

PointXR: A toolbox for visualization and subjective evaluation of point clouds in virtual reality

Evangelos Alexiou, Nanyang Yang* and Touradj Ebrahimi

Multimedia Signal Processing Group (MMSPG)

École Polytechnique Fédérale de Lausanne (EPFL)

firstName.lastName@epfl.ch

Abstract—In this study, we explore the use of virtual reality to subjectively evaluate the visual quality of point cloud contents. To this aim, we develop the *PointXR toolbox*, a set of Unity applications that can host experiments under variants of interactive and passive evaluation protocols. An auxiliary tool to facilitate the configuration of the supported rendering schemes for point cloud visualization is provided as part of it. Our toolbox is employed to conduct two validating experiments in a virtual environment with 6 degrees of freedom. The purpose is to assess the performance of color encoders that are incorporated in the upcoming MPEG standard on point cloud compression. For this study, we convert a set of mesh models to point cloud contents, and form a high-quality cultural heritage repository, namely, *PointXR dataset*. A comparison between the adopted protocols and the codecs' performance is carried based on the ratings obtained from both experiments. Finally, interactivity patterns based on behavioral data that were recorded during the evaluations are extracted, and results are discussed. The *PointXR toolbox*, the *PointXR dataset*, and the experimental results are made publicly available.

Index Terms—point cloud, virtual reality, subjective evaluation

I. INTRODUCTION

The remarkable advances of augmented reality (AR) and virtual reality (VR) technologies in recent years have resulted in a greater demand for richer imaging modalities to represent more realistic and immersive environments. Relevant applications can benefit from high-quality 3D content representations that can be easily captured, stored and rendered. In this context, point clouds have emerged as an attractive option by providing the possibility of adjusting the visual quality of a model in a per-point basis, eliminating any dependency imposed by connectivity information from acquisition to rendering. A point cloud consists of location information, expressed through coordinates in the 3D space, along with optional attributes that describe underlying surface properties. Typically, to faithfully represent a 3D model, a vast amount of information is required. Therefore, there is a need for efficient data structures and compression algorithms. The latter aim at

reducing the data requirements for storage and transmission, at the expense of visual degradations. Thus, in order to achieve a fine balance between visual quality and data size, quality assessment methodologies are essential.

Subjective evaluation methodologies require the participation of observers in experiments to collect individual ratings of perceptual quality for contents. Although cumbersome and time consuming, they provide accurate scores, since they depend on opinions of targeted end-users. In the case of point clouds, several methodologies have been proposed in the literature providing a foundation for subjective quality assessment for this visual data representation. Yet, the evaluation in VR using 6 degrees-of-freedom (DoF) is largely unexplored, albeit is considered one of the main use-cases. VR does not only enable interactivity and immersive experiences, but it also allows the establishment and reproducibility of identical viewing conditions in fully-controlled environments.

In this work, we exploit the potential of VR in a subjective evaluation experiment using point clouds that are encoded with the MPEG Geometry-based Point Cloud Compression (G-PCC) test model. We extend the capabilities of the most popular point cloud importer/renderer in Unity enabling additional rendering configurations for the point's appearance. Moreover, we develop a set of scenes to allow model display adjustments, and support of interactive and passive evaluation protocols. Finally, a dataset of high-quality cultural heritage point cloud models is generated and used. Our main contributions are:

- Development of the *PointXR toolbox* for point cloud rendering and subjective quality assessment in Unity.
- Generation of the *PointXR dataset*, consisting of high-quality cultural heritage point clouds.
- Subjective evaluation of the G-PCC color encoders in VR.
- Release of the *PointXR*-related material under the URL:

<https://www.epfl.ch/labs/mmspg/pointxr/>

II. RELATED WORK

Point cloud subjective evaluation methodologies can be classified as: (a) passive, and (b) interactive. In the former case, subjects provide their scores based on sequences of images that contain views of the models. In the latter case, subjects interact with the models before scoring their visual quality. For passive methodologies, well-established recommendations for 2D imaging can be employed while ensuring identical

This work has been conducted in the framework of the Swiss National Foundation for Scientific Research project Advanced Visual Representation and Coding in Augmented and Virtual Reality (FN 178854).

