 3d hand shape and pose estimation from a single rgb image. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10825–10834.
- Gelasca, E. D., Ebrahimi, T., Corsini, M., and Barni, M. (2005). Objective evaluation of the perceptual quality of 3d watermarking. In *IEEE International Conference on Image Processing 2005*, volume 1, pages I–241.
- Geng, J. (2011). Structured-light 3d surface imaging: a tutorial. *Adv. Opt. Photon.*, 3(2):128–160.
- Gobbetti, E. and Marton, F. (2004). Layered point clouds. In *Proceedings of the First Eurographics Conference on Point-Based Graphics*, SPBG’04, pages 113–120, Goslar, DEU. Eurographics Association.
- Gokturk, S. B., Yalcin, H., and Bamji, C. (2004). A time-of-flight depth sensor - system description, issues and solutions. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 35–35.
- Golla, T. and Klein, R. (2015). Real-time point cloud compression. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5087–5092.
- Grossman, J. P. and Dally, W. J. (1998). Point sample rendering. In *Eurographics Workshop on Rendering Techniques*, pages 181–192. Springer.
- Gu, S., Hou, J., Zeng, H., and Yuan, H. (2020a). 3d point cloud attribute compression via graph prediction. *IEEE Signal Processing Letters*, 27:176–180.
- Gu, S., Hou, J., Zeng, H., Yuan, H., and Ma, K. (2020b). 3d point cloud attribute compression using geometry-guided sparse representation. *IEEE Transactions on Image Processing*, 29:796–808.
- Guarda, A. F. R., Rodrigues, N. M. M., and Pereira, F. (2019a). Deep learning-based point cloud coding: A behavior and performance study. In *2019 8th European Workshop on Visual Information Processing (EUVIP)*, pages 34–39.
- Guarda, A. F. R., Rodrigues, N. M. M., and Pereira, F. (2019b). Point cloud coding: Adopting a deep learning-based approach. In *2019 Picture Coding Symposium (PCS)*, pages 1–5.

- Guarda, A. F. R., Rodrigues, N. M. M., and Pereira, F. (2020). Deep learning-based point cloud geometry coding: Rd control through implicit and explicit quantization. In *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–6.
- Guennebaud, G. and Paulin, M. (2003). Efficient screen space approach for Hardware Accelerated Surfel Rendering. In *Vision, Modeling and Visualization*, pages 485–495, Munich, Germany. IEEE Computer Society.
- Gumhold, S., Kami, Z., Isenburg, M., and Seidel, H.-P. (2005). Predictive point-cloud compression. In *ACM SIGGRAPH 2005 Sketches, SIGGRAPH '05*, pages 137–es, New York, NY, USA. Association for Computing Machinery.
- Guo, J., Vidal, V., Cheng, I., Basu, A., Baskurt, A., and Lavoue, G. (2016). Subjective and objective visual quality assessment of textured 3d meshes. *ACM Trans. Appl. Percept.*, 14(2).
- Guo, Y., Wang, F., and Xin, J. (2018). Point-wise saliency detection on 3D point clouds via covariance descriptors. *Vis. Comput.*, 34(10):1325–1338.
- Gutiérrez, J., Vigier, T., and Callet, P. L. (2020). Quality evaluation of 3d objects in mixed reality for different lighting conditions. *Electronic Imaging*, 2020(11):128–1.
- Hanhart, P. (2016). *Quality of Experience in Immersive Video Technologies*. PhD thesis, EPFL, Lausanne.
- Hansard, M., Lee, S., Choi, O., and Horaud, R. P. (2012). *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media.
- Hartley, R. I. and Mundy, J. L. (1993). Relationship between photogrammetry and computer vision. In Barrett, E. B. and Jr., D. M. M., editors, *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*, volume 1944, pages 92 – 105. International Society for Optics and Photonics, SPIE.
- Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding*, 68(2):146 – 157.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1992). Surface reconstruction from unorganized points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '92*, pages 71–78, New York, NY, USA. Association for Computing Machinery.
- Hou, J., Chau, L., He, Y., and Chou, P. A. (2017). Sparse representation for colors of 3d point cloud via virtual adaptive sampling. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2926–2930.
- Houshiar, H. and Nüchter, A. (2015). 3d point cloud compression using conventional image compression for efficient data transmission. In *2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT)*, pages 1–8.

Bibliography

- Howlett, S., Hamill, J., and O'Sullivan, C. (2005). Predicting and evaluating saliency for simplified polygonal models. *ACM Trans. Appl. Percept.*, 2(3):286–308.
- Huang, T. and Liu, Y. (2019). 3d point cloud geometry compression on deep learning. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 890–898.
- Huang, Y., Peng, J., Kuo, C. . J., and Gopi, M. (2008). A generic scheme for progressive point cloud coding. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):440–453.
- Huang, Y., Peng, J., Kuo, C.-C. J., and Gopi, M. (2006). Octree-based progressive geometry coding of point clouds. In *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics*, SPBG'06, pages 103–110, Goslar, DEU. Eurographics Association.
- ITU-R BT.2022 (2012). General viewing conditions for subjective assessment of quality of SDTV and HDTV television pictures on flat panel displays. International Telecommunications Union.
- ITU-R BT.500-13 (2012). Methodology for the subjective assessment of the quality of television pictures. International Telecommunications Union.
- ITU-R BT.709-6 (2015). Parameter values for the HDTV standards for production and international programme exchange. International Telecommunication Unionn.
- ITU-T J.149 (2004). Method for specifying accuracy and cross-calibration of Video Quality Metrics (VQM). International Telecommunication Union.
- ITU-T P.1401 (2012). Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models. International Telecommunication Union.
- ITU-T P.910 (2008). Subjective video quality assessment methods for multimedia applications. International Telecommunication Union.
- ITU-T P.913 (2016). Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment. International Telecommunication Union.
- Jackins, C. L. and Tanimoto, S. L. (1980). Oct-trees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3):249 – 270.
- Javaheri, A., Brites, C., Pereira, F., and Ascenso, J. (2017a). Subjective and objective quality evaluation of 3d point cloud denoising algorithms. In *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–6.
- Javaheri, A., Brites, C., Pereira, F., and Ascenso, J. (2017b). Subjective and objective quality evaluation of compressed point clouds. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6.

