Thèse n° 10 164

From finger animation to full-body embodiment of avatars with different morphologies and proportions

Présentée le 24 novembre 2023

Faculté informatique et communications
Groupe SCI IC RB
Programme doctoral en robotique, contrôle et systèmes intelligents

pour l'obtention du grade de Docteur ès Sciences

par

Mathias Guy DELAHAYE

Acceptée sur proposition du jury

Dr J. Skaloud, président du jury
Dr R. Boulic, Dr B. Herbelin, directeurs de thèse
Dr J. Pettré, rapporteur
Prof. G. Papagiannakis, rapporteur
Dr P. Pu, rapporteuse

École
polytechnique
fédérale
de Lausanne

2023

We were here
— Nightwish

To my family and friends . . .

Acknowledgements

I would like to express my sincere gratitude to my two thesis advisors. Firstly, Dr. Ronan BOULIC for his personal investment, for his daily presence, support, and constant availability. He shared with us swimming tips in Bern and was often ready for swimming breaks at noon in the lake. He also gave me the opportunity to contribute to the C++ and VR courses. I would like to thank Dr. Bruno HERBELIN for his pronounced research mind, for his insightful advice, and also for the good moments we shared outside of the work.

I extend my thanks to the committee members who reviewed my manuscript and participated in my thesis defense, namely Dr. Jan SKALOUD, president of the jury, Dr. Pearl PU, Dr. Julien PETTRÉ, and Prof. George PAPAGIANNAKIS. I am also grateful to Prof. Olaf BLANKE for having hosted me as a member of the LNCO during my thesis journey.

Furthermore, I express my appreciation to the EPFL for providing me with all the necessary resources and support for my research. Their efficient and straightforward procedures, prompt responses, and pleasant environment have been highly beneficial. I would also like to thank the EPFL Service Desk team, in particular Carlos PEREZ and Stéphane ECUYER, for their technical support and assistance in deploying experimental projects, such as the autograder for the ICC course, and for their help in setting up solutions to allow prototype devices to function correctly.

In addition, one floor above, I would like to thank my colleagues, starting with Dr. Thibault PORSSUT, for his friendship and for all the fun activities we participated in together, such as hiking, biking, kayaking, and enjoying pizza on Saturday evenings in Morges. Special thanks go to Loën BOBAN for being a test subject, for her help with literature, or for food discussion and coffee breaks at Sat'. Thanks to Phil LOPES and Nana TIAN for their assistance during lab meetings and their unique cultural insights. I was also pleased to collaborate with my colleagues at LNCO, as well as with my peers from the ICC assistantship sessions, namely Pierre OPPLIGER and Vincent TOURNIER. Through this experience, I gained valuable insight into the role of an educator and the responsibilities that come with it. This is why I would like to warmly thank my former teachers, in particular Prof. Luc FABRESSE, Prof. Arnaud DONIEC, Prof. Noury BOURAQADI, Stéphane VANACKER, Prof. Loïc LAGADEC, Lionel GIRARD, Alexis PLANCHE, David GASECKI, and Claire SANTONI, for their support and their knowledge they provided me during my studies.

Acknowledgements

Throughout this thesis journey, which was characterized by various challenges and unexpected events, I had the opportunity to explore the UNIL-EPFL Sports Center and immerse myself in its diverse range of activities, and I am deeply thankful for this. I appreciated a lot the level of its instructors and the food quality of its cafeteria. I wish to convey my sincere appreciation and my thanks to the people from the associations Rock Affinity and the Club Photo EPFL for the unforgettable experiences and friendships I had through their events that I enjoyed a lot. In particular, I express my sincere gratitude to Oriane FOUSSADIER, Valentine DELEVAUX, and Sara AIT-TORJMAN, for being such attentive listeners and great dancers; to Anastasiia NALYVAIKO, Julien MARENGO, Michel SCHINZ, Ekaterina SVIKHNUSHINA, and Stanislav SERGEEV for the mountain hikes; to Miguel BASANTE, Luca GALLO, and David BARIONI for organizing salsa evenings and to all my other friends, that I cannot list here, in Switzerland!