*At the time of writing, the author was an exchange student at École Polytechnique Fédérale de Lausanne from Nanyang Technological University.

experience for subjects. However, interactive protocols are closer to natural consumption of 3D models, as interactivity is inherently associated and targeted from related applications. Passive evaluation studies are reported for both static [1]–[4] and dynamic point clouds [5]. The stimuli are captured from a fixed distance using a camera that rotates circularly [2], [4], or from varying distances from a camera that follows spiral motion [1] or a navigation path, potentially, defined per model [3]. Interactive evaluation studies are mainly carried in desktop installations allowing subjects to change their viewpoint using the mouse cursor [6]–[10]. Subjective evaluation in AR is proposed in [11], where point clouds are placed as virtual assets in the real-world and observers are able to physically navigate, whereas in [12], a subjective study is conducted in VR. In the aforementioned studies, both colored [1], [3]–[6], [8]–[10], [12] and colorless [2], [7], [11] models are evaluated. In their majority, compressed point clouds are used, while in some cases [2], [4], [7], [11] other types of degradations (e.g. noise) are additionally employed. No rendering [2], [4], [7], [11], rendering with fixed [5], [12] or adaptive point size [1], [9], [10], real-time voxelization [8] and conversion to mesh [3] have been proposed for display purposes.

For point cloud inspection, processing software such as CloudCompare¹, MeshLab² and Point Cloud Library (PCL)³ denote the most popular alternatives that have been used in subjective evaluation experiments. 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). Potree [13] is a web-based renderer that can handle large point clouds, offering a wide range of features and fast interactivity. A simpler web viewer targeted for subjective quality evaluations was released in [10].

In this study, we develop an alternative rendering solution that enhances the default capabilities of the Unity point cloud importer that is build upon. Enriched rendering features are introduced that can be instrumented for visualization of high-quality models (i.e., shader's shape, point size, shader interpolation). Moreover, we develop a set of scenes with functionalities related to the tuning of a model's appearance, and the hosting of various subjective evaluation protocols.

III. POINTXR TOOLBOX

The Unity platform is used to develop our toolbox. The Pcx Point Cloud Importer⁴ 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

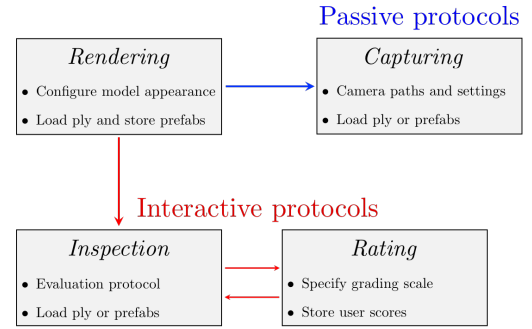


Fig. 1: Color coded block diagram with scene dependencies to enable corresponding evaluation protocols.

distance. In our application, we enhance the natively supported rendering capabilities by integrating the following features: (a) Square 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 algorithm that is deployed is inspired by [1], [10], with the size of each point depending on the K nearest neighbors' distances. The number K can be manually specified per model. (c) Shader interpolation [14], 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. Thus, larger point sizes can be used to avoid hollows, while attaining perception of smooth surfaces.

Our toolbox consists of four Unity scenes, namely, *Rendering*, *Inspection*, *Rating*, and *Capturing*. In Figure 1, 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 graphical user interface (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 *Inspection* scene, when used as a standalone application, consists of the default Unity viewer where point clouds

¹<http://www.danielgm.net/cc/>

²<http://www.meshlab.net/>

³<http://pointclouds.org/>

⁴<https://github.com/keijiro/Pcx>



Fig. 2: Frontal view of the models.

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 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. 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.

All the aforementioned scenes share the same virtual environment. For the scope of this study, a simple room was designed in order to host the subjective evaluations. However, the shape, appearance, material properties, lighting and environmental conditions can be easily adjusted at per user's needs, through the default Unity interface.