- Javaheri, A., Brites, C., Pereira, F., and Ascenso, J. (2019). Point cloud rendering after coding: impacts on subjective and objective quality. *arXiv preprint arXiv:1912.09137*.
- Javaheri, A., Brites, C., Pereira, F., and Ascenso, J. (2020a). A generalized Hausdorff distance based quality metric for point cloud geometry. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Javaheri, A., Brites, C., Pereira, F., and Ascenso, J. (2020b). Improving psnr-based quality metrics performance for point cloud geometry. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3438–3442.
- Javaheri, A., Brites, C., Pereira, F., and Ascenso, J. (2020c). Mahalanobis based point to distribution metric for point cloud geometry quality evaluation. *IEEE Signal Processing Letters*, 27:1350–1354.
- Jordan, K. and Mordohai, P. (2014). A quantitative evaluation of surface normal estimation in point clouds. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4220–4226.
- Kammerl, J., Blodow, N., Rusu, R. B., Gedikli, S., Beetz, M., and Steinbach, E. (2012). Real-time compression of point cloud streams. In *2012 IEEE International Conference on Robotics and Automation*, pages 778–785.
- Karni, Z. and Gotsman, C. (2000). Spectral compression of mesh geometry. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00*, pages 279–286, USA. ACM Press/Addison-Wesley Publishing Co.
- Kassner, M., Patera, W., and Bulling, A. (2014). Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. *CoRR*, abs/1405.0006.
- Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP '06*, pages 61–70, Goslar, DEU. Eurographics Association.
- Kazhdan, M. and Hoppe, H. (2013). Screened Poisson Surface Reconstruction. *ACM Trans. Graph.*, 32(3):29:1–29:13.
- Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., and Bhowmik, A. (2017). Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–10.
- Kim, H.-J., Cengiz Öztireli, A., Gross, M., and Choi, S.-M. (2012). Adaptive surface splatting for facial rendering. *Computer Animation and Virtual Worlds*, 23(3-4):363–373.
- Kim, Y., Varshney, A., Jacobs, D. W., and Guimbretière, F. (2010). Mesh saliency and human eye fixations. *ACM Trans. Appl. Percept.*, 7(2):12:1–12:13.

Bibliography

- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Klasing, K., Althoff, D., Wollherr, D., and Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. In *2009 IEEE International Conference on Robotics and Automation*, pages 3206–3211.
- Knorr, S., Ozcinar, C., Fearghail, C. O., and Smolic, A. (2018). Director’s cut - a combined dataset for visual attention analysis in cinematic VR content. In *The 15th ACM SIGGRAPH European Conference on Visual Media Production*.
- Kobbelt, L. and Botsch, M. (2004). A survey of point-based techniques in computer graphics. *Comput. Graph.*, 28(6):801–814.
- Kolb, A., Barth, E., Koch, R., and Larsen, R. (2010). Time-of-flight cameras in computer graphics. In *Computer Graphics Forum*, volume 29, pages 141–159. Wiley Online Library.
- Krivokuća, M., Chou, P. A., and Koroteev, M. (2020). A volumetric approach to point cloud compression–part ii: Geometry compression. *IEEE Transactions on Image Processing*, 29:2217–2229.
- Lavoué, G. (2009). A local roughness measure for 3d meshes and its application to visual masking. *ACM Trans. Appl. Percept.*, 5(4).
- Lavoué, G. (2011). A multiscale metric for 3d mesh visual quality assessment. *Computer Graphics Forum*, 30(5):1427–1437.
- Lavoué, G., Cordier, F., Seo, H., and Larabi, M.-C. (2018). Visual attention for rendered 3D shapes. *Comput. Graph. Forum*, 37:191–203.
- Lavoué, G., Gelasca, E. D., Dupont, F., Baskurt, A., and Ebrahimi, T. (2006). Perceptually driven 3D distance metrics with application to watermarking. In Tescher, A. G., editor, *Applications of Digital Image Processing XXIX*, volume 6312, pages 150 – 161. International Society for Optics and Photonics, SPIE.
- Lavoué, G., Larabi, M. C., and Váša, L. (2016). On the efficiency of image metrics for evaluating the visual quality of 3d models. *IEEE Transactions on Visualization and Computer Graphics*, 22(8):1987–1999.
- Lavoué, G. and Mantiuk, R. (2015). Quality Assessment in Computer Graphics. In *Visual Signal Quality Assessment – Quality of Experience (QoE)*, pages 243–286. Springer.
- Lee, C. H., Varshney, A., and Jacobs, D. W. (2005). Mesh saliency. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH ’05, pages 659–666, New York, NY, USA. ACM.
- Lee, J.-S., Goldmann, L., and Ebrahimi, T. (2013). Paired comparison-based subjective quality assessment of stereoscopic images. *Multimedia tools and applications*, 67(1):31–48.