I extend my heartfelt gratitude to my friend Thibaud HONNORAT, who did his final year internship in Geneva so that we could watch countless movies in theaters that were always accompanied by great burgers. I am grateful for the cherished memories and enjoyable projects I shared with my friends Yvain LUZEAU, Thibaud CANTALUPPI, and Cédric DEROZIER from Mines Douai and with Clovis MARCHETTI from Unéole. In a similar vein, I would like to express my immense gratitude to Guillaume and Clément BROGGI, as well as Margot OLIVE, for the numerous activities, including the memorable hikes, card games, and delicious fondues in Morges. David RENAUD and Baptiste LEMMON were frequently in attendance during those hikes, and I would like to express my gratitude towards them for the various activities we participated in together, including museum visits, trips to Bern, and festivals.

I would like to express my gratitude to friends within my family circle with Ferdinand DUTILLY, Adrien DELESTRE, Aliénor MAXENCE, Virginie ROLET, Thierry ROUYER and Marlène BONNET, Michel POUJADE, which followed my thesis journey with interest.

Finally, I am forever grateful to my entire family for their unwavering support, unconditional love, and constant presence in my life. Thanks to my brother for our frequent five-minute calls that somehow turn into three-hour conversations, for sharing common passions for car stories, more recently for computer science, and most importantly, for all the fun. Thanks to my sister for her artistic assistance and for all the family events she organized. Lastly, I want to express my deep gratitude to my parents for their love, for having instilling me with important principles and skills in me since I was a child, including using tools, riding a bike, programming, driving, fixing leaks in pipes, or for having nursed my wounds. Their presence in my life has been constant, and no amount of thanks could ever be enough to say how much they were and are important to me.

Unfortunately, it would be difficult to list here all the people who contributed to my life, but for all those kind-hearted individuals that are missing here, I would like to express my immense gratitude to them!

Lausanne, May 1st, 2023 Mathias DELAHAYE

Abstract

VR (Virtual Reality) is a real-time simulation that creates the subjective illusion of being in a virtual world. This thesis explores how integrating the user's body and fingers can be achieved and beneficial for the user to experience VR.

At the advent of VR, the original idea was to completely transpose users' bodies, including hands and fingers, so that we could still be able to see our own movements in the VE (Virtual Environment). However, technological limitations prevented the full integration of the body, hence negatively impairing the subjective user experience. Hands-tracking devices were replaced by controllers and the body integration was, for a time, forsaken. Recently, Mocap (Motion Capture) systems became more reliable, more convenient, and the concern for body integration came back. Body movements can be tracked in real-time with trackers strapped on the body limbs, and IK (Inverse Kinematic) can be used to animate avatars' skeletons from the MoCap data. However, MoCap techniques were not still sufficiently robust to reliably track finger motion, hence preventing our primary way to interact with the environment from being visible in the VE. To address this critical issue, we proposed an approach relying on an active camera-based MoCap (Motion Capture) system to animate virtual hands and fingers in real-time. Here, a first neural network was used to fill the gaps in the input due to occlusions, while a second one was used to provide an IK (Inverse Kinematic) solution handling the animation of the hand and fingers. This method focused on maintaining plausible poses (eventually with slight distortions in the movements) to compensate for tracking errors rather than seeking a perfect MoCap system that would not present any drawback.

To confirm the usability of our approach, we investigated, through a user study, whether one could tolerate those errors in the finger animation through the evaluation of an even more distinct type of distortion than motion amplification in the context of succeeding interactions: finger swaps. Our results showed that participants mostly took credit for introduced finger swaps, to the point where participants could bearly notice when they were helped, allowing us to provide guidelines to avoid disrupting the SoE when animating avatar fingers.

The learned mechanisms of the cognitive functioning of the SoE at the finger levels combined with the knowledge from the literature on arm/leg reaching movements were then integrated together to provide an approach aiming to avoid BiE (Break in Embodiment) when embodying avatars, with different shapes and proportions, animated at both the body and finger levels. This approach relies on an active optical MoCap system (to acquire the user's movements), a user's body calibration procedure (to construct a numerical model of the user's morphology), and an animation pipeline to transfer the original motion from the user onto an avatar with different shapes and sizes in real-time. The proposed approach was evaluated through a subjective evaluation procedure comparing the proposed approach against a full-body animation using direct forward kinematics and our results showed that the retargeted approach outperformed the direct kinematics forward one. Despite the smaller effect size observed than initially expected, the evaluation highlighted the necessity of adapting the motion, even if the avatar and the user look similar.