IV. VALIDATING EXPERIMENTS

A. PointXR dataset and content preparation

A set of 20 high-quality meshes that represent cultural heritage models were carefully selected from the online platform Sketchfab⁵. The models were generated using typical

photogrammetry tools and represent objects and statues that are exposed in galleries or external locations, making them suitable for virtual museum scenarios. As the majority of them were captured in non-controlled environments, special care was placed into selecting contents without excessive amount of acquisition-related artifacts, such as texture shading or shadows. The mesh models were loaded in Meshlab, and after texel sampling using a texture resolution of 4096×4096 (width \times height), highly dense point clouds were generated and assembled in the so-called *PointXR dataset*.

From the *PointXR dataset*, 5 point clouds were recruited in our experiments. The models were initially voxelized at 10-bit depth. 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 2, indicating the naming that is adopted in the paper and the number of points after voxelization in parenthesis.

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 [15] codecs were used for color compression. To define the configuration parameters for the Octree-plus-Lifting combination, the MPEG Common Test Conditions (CTC) [16] 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 bitrate as Lifting, per model, to secure a fair comparison. This is because although identical QP values are 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 I, 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.

B. Rendering configurations

The reference models were loaded in the *Rendering* scene in order to adjust visualization-related parameters. After experimentation, it was decided to render the models using adaptive

⁵<https://sketchfab.com/>

TABLE I: Encoding configurations per model.

Degradation level	PQS	L-QP	R-QP				
	All	All	Guanyin	Muse	Roy	Shield	Tiki
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

point size based on 3 nearest neighbors, and enable the shader interpolation mode. Both square and disk shaders were evaluated, with the former bringing no visual enhancements under the aforementioned configuration. On the contrary, the rendering performance was improved in terms of frame rate (i.e., fps), when using the square 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 II, the rendering configurations are summarized per model, while the same settings were used for corresponding encoded versions.

TABLE II: Rendering configurations per model.

Rendering factors	Guanyin	Muse	Roy	Shield	Tiki
shader	Square	Square	Square	Square	Square
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

C. Virtual environment

The visual quality of the encoded models was evaluated in an immersive VR environment with 6 DoF. 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. The users 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 the VIVE controller. To avoid additional test parameters, no manipulation of the models (e.g., drag, re-size) was allowed in the experiments.

The virtual environment designed for the test consists 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. At the beginning of each

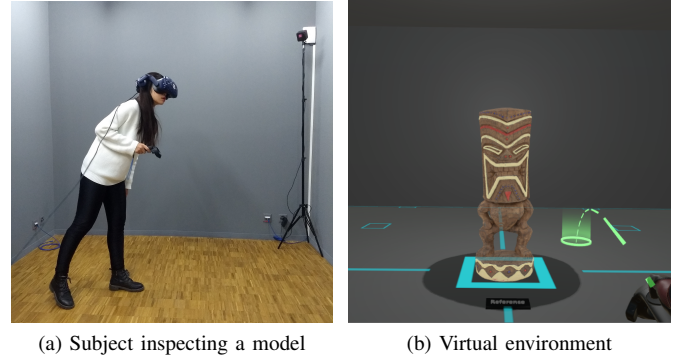


Fig. 3: Experimental setup.

evaluation, the position of the subject was randomly selected in the room to motivate interactivity and inspection from various viewpoints. Note that by using prefab objects generated from the *Rendering* scene, the loading and rendering of the models in the virtual room is immediate. Thus, the experience of the users was not deteriorated by potential loading delays during the evaluation procedure. An example of the virtual environment and a snapshot of a subject inspecting a model are illustrated in Figure 3.

D. Subjective evaluation protocol

In this study, two evaluation protocols were employed, both based on Double-Stimulus Impairment Scale (DSIS) with a 9-grading scale (9: Imperceptible, 7: Perceptible, 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 version. 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. The sequential DSIS experiment chronologically preceded the alternating. Moreover, the stimuli were displayed in a random order per subject and protocol, avoiding consecutive evaluations of the same content.