- Leifman, G., Shtrom, E., and Tal, A. (2012). Surface regions of interest for viewpoint selection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 414–421.
- Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., and Fulk, D. (2000). The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 131–144, USA. ACM Press/Addison-Wesley Publishing Co.
- Levoy, M. and Whitted, T. (1985). The use of points as a display primitive. Technical report, University of North Carolina, Department of Computer Science, Chapel Hill.
- Li, L., Li, Z., Liu, S., and Li, H. (2020a). Efficient projected frame padding for video-based point cloud compression. *IEEE Transactions on Multimedia*, pages 1–1.
- Li, L., Li, Z., Zakharchenko, V., Chen, J., and Li, H. (2020b). Advanced 3d motion prediction for video-based dynamic point cloud compression. *IEEE Transactions on Image Processing*, 29:289–302.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.
- Lindstrom, P. and Turk, G. (2000). Image-driven simplification. *ACM Trans. Graph.*, 19(3):204–241.
- Liu, X., Liu, L., Song, W., Liu, Y., and Ma, L. (2016). Shape context based mesh saliency detection and its applications. *Comput. Graph.*, 57(C):12–30.
- Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA. Association for Computing Machinery.
- Luebke, D. and Hallen, B. (2001). Perceptually Driven Simplification for Interactive Rendering. In Gortle, S. J. and Myszkowski, K., editors, *Eurographics Workshop on Rendering*. The Eurographics Association.
- Mada, S. K., Smith, M. L., Smith, L. N., and Midha, P. S. (2003). Overview of passive and active vision techniques for hand-held 3D data acquisition. In Shearer, A., Murtagh, F. D., Mahon, J., and Whelan, P. F., editors, *Opto-Ireland 2002: Optical Metrology, Imaging, and Machine Vision*, volume 4877, pages 16 – 27. International Society for Optics and Photonics, SPIE.
- Mammou, K. (2017). PCC Test Model Category 2 v0. ISO/IEC JTC1/SC29/WG11 Doc. N17248.
- Mammou, K., Chou, P. A., Flynn, D., and Krivokuća, M. (2019). G-PCC codec description v2. ISO/IEC JTC1/SC29/WG11 Doc. N18189.

Bibliography

- Mammou, K., Tourapis, A. M., Kim, J., Robinet, F., Valentin, V., and Su, Y. (2018). Lifting scheme for lossy attribute encoding in tmc1. ISO/IEC JTC1/SC29/WG11 Doc. M42640.
- Mammou, K., Tourapis, A. M., Singer, D., and Su, Y. (2017). Video-based and hierarchical approaches point cloud compression. ISO/IEC JTC1/SC29/WG11 Doc. M41649.
- Marroquim, R., Kraus, M., and Cavalcanti, P. R. (2007). Efficient point-based rendering using image reconstruction. In *Proceedings Symposium on Point-Based Graphics*, pages 101–108.
- Maugey, T., Meur, O. L., and Liu, Z. (2017). Saliency-based navigation in omnidirectional image. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6.
- Meagher, D. (1982). Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129 – 147.
- Mekuria, R., Blom, K., and Cesar, P. (2017a). Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4):828–842.
- Mekuria, R., Laserre, S., and Tulvan, C. (2017b). Performance assessment of point cloud compression. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4.
- Merry, B., Marais, P., and Gain, J. (2006). Compression of dense and regular point clouds. In *Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa, AFRIGRAPH '06*, pages 15–20, New York, NY, USA. Association for Computing Machinery.
- Meynet, G., Digne, J., and Lavoué, G. (2019). Pc-msdm: A quality metric for 3d point clouds. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3.
- Meynet, G., Nehmé, Y., Digne, J., and Lavoué, G. (2020). PCQM: a full-reference quality metric for colored 3d point clouds. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Mitra, N. J. and Nguyen, A. (2003). Estimating surface normals in noisy point cloud data. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry, SCG '03*, pages 322–328, New York, NY, USA. Association for Computing Machinery.
- MPEG 3D Graphics Coding (2020). Text of iso/iec cd 23090-5 visual volumetric video-based coding and video-based point cloud compression 2nd edition. ISO/IEC JTC1/SC29/WG07 Doc. N0003.
- MPEG 3DG (2017). Common test conditions for point cloud compression. ISO/IEC JTC1/SC29/WG11 Doc. N18474.
- MPEG 3DG (2019). G-pcc codec description v5. ISO/IEC JTC1/SC29/WG11 Doc. N18891.