Résumé

La Réalité Virtuelle (VR) est une simulation en temps réel qui crée l'illusion subjective d'un monde virtuel pour l'utilisateur. La thèse suivante explore l'intégration du corps et des doigts de l'utilisateur dans l'environnement virtuel (VE) ainsi que ses bénéfices pour l'expérience utilisateur en VR.

À l'aube de la réalité virtuelle (VR), l'idée originale était de transposer complètement le corps des utilisateurs, y compris les mains et les doigts, de sorte que nous puissions toujours voir nos propres mouvements dans l'environnement virtuel (VE). Cependant, des limites technologiques ont empêché l'intégration complète du corps, affectant négativement l'expérience subjective de l'utilisateur. Les dispositifs d'acquisition de mouvement des mains ont été remplacés par des contrôleurs et l'intégration du corps a été mise de coté pendant un certain temps. Récemment, les systèmes de capture de mouvement (MoCap) sont devenus plus fiables, plus pratiques, et l'intégration du corps est revenue un sujet de premier plan. Les mouvements du corps peuvent être capturés en temps réel à l'aide de capteurs attachés sur les différentes parties du corps, et la cinématique inverse (IK) est utilisée pour animer les squelettes des avatars à partir des données de la capture de mouvement (MoCap). Cependant, les techniques de MoCap ne sont pas encore suffisamment robustes pour suivre de manière fiable les mouvements des doigts, empêchant ainsi notre principal moyen d'interaction avec l'environnement d'être visible dans l'environnement virtuel (VE).

Pour résoudre ce problème critique, nous avons proposé une approche reposant sur un système de capture de mouvement (MoCap) basé sur des marquers actifs et suivi par des caméras, afin d'animer en temps réel les mains et les doigts virtuels de l'utilisateur. Ici, un premier réseau neuronal est utilisé pour combler les données manquantes de la capture liées aux occlusions, tandis qu'un deuxième réseau est utilisé pour fournir une solution de cinématique inverse (IK) gérant l'animation de la main et des doigts. Cette méthode se concentre sur le maintien de poses plausibles (éventuellement avec de légères distorsions dans les mouvements) pour compenser les erreurs de capture à la place de rechercher une solution technologique de MoCap qui ne présenterait aucun inconvénient.

Pour confirmer l'utilisabilité de notre approche, nous avons étudié, à travers une étude utilisateur, à savoir si l'on pouvait tolérer ces erreurs d'animation des doigts à via l´évaluation d'un type de distorsion encore plus marqué que l'amplification d'un mouvement dans le contexte de réussir une interaction : la permutation de l'animation de deux doigts. Nos résultats ont montré que dans la grande majorité des cas les participants se sont attribué les mouvements permutés des doigts, au point où les participants ne se sont pratiquement pas rendu compte d'avoir été aidé,

nous permettant ainsi de fournir directives pour éviter de perturber le sens d'incarner le corps virtuel (SoE) lors de l'animation des doigts de l'avatar.
La compréhension des mécanismes de ce fonctionnement cognitif au niveau des doigts combinés aux connaissances de la littérature sur les mouvements de bras/jambes visant à atteindre un objet, ont ensuite été intégrés pour proposer une approche visant à éviter le rejet de l'avatar, animé au niveau du corps et des mains, lorsque ce dernier présente différentes formes et proportions.
Cette approche repose sur un système actif de MoCap optiques (pour acquérir les mouvements de l'utilisateur), une procédure de calibrage du corps de l'utilisateur (pour construire un modèle numérique de la morphologie de l'utilisateur) et un pipeline d'animation pour transférer le mouvement d'origine de l'utilisateur sur un avatar avec des formes et des tailles différentes en temps réel. L'approche proposée a été comparée, à travers une procédure d'évaluation subjective, à une animation du corps reposant sur l'application directe des rotations des joints du model sur l'avatar, et nos résultats ont montré que l'approche de retargeting surpassait la seconde méthode.
En dépit d'une taille d'effet inférieure à ce qui était initialement prévu, l'évaluation a souligné la nécessité de l'adaptation du mouvement, même si l'avatar et l'utilisateur se ressemblent.