In our platform, the alternation between the models can be performed instantaneously, since both prefab objects are loaded into the scene at the beginning of each evaluation step. However, in order to avoid flickering that can be sensed even in cases of negligible differences, a delay in the presentation of the next model is 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.

E. Participants

A total of 24 subjects participated in the experiments, with 20 subjects evaluating the models in each protocol. For

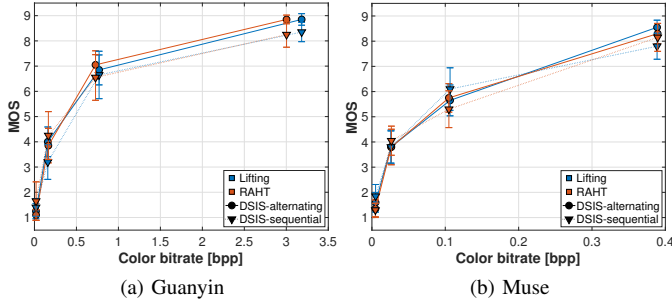


Fig. 4: Subjective scores versus bitrates.

individuals who participated in both sessions, a 2-days 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).

V. 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 [17] was applied on each test, separately. No subject with deviating scoring behaviour was identified and, thus, each individual rating was accounted to compute the Mean Opinion Scores (MOS) and the 95% confidence intervals (CI) 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 4, we present graphs for two of the models used in the experiments, indicating the MOS as a function of the color bitrates, for both codecs and evaluation protocols. The bitrate is presented in bits-per-point (bpp), 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 bitrate 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 bitrates as low as 0.4 bpp to achieve transparency, whereas models with high color variability, such as Guanyin, need higher bitrates. Very similar scoring trends are obtained for the rest of the models.

Our results indicate that subjects rate almost identically in both experiments. 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 confidence intervals in the alternating protocol were found to be by 27% smaller with respect to the sequential, showing that the former 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.

The performance of the color encoders is very similar, according to the subjective quality scores indicated in Figure 4. A non-parametric one-way ANOVA is applied on the scores

obtained from both evaluation protocols separately, per color encoder. The p -values of 0.6927 and 0.809 for the alternating and the sequential DSIS, respectively, indicate that the color codecs are statistically equivalent in both cases.

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, 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. Although such trends are not present 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.

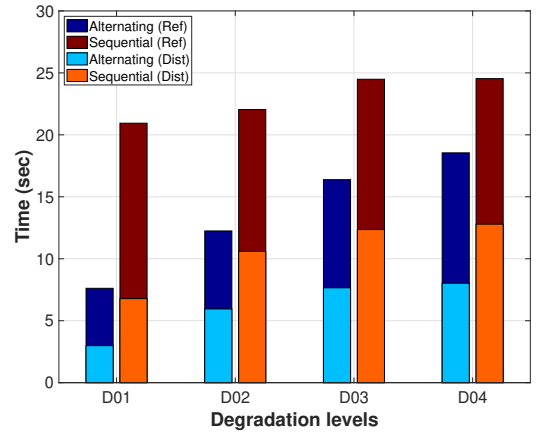


Fig. 5: Average time of interaction.

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. 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 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).

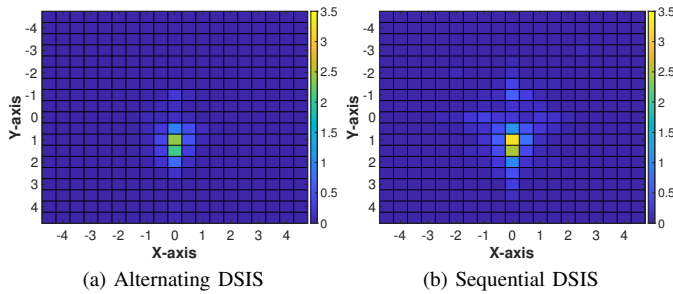


Fig. 6: Interactivity patterns per evaluation protocol.

In Figure 6, 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 encoded, 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.