- MPEG 3DG (2020). Video-based and hierarchical approaches point cloud compression. ISO/IEC JTC1/SC29/WG11 Doc. N19092.
- MPEG 3DG and Requirements (2017). Call for proposals for point cloud compression v2. ISO/IEC JTC1/SC29/WG11 Doc. N16763.
- MPEG Systems (2020). Text of iso/iec dis 23090-18 carriage of geometry-based point cloud compression data. ISO/IEC JTC1/SC29/WG03 Doc. N0075.
- Ochotta, T. and Saupe, D. (2004). Compression of Point-Based 3D Models by Shape-Adaptive Wavelet Coding of Multi-Height Fields. In Gross, M., Pfister, H., Alexa, M., and Rusinkiewicz, S., editors, *SPBG'04 Symposium on Point - Based Graphics 2004*. The Eurographics Association.
- Ohm, J.-R., Sullivan, G. J., Schwarz, H., Tan, T. K., and Wiegand, T. (2012). Comparison of the Coding Efficiency of Video Coding Standards-Including High Efficiency Video Coding (HEVC). *IEEE Trans. Cir. and Sys. for Video Technol.*, 22(12):1669–1684.
- Okutomi, M. and Kanade, T. (1993). A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363.
- Pavez, E., Chou, P. A., de Queiroz, R. L., and Ortega, A. (2018). Dynamic polygon clouds: representation and compression for VR/AR. *APSIPA Transactions on Signal and Information Processing*, 7:e15.
- Pavlidis, G., Koutsoudis, A., Arnaoutoglou, E., Tsioukas, V., and Chamzas, C. (2007). Methods for 3d digitization of cultural heritage. *Journal of Cultural Heritage*, 8(1):93 – 98.
- Peng, J. and Kuo, C. C. J. (2003). Octree-based progressive geometry encoder. In Smith, J. R., Panchanathan, S., and Zhang, T., editors, *Internet Multimedia Management Systems IV*, volume 5242, pages 301 – 311. International Society for Optics and Photonics, SPIE.
- Pereira, F., Dricot, A., Ascenso, J., and Brites, C. (2020). Point cloud coding: A privileged view driven by a classification taxonomy. *Signal Processing: Image Communication*, 85:115862.
- Perry, S. (2018). JPEG Pleno Point Cloud – Use cases and requirements Ver.2.2. ISO/IEC JTC1/SC29/WG1 Doc. N80018.
- Perry, S. (2020). JPEG Pleno Point Cloud Coding Common Test Conditions v3.1. ISO/IEC JTC1/SC29/WG1 Doc. N86044.
- Perry, S., Cong, H. P., da Silva Cruz, L. A., Prazeres, J., Pereira, M., Pinheiro, A., Dunic, E., Alexiou, E., and Ebrahimi, T. (2020). Quality evaluation of static point clouds encoded using mpeg codecs. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3428–3432.

Bibliography

- Pfister, H., Zwicker, M., van Baar, J., and Gross, M. (2000). Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 335–342, USA. ACM Press/Addison-Wesley Publishing Co.
- Preiner, R., Jeschke, S., and Wimmer, M. (2012). Auto splats: Dynamic point cloud visualization on the gpu. In Childs, H. and Kuhlen, T., editors, *Proceedings of Eurographics Symposium on Parallel Graphics and Visualization*, pages 139–148. Eurographics Association 2012.
- Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.
- Qu, L. and Meyer, G. W. (2008). Perceptually guided polygon reduction. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1015–1029.
- Quach, M., Valenzise, G., and Dufaux, F. (2019). Learning convolutional transforms for lossy point cloud geometry compression. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4320–4324.
- Quach, M., Valenzise, G., and Dufaux, F. (2020a). Folding-based compression of point cloud attributes.
- Quach, M., Valenzise, G., and Dufaux, F. (2020b). Improved Deep Point Cloud Geometry Compression. In *IEEE International Workshop on Multimedia Signal Processing (MMSP'2020)*, Tampere, Finland.
- Rai, Y., Gutiérrez, J., and Le Callet, P. (2017). A dataset of head and eye movements for 360 degree images. In *Proceedings of the 8th ACM on Multimedia Systems Conference, MMSys'17*, pages 205–210, New York, NY, USA. ACM.
- Remondino, F. and El-Hakim, S. (2006). Image-based 3d modelling: a review. *The photogrammetric record*, 21(115):269–291.
- Ren, L., Pfister, H., and Zwicker, M. (2002). Object space ewa surface splatting: A hardware accelerated approach to high quality point rendering. *Computer Graphics Forum*, 21(3):461–470.
- Rogowitz, B. E. and Rushmeier, H. E. (2001). Are image quality metrics adequate to evaluate the quality of geometric objects? In Rogowitz, B. E. and Pappas, T. N., editors, *Human Vision and Electronic Imaging VI*, volume 4299, pages 340 – 348. International Society for Optics and Photonics, SPIE.
- Rusinkiewicz, S. and Levoy, M. (2000). Qsplat: A multiresolution point rendering system for large meshes. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 343–352.