Contents

Acknowledgements **i**

Abstract **iii**

Résumé **v**

List of Figures **xi**

List of Tables **xvii**

1 Introduction **1**

1.1 VR use and limitations 2

1.1.1 VR field of applications 2

1.1.2 Training in VR with tangible haptic elements: When controllers become a limitation 3

1.2 Immersion, presence, and embodiment in VR (Virtual Reality) 4

1.2.1 Immersion: the characteristics of the display to the Virtual Environment 4

1.2.2 Presence 5

1.2.3 The SoE (Sense of Embodiment) and the role of the body 5

1.3 Providing a virtual body in immersive VR 7

1.3.1 The virtual body as an animated 3D character 8

1.3.2 Motion Capture technologies 9

1.3.3 Mitigation techniques for unreliable MoCap input 11

1.3.4 From MoCap data to the animation of a 3D model with inverse kinematics (IK) 12

1.3.5 Retargeting: the adaptation of one's body pose to a different virtual body 12

1.4 Research plan 14

2 Providing users with finger animation in VR **17**

2.1 Introduction 17

2.2 Related work 18

2.3 Our Approach 20

2.3.1 Training dataset acquisition 21

2.3.2 Occlusion recovery model 26

2.3.3 Finger animation model 32
2.3.4 Training 32
2.3.5 Second hand pipeline mirroring 33
2.4 Approach validation 34
2.4.1 Evaluation dataset 34
2.4.2 Results 36
2.4.3 Performance 39
2.5 Discussion and future work 39
2.5.1 Limitations 40
2.5.2 Future works 42
2.6 Conclusion 42

3 How are errors in finger animation perceived? 43
3.1 Introduction 43
3.2 Setup 46
3.3 Task 47
3.4 Implementation 49
3.4.1 Speed regulation 49
3.5 Experimental design 49
3.6 Hypotheses, measurements, and analysis 51
3.6.1 Formal hypotheses 51
3.6.2 Measurements 52
3.6.3 Statistical analysis 53
3.7 Results 54
3.7.1 Demographics 55
3.7.2 Finger swap detection 55
3.7.3 Confidence level and reaction time analysis 56
3.8 Discussion and future work 56
3.8.1 Relation with agency and embodiment 57
3.8.2 Limitations and future works 58
3.9 Conclusion 59

4 Integration of finger and full body animation with self-contact consistency 61
4.1 Introduction 61
4.2 System overview 62
4.3 Setup 63
4.4 Users' body calibration 65
4.4.1 Hands and fingers calibration 66
4.4.2 Feet calibration 67
4.4.3 Limb calibration 67
4.4.4 Head calibration 71
4.4.5 Trunk calibration 72
4.5 Offline avatar pre-calibration 73

4.6 Online retargeting 74
  4.6.1 Skeleton model animation 74
  4.6.2 User egocentric coordinates computation 79
  4.6.3 Avatar posture adaptation loop 88
4.7 Discussion and future works 97
4.8 Conclusion 99

5 Subjective evaluation of the full body animation 101
5.1 Video dataset 101
5.2 Subjective evaluation procedure 102
5.3 Analysis 106
5.4 Results 106
5.5 Discussion 108
5.6 Conclusion 108

6 Synthesis and conclusion 109
6.1 Finger animation and user perception 109
6.2 Full body animation down to the finger level 111
6.3 What about first-person viewpoint subjective experience? 114
6.4 Conclusion 115

A On the Importance of Providing a Tangible Haptic Response for Training Cardiopulmonary Resuscitation in Virtual Reality 117

B Training in VR with tangible haptic elements: when controllers become a limitation (e.g., CPR) 119
B.1 Introduction 119
B.2 Related work 122
B.3 System overview 123
  B.3.1 Data acquisition 123
  B.3.2 Hand tracking validation 125
  B.3.3 Visualization 129
B.4 Experimental evaluation 131
  B.4.1 Hypotheses 131
  B.4.2 Method 131
B.5 Results 132
  B.5.1 Presence and embodiment levels 133
  B.5.2 Performance quality 133
B.6 Discussion 133
B.7 Conclusion 136
B.8 Acknowledgement 137

C How errors involving a finger swap in finger animation are perceived? 139
C.1 Demographic questionnaire 139

D Does scaling player size skew one’s ability to correctly evaluate object sizes in a virtual environment? 143