VI. CONCLUSIONS

In this study, we investigated the use of VR for subjective quality evaluation of point cloud models using 6 DoF. For this purpose, we developed the Unity-based *PointXR toolbox*, which can be configured to carry several variants of interactive and passive evaluation protocols, with enriched rendering options for the appearance of a model. Moreover, we assembled a high-quality point cloud repository of cultural heritage models,

namely *PointXR dataset*. Using the developed applications and a selection of our generated point clouds, we conducted two experiments following the sequential and the alternating DSIS protocols, in order to evaluate the performance of the color encoders of the G-PCC test model. Results highlighted the statistical equivalence of the two color modules used in the compression. Although strong correlation was found among the two evaluation protocols, the alternating variant was found to be preferred. Analysis of the interaction patterns extracted from participants during evaluation showed a preference for close-range, frontal view examination.

REFERENCES

- [1] A. Javaheri, C. Brites, F. Pereira, and J. Ascenso, "Subjective and objective quality evaluation of compressed point clouds," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSp)*, Oct 2017, pp. 1–6.
- [2] E. Alexiou et al., "Point cloud subjective evaluation methodology based on 2D rendering," in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2018, pp. 1–6.
- [3] L. A. da Silva Cruz et al., "Point cloud quality evaluation: Towards a definition for test conditions," in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, Jun 2019, pp. 1–6.
- [4] H. Su, Z. Duanmu, W. Liu, Q. Liu, and Z. Wang, "Perceptual quality assessment of 3D point clouds," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep 2019, pp. 3182–3186.
- [5] E. Zerman, P. Gao, C. Ozcinar, and A. Smolic, "Subjective and objective quality assessment for volumetric video compression," *Electronic Imaging*, vol. 2019, no. 10, pp. 323–1–323–7, 2019.
- [6] R. Mekuria, K. Blom, and P. Cesar, "Design, implementation, and evaluation of a point cloud codec for tele-immersive video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828–842, Apr 2017.
- [7] E. Alexiou and T. Ebrahimi, "On the performance of metrics to predict quality in point cloud representations," in *Proceedings of SPIE*, ser. Applications of Digital Image Processing XL, vol. 103961H, Sep 2017.
- [8] E. M. Torlig, E. Alexiou, T. A. Fonseca, R. L. de Queiroz, and T. Ebrahimi, "A novel methodology for quality assessment of voxelized point clouds," in *Proceedings of SPIE*, ser. Applications of Digital Image Processing XLI, vol. 107520I, Sep 2018.
- [9] E. Alexiou and T. Ebrahimi, "Exploiting user interactivity in quality assessment of point cloud imaging," in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, Jun 2019, pp. 1–6.
- [10] E. Alexiou, I. Viola, T. M. Borges, T. A. Fonseca, R. L. de Queiroz, and T. Ebrahimi, "A comprehensive study of the rate-distortion performance in MPEG point cloud compression," *APSIPA Transactions on Signal and Information Processing*, vol. 8, p. e27, Nov 2019.
- [11] E. Alexiou, E. Upenik, and T. Ebrahimi, "Towards subjective quality assessment of point cloud imaging in augmented reality," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSp)*, Oct 2017, pp. 1–6.
- [12] S. Subramanyam, J. Li, I. Viola, and P. Cesar, "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)*. IEEE, 2020, pp. 127–136.
- [13] M. Schütz, "Potree: Rendering Large Point Clouds in Web Browsers," Master's thesis, Vienna University of Technology, 2016.
- [14] M. Schütz and M. Wimmer, "High-quality point-based rendering using fast single-pass interpolation," in *2015 Digital Heritage*, vol. 1, Sep 2015, pp. 369–372.
- [15] R. L. de Queiroz and P. A. Chou, "Compression of 3D point clouds using a Region-Adaptive Hierarchical Transform," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, Aug 2016.
- [16] MPEG 3DG, "Common test conditions for point cloud compression," ISO/IEC JTC1/SC29/WG11 Doc. N18474, Geneva, Switzerland, Mar 2017.
- [17] ITU-R BT.500-13, "Methodology for the subjective assessment of the quality of television pictures," International Telecommunications Union, Jan 2012.