- Rusu, R. B. and Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4.
- Sainz, M. and Pajarola, R. (2004). Point-based rendering techniques. *Comput. Graph.*, 28(6):869–879.
- Salvi, J., Pagès, J., and Batlle, J. (2004). Pattern codification strategies in structured light systems. *PATTERN RECOGNITION*, 37:827–849.
- Salvucci, D. D. and Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 71–78, New York, NY, USA. ACM.
- Sansoni, G., Trebeschi, M., and Docchio, F. (2009). State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9(1):568–601.
- Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42.
- Schnabel, R. and Klein, R. (2006). Octree-based point-cloud compression. In Botsch, M. and Chen, B., editors, *Symposium on Point-Based Graphics 2006*. Eurographics.
- Schütz, M. (2016). Potree: Rendering Large Point Clouds in Web Browsers. Master's thesis, Vienna University of Technology.
- Schütz, M., Krösl, K., and Wimmer, M. (2019). Real-time continuous level of detail rendering of point clouds. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 103–110.
- Schütz, M. and Wimmer, M. (2015). High-quality point-based rendering using fast single-pass interpolation. In *2015 Digital Heritage*, volume 1, pages 369–372.
- Schwarz, S., Preda, M., Baroncini, V., Budagavi, M., Cesar, P., Chou, P. A., Cohen, R. A., Kri-vokuća, M., Lasserre, S., Li, Z., Llach, J., Mammou, K., Mekuria, R., Nakagami, O., Siahaan, E., Tabatabai, A., Tourapis, A. M., and Zakharchenko, V. (2019). Emerging mpeg standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):133–148.
- Shao, Y., Zhang, Z., Li, Z., Fan, K., and Li, G. (2017). Attribute compression of 3d point clouds using Laplacian sparsity optimized graph transform. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4.
- Sheikh, H. R. and Bovik, A. C. (2006). Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444.
- Shilane, P. and Funkhouser, T. (2007). Distinctive regions of 3D surfaces. *ACM Trans. Graph.*, 26(2).

Bibliography

- Shtrom, E., Leifman, G., and Tal, A. (2013). Saliency detection in large point sets. In *2013 IEEE International Conference on Computer Vision*, pages 3591–3598.
- Sitzmann, V., Serrano, A., Pavel, A., Agrawala, M., Gutierrez, D., Masia, B., and Wetzstein, G. (2018). Saliency in VR: How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1633–1642.
- Song, R., Liu, Y., Martin, R. R., and Rosin, P. L. (2014). Mesh saliency via spectral processing. *ACM Trans. Graph.*, 33(1):6:1–6:17.
- Sorkine, O., Cohen-Or, D., and Toledo, S. (2003). High-pass quantization for mesh encoding. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, pages 42–51, Goslar, DEU. Eurographics Association.
- Su, H., Duanmu, Z., Liu, W., Liu, Q., and Wang, Z. (2019). Perceptual quality assessment of 3d point clouds. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3182–3186.
- Subramanyam, S., Li, J., Viola, I., and Cesar, P. (2020). Comparing the quality of highly realistic digital humans in 3dof and 6dof: A volumetric video case study. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 127–136. IEEE.
- Tao, P., Cao, J., Li, S., Liu, X., and Liu, L. (2015). Mesh saliency via ranking unsalient patches in a descriptor space. *Computers & Graphics*, 46:264 – 274. Shape Modeling International 2014.
- Tasse, F. P., Kosinka, J., and Dodgson, N. (2015). Cluster-based point set saliency. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 163–171, Washington, DC, USA. IEEE Computer Society.
- Thanou, D., Chou, P. A., and Frossard, P. (2016). Graph-based compression of dynamic 3d point cloud sequences. *IEEE Transactions on Image Processing*, 25(4):1765–1778.
- Tian, D., Ochimizu, H., Feng, C., Cohen, R., and Vetro, A. (2017a). Evaluation metrics for point cloud compression. ISO/IEC JTC1/SC29/WG11 Doc. M39966.
- Tian, D., Ochimizu, H., Feng, C., Cohen, R., and Vetro, A. (2017b). Geometric distortion metrics for point cloud compression. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3460–3464.
- Tian, D., Ochimizu, H., Feng, C., Cohen, R., and Vetro, A. (2017c). Updates and Integration of Evaluation Metric Software for PCC. ISO/IEC JTC1/SC29/WG11 Doc. MPEG2017/M40522.
- Torkhani, F., Wang, K., and Chassery, J.-M. (2012). A curvature tensor distance for mesh visual quality assessment. In *Proceedings of the International Conference on Computer Vision and Graphics - Volume 7594*, ICCVG 2012, pages 253–263, Berlin, Heidelberg. Springer-Verlag.

- Torlig, E. M., Alexiou, E., Fonseca, T. A., de Queiroz, R. L., and Ebrahimi, T. (2018a). A novel methodology for quality assessment of voxelized point clouds. In Tescher, A. G., editor, *Applications of Digital Image Processing XLI*, volume 10752, pages 174 – 190. International Society for Optics and Photonics, SPIE.
- Torlig, E. M., Fonseca, T. A., and de Queiroz, R. L. (2018b). Objective metrics and subjective tests for quality evaluation of point clouds. ISO/IEC JTC1/SC29/WG1 Doc. M78030.
- Upenik, E., Řeřábek, M., and Ebrahimi, T. (2016). Testbed for subjective evaluation of omnidirectional visual content. In *2016 Picture Coding Symposium (PCS)*, pages 1–5.
- van der Hooft, J., Vega, M. T., Timmerer, C., Begen, A. C., De Turck, F., and Schatz, R. (2020). Objective and subjective qoe evaluation for adaptive point cloud streaming. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- van der Hooft, J., Wauters, T., De Turck, F., Timmerer, C., and Hellwagner, H. (2019). Towards 6dof http adaptive streaming through point cloud compression. In *Proceedings of the 27th ACM International Conference on Multimedia, MM '19*, pages 2405–2413, New York, NY, USA. Association for Computing Machinery.
- Váša, L. and Rus, J. (2012). Dihedral angle mesh error: a fast perception correlated distortion measure for fixed connectivity triangle meshes. *Computer Graphics Forum*, 31(5):1715–1724.
- Viola, I. and Cesar, P. (2020). A reduced reference metric for visual quality evaluation of point cloud contents. *IEEE Signal Processing Letters*, 27:1660–1664.
- Viola, I., Subramanyam, S., and Cesar, P. (2020). A color-based objective quality metric for point cloud contents. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Wang, J., Zhu, H., Ma, Z., Chen, T., Liu, H., and Shen, Q. (2019). Learned point cloud geometry compression.
- Wang, K., Torkhani, F., and Montanvert, A. (2012). A fast roughness-based approach to the assessment of 3D mesh visual quality. *Computers and Graphics*, 36(7):808–818.
- Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G. (2018a). Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Wang, X., Koch, S., Holmqvist, K., and Alexa, M. (2018b). Tracking the gaze on objects in 3D: How do people really look at the bunny? *ACM Trans. Graph.*, 37(6):188:1–188:18.
- Wang, X., Lindlbauer, D., Lessig, C., Maertens, M., and Alexa, M. (2016). Measuring the visual salience of 3D printed objects. *IEEE Computer Graphics and Applications*, 36(4):46–55.

Bibliography

- Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.
- Wang, Z., Simoncelli, E. P., and Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2.
- Waschbüsch, M., Gross, M., Eberhard, F., Lamboray, E., and Würmlin, S. (2004). Progressive compression of point-sampled models. In *Proceedings of the First Eurographics Conference on Point-Based Graphics*, SPBG'04, pages 95–103, Goslar, DEU. Eurographics Association.
- Wasenmüller, O. and Stricker, D. (2017). Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In Chen, C.-S., Lu, J., and Ma, K.-K., editors, *Computer Vision – ACCV 2016 Workshops*, pages 34–45, Cham. Springer International Publishing.
- Watson, B., Friedman, A., and McGaffey, A. (2001). Measuring and predicting visual fidelity. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 213–220, New York, NY, USA. Association for Computing Machinery.
- Westoby, M., Brasington, J., Glasser, N., Hambrey, M., and Reynolds, J. (2012). 'structure-from-motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300 – 314.
- WG1 (2020). First call for evidence on jpeg pleno point cloud coding. ISO/IEC JTC1/SC29/WG1 Doc. N86013.
- Wimmer, M. and Scheiblauer, C. (2006). Instant points: Fast rendering of unprocessed point clouds. In *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics*, SPBG'06, pages 129–137, Goslar, DEU. Eurographics Association.
- Wu, J., Shen, X., Zhu, W., and Liu, L. (2013). Mesh saliency with global rarity. *Graph. Models*, 75(5):255–264.
- Yan, W., shao, Y., Liu, S., Li, T. H., Li, Z., and Li, G. (2019). Deep autoencoder-based lossy geometry compression for point clouds.
- Yang, Q., Chen, H., Ma, Z., Xu, Y., Tang, R., and Sun, J. (2020). Predicting the perceptual quality of point cloud: A 3d-to-2d projection-based exploration. *IEEE Transactions on Multimedia*, pages 1–1.
- Yixin Pan, Irene Cheng, and Basu, A. (2005). Quality metric for approximating subjective evaluation of 3-d objects. *IEEE Transactions on Multimedia*, 7(2):269–279.
- Zerman, E., Gao, P., Ozcinar, C., and Smolic, A. (2019). Subjective and objective quality assessment for volumetric video compression. *Electronic Imaging*, 2019(10):323–1.

- Zerman, E., Ozcinar, C., Gao, P., and Smolic, A. (2020). Textured mesh vs coloured point cloud: A subjective study for volumetric video compression. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE.
- Zhang, C., Florencio, D., and Loop, C. (2014). Point cloud attribute compression with graph transform. IEEE - Institute of Electrical and Electronics Engineers.
- Zhang, J., Huang, W., Zhu, X., and Hwang, J. N. (2014). A subjective quality evaluation for 3d point cloud models. In *2014 International Conference on Audio, Language and Image Processing*, pages 827–831.
- Zhang, K., Zhu, W., and Xu, Y. (2018). Hierarchical segmentation based point cloud attribute compression. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3131–3135.
- Zhao, H., Gallo, O., Frosio, I., and Kautz, J. (2016). Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57.
- Zhou, K., Gong, M., Huang, X., and Guo, B. (2011). Data-parallel octrees for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):669–681.
- Zwicker, M., Pfister, H., van Baar, J., and Gross, M. (2001). Surface splatting. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, pages 371–378, New York, NY, USA. Association for Computing Machinery.
- Zwicker, M., Räsänen, J., Botsch, M., Dachsbacher, C., and Pauly, M. (2004). Perspective accurate splatting. In *Proceedings of Graphics Interface 2004, GI '04*, pages 247–254, Waterloo, CAN. Canadian Human-Computer Communications Society.

Evangelos Alexiou

CURRICULUM VITAE

PERSONAL INFORMATION

Date of birth: September 18, 1986
Nationality: Greek
Address: Chemin de la Prairie 5D, Malley, 1007, Lausanne
Telephone: (+41) 76 672 27 95
Email: alexiou.vaggelis@gmail.com

RESEARCH INTERESTS

Perceptual Quality Evaluation, Quality of Experience, Immersive Media, Multimedia Compression, Multimedia Systems.

EDUCATION

- **Ph.D. candidate** Jun 2016 - Dec 2020
Multimedia Signal Processing Group (MMSPG),
Department of Electrical Engineering (EDEE),
Ecole Polytechnique Federal de Lausanne (EPFL), Switzerland.
Thesis: "Perceptual quality of point clouds with application to compression".
Supervisor: Prof. Touradj Ebrahimi.
- **Master of Science** (2-year prog.) Dec 2011 - Apr 2014
Signal Processing for Communications and Multimedia,
Department of Informatics and Telecommunications (DIT),
National and Kapodistrian University of Athens (UOA), Greece.
Thesis: "Real-time high-resolution delay estimation in audio communication using inaudible pilot signals".
Supervisor: Prof. Alexandros Eleftheriadis.
GPA: 9.48/10.0 (2nd among the graduates)
- **Diploma in Electronics and Computer Engineering** (5-year prog.) Dec 2011
Department of Electrical and Computer Engineering (ECE),
Technical University of Crete, Chania (TUC), Greece.
Thesis: "Analysis and simulation of ultrawide band modulation on MATLAB and coexistence with narrow band interference".
Supervisor: Prof. Athanasios P. Liavas.
GPA: 7.64/10.0

WORKING EXPERIENCE

- **Research Assistant**, MMSPG, EPFL, Switzerland.

- *Project*: “Advanced Visual Representation and Coding in AR/VR” Jul 2018 - Dec 2020
[Swiss National Science Foundation]

The objective is to extend the state-of-the-art in immersive communications exploiting advanced content representations in the form of point clouds. In this context, novel frameworks for subjective and objective quality evaluation have been proposed, and deep learning-based encoding paradigms have been developed. Research has been also conducted to capture and analyse user’s behaviour in controlled environments that offer different levels of interactivity; that is, set-ups ranging from desktop arrangements, with users visiting different views through mouse movements, up to VR scenes that offer immersive experiences with 6 degrees-of-freedom.

- *Project*: “ImmersiaTV” Aug 2017 - Jun 2018
[European Unions Horizon 2020]

The objective was to explore and devise efficient coding strategies for delivery of omni-directional video streams. A proof-of-concept implementation making use of the user’s visual attention was proposed by our team.

- *Project*: “RAVE: Random Access Video Encoding” Jul 2016 - Jul 2017
[Swiss Commission for Technology and Innovation]

A web-based application was developed to supports interactive videos through mouse scrolling. Starting from mosaic of images, we advanced to the support of video sequences in order to exploit temporal redundancies and reduce bandwidth requirements. Appropriate video encoding and delivery strategies were developed in order to ensure imperceptible delays for the decoding of video frames that are sought by the user at the receiver side.

- **Business Analyst**, Intrisoft International, Athens, Greece.

- *Project*: ITSM2 under European Commission DG-TAXUD Jul 2015 - Jun 2016

Member of the Business Monitoring team with specialization on Customs for Information Technology Service Management (ITSM2) project. ITSM2 is a project of the European Commission Directorate-General Taxation and Customs Union (DG TAXUD) responsible for providing IT Service Management services for New Computerised Transit System applications (hosting, operations, testing, monitoring, and service support activities).

- **Research Assistant**, DIT, UOA, Greece.

- *Project*: “MusiNet: Networked Music Performance System” Jun 2013 - Sep 2015
[European Social Fund & National Strategic Reference Framework]

A Network Music Performance System was developed to allow distributed musicians to collaborate with each other. Our implementation relied on an open-source and modular SIP User-Agent, where we integrated and fine-tuned state-of-the-art audio and video codecs. We defined a media relay server and investigated techniques for reducing the encoding delay of the streamed video. We integrated a re-transmission algorithm for lost packets of the base layer of video sequences encoded with temporal scalability. Small-scale experiments were conducted to evaluate the impact of influencing factors in real-time musical sessions.

TEACHING EXPERIENCE

- **Teaching Assistant**, EDEE, EPFL, Switzerland.
 - *Courses*: Image and Video Processing, Media Security. Sep 2016 - Jan 2020
 - *Students*: (Co-)supervisor of 18 bachelor/master semester projects. Sep 2016 - Jan 2020
- **Teaching Assistant**, DIT, UOA, Greece.
 - *Courses*: Introd. to Comm. Systems, Music Signal Processing. Jan 2012 - Sep 2015
 - *Students*: Supervisor of 1 undergraduate thesis. Jan 2015 - Sep 2015

PUBLICATIONS

• Journal Articles

1. Alexiou, E., Viola, I., Borges, T., Fonseca, T., De Queiroz, R., & Ebrahimi, T. (2019). A comprehensive study of the rate-distortion performance in MPEG point cloud compression. *APSIPA Transactions on Signal and Information Processing*, 8, E27. (**APSIPA Sadaoki Furui Prize Paper Award**)

• Conference Articles

1. S. Perry et al., “Quality Evaluation Of Static Point Clouds Encoded Using MPEG Codecs,” 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 3428-3432.
2. Evangelos Alexiou, Kuan Tung, Touradj Ebrahimi, “Towards neural network approaches for point cloud compression,” *Proc. SPIE 11510, Applications of Digital Image Processing XLIII*, 1151008.
3. E. Alexiou and T. Ebrahimi, “Towards a Point Cloud Structural Similarity Metric,” 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), London, United Kingdom, 2020, pp. 1-6.
4. E. Alexiou, N. Yang and T. Ebrahimi, “PointXR: A Toolbox for Visualization and Subjective Evaluation of Point Clouds in Virtual Reality,” 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), Athlone, Ireland, 2020, pp. 1-6.
5. E. Alexiou, P. Xu and T. Ebrahimi, “Towards Modelling of Visual Saliency in Point Clouds for Immersive Applications,” 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 2019, pp. 4325-4329. (**Top 10% paper**)
6. L. A. da Silva Cruz et al., “Point cloud quality evaluation: Towards a definition for test conditions,” 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX), Berlin, Germany, 2019, pp. 1-6.
7. E. Alexiou and T. Ebrahimi, “Exploiting user interactivity in quality assessment of point cloud imaging,” 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX), Berlin, Germany, 2019, pp. 1-6. (**Short-listed for best paper**)
8. E. Alexiou et al., “Point Cloud Subjective Evaluation Methodology based on 2D Rendering,” 2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX), Cagliari, 2018, pp. 1-6.

9. Eric M. Torlig, Evangelos Alexiou, Tiago A. Fonseca, Ricardo L. de Queiroz, Touradj Ebrahimi, "A novel methodology for quality assessment of voxelized point clouds," Proc. SPIE 10752, Applications of Digital Image Processing XLI, 107520I.
10. E. Alexiou and T. Ebrahimi, "Impact of Visualisation Strategy for Subjective Quality Assessment of Point Clouds," 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), San Diego, CA, 2018, pp. 1-6.
11. E. Alexiou and T. Ebrahimi, "Point Cloud Quality Assessment Metric Based on Angular Similarity," 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, 2018, pp. 1-6.
12. E. Alexiou and T. Ebrahimi, "Benchmarking of Objective Quality Metrics for Colorless Point Clouds," 2018 Picture Coding Symposium (PCS), San Francisco, CA, 2018, pp. 51-55. (**Short-listed for best paper**)
13. Evangelos Alexious, Antonio M. G. Pinheiro, Carlos Duarte, Dragan MatkoviÄĖ, Emil DumiÄĖ, Luis A. da Silva Cruz, Lovorka Gotl DmitroviÄĖ, Marco V. Bernardo, Manuela Pereira, Touradj Ebrahimi, "Point cloud subjective evaluation methodology based on reconstructed surfaces," Proc. SPIE 10752, Applications of Digital Image Processing XLI, 107520H.
14. E. Alexiou, E. Upenik and T. Ebrahimi, "Towards subjective quality assessment of point cloud imaging in augmented reality," 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Luton, 2017, pp. 1-6. (**Top 10% paper**)
15. Evangelos Alexiou, Touradj Ebrahimi, "On the performance of metrics to predict quality in point cloud representations," Proc. SPIE 10396, Applications of Digital Image Processing XL, 103961H.
16. E. Alexiou and T. Ebrahimi, "On subjective and objective quality evaluation of point cloud geometry," 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX), Erfurt, 2017, pp. 1-3.
17. D. Akoumianakis et al., "The MusiNet project: Addressing the challenges in Networked Music Performance systems," 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA), Corfu, 2015, pp. 1-6.
18. D. Akoumianakis et al., "The MusiNet project: Towards unraveling the full potential of Networked Music Performance systems," IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications, Chania, 2014, pp. 1-6.
19. V. Alexiou and A. Eleftheriadis, "Real-time high-resolution delay estimation in audio communication using inaudible pilot signals," 2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP), Athens, 2014, pp. 290-293.

ACHIEVEMENTS

- **Standardization**

- Active member of JPEG and MPEG standardization activities on point clouds, contributing with input documents (15) and participating in efforts on Exploration Studies by the JPEG AhG on Point Clouds.
- Member of Qualinet organization.

- **Patents**

- RAVE processing structure - USPTO submitted and pending.

- **Grand Challenges**

- *ICIP 2019 Challenge on Point Cloud Coding*: Participation on the preparation of testing material.
- *ICME 2018 Grand Challenge: Point Cloud Coding*: Participation on the preparation of testing material.
- *ICIP 2016 Grand Challenge on Image Compression*: Participation in quality evaluation and analysis of results.

- **Exhibitions**

- *Montreux Jazz Festival 2019*: Participation in project for demonstration of pre-recorded 360 video and 3D audio from live-concerts, consumed by attendees in VR headsets.
- *Scientastic 2018*: Demonstration of current status of JPEG Pleno.
- *Montreux Jazz Festival 2018*: Participation in project with real-time acquisition of 360 video and 3D audio from live-concerts, consumed by attendees in VR headsets.
- *VRForum 2017*: Presentation of poster on point cloud-related activities.

- **Reviewer**

- Served as reviewer at IEEE Transactions on Multimedia, IEEE Signal Processing Letters, ACM MultiMedia, IEEE MultiMedia, Transactions of Applied Perception, IEEE International Conference on Image Processing, IEEE International Conference on Multimedia and Expo, Picture Coding Symposium, IEEE International Workshop on Multimedia Signal Processing.

- **Education**

- Second among graduate students in the Master's program (GPA = 9.48/10.0).
- Excellence award in each class of secondary education (GPA > 18.5/20.0).
- Qualified at the final step of Hellenic Mathematical Society competition, Archimedes (200 out of ~300,000 students from all high school classes).

TECHNICAL SKILLS

- Programming Tools: Python, Javascript, HTML/CSS, C/C++.
- Cross-platform Suites: MATLAB, FFmpeg, Unity, Git.
- Operating Systems: OS X, Linux, Microsoft Windows family.

LANGUAGES

- English: Proficient.
- Greek: Mother-tongue.

REFERENCES

Upon request.