E User study of omnidirectional treadmill control algorithms in VR 151

Bibliography 167

List of Acronyms 169

Glossary 169

Curriculum Vitae 171

List of Figures

1.1 Cardiopulmonary Resuscitation training in VR: setup with the tracked mannequin device and tracked hand (left) and first PV (Person Viewpoint) with performance feedback provided in the HMD (right) 3

1.2 Illustration of the DataGlove ™from Zimmerman et al. (1986) used to move 3D objects in a virtual scene. 4

1.3 Milgram's Virtuality continuum with on the left the real world as we perceive it, and on the full right the immersive VR where the user can believe he is in another world. Illustration adapted schema from Zlatanova (2002) 5

1.4 On the left is the source position of the user, and on the right, the pose retargeted on different avatars produced respecting the self-contact with the body. 7

2.1 Conceptual full prediction pipeline. 20

2.2 The virtual hand (left) and the mocap glove (right). 21

2.3 Illustration of the ground truth system (right), with its numerical model (left), used to create the training dataset. 22

2.4 Illustration of the hand calibration pose 24

2.5 Illustration of the simple calibration 24

2.6 Finger alignment post-processing. 25

2.7 Conceptual full prediction pipeline 26

2.8 An autoencoder with three hidden layers. In a scenario where **a** and **c** are available, and **b** is occluded, we disconnect the inputs corresponding to **b**, and we get the prediction of **b** in the output. Here, only three markers are depicted; in practice, we would have nine markers. 31

2.9 Handling of discontinuities. **(a)** At $t-1$ no marker is occluded. **(b)** At t the index marker is occluded. The NN predicts its hypothetical position at $t-1$ (green), which results in a small discontinuity from the true position (red). **(c)** From t onwards, the discontinuity is explicitly canceled by moving the marker by the offset vector (black arrow). For re-entry discontinuities, the process is reapplied in the opposite direction, but the offset vector is progressively shrunk to remove the bias. 32

2.10 Degrees of freedom of the hand joints (26 in total). 33

2.11 Main steps of the pipeline used to transform raw markers position from VRPN to the position of the avatar's body. Refer to Figure 2.7 for details of the left-hand model 34
2.12 **Left:** a person wearing the recording equipment (PhaseSpace glove and Perception Neuron). **Right:** a person testing the application in a VR environment with an Oculus HMD. 35
2.13 A subject wearing the two gloves during the recording phase of the dataset . . . 35
2.14 Probability of N occlusions at once (Left). / Occlusion duration histogram (Right). 36
2.15 Probability of N occlusions at once (Left). / Occlusion duration histogram (Right). Plots represent the same as in Figure 2.14 37
2.16 **(a)** Comparison between the baselines and our method for the marker predictor error. The moving average baseline is not included because of its excessive error. **(b)** End-to-end error: our approach at varying conditions versus an IK baseline. 38
2.17 Too many occlusions 40
2.18 An illustration of the gap between the two fingers 41
2.19 Fast movement inducing a bad marker position 42

3.1 Experimental context 47
3.2 Schematic illustration of a swap in fingers' motion. The original motion of the index (i.e., the real motion from the user, the dashed arrow) is redirected onto the middle finger the user sees (i.e., the displayed motion, the cyan arrow). 47
3.3 Timeline illustration of the experimental protocol of the experiment. 50
3.4 Plots of the results from the Analysis section 54

4.1 System overview: the upper stage on the schema represents the avatar calibration; this step can be performed once using the Avatar Calibrator, and the calibration file can be stored for future use. The middle stage corresponds to the user's calibration process. Here the user performs several gym motions and self-contacts to calibrate its virtual skeleton and approximation body. Finally, with the user's virtual skeleton calibrated and the avatar's calibration profile, the third and lower stage computes in real-time the instantaneous pose to be applied to the avatar. . 62
4.2 Vive Tracker 3.0 are consumer-grade devices whose 3D localization can be retrieved through SteamVR. Those trackers are well suited for the gaming experience, for tracking limbs (e.g., feet, knees, etc.) or objects to interact with (e.g., tracking a dummy gun). However, their dimensions prevent them from being placed on each finger. Picture sourced and edited from https://www.vive.com/fr/accessory/tracker3/ 64
4.3 The gloves' black texture helps to reduce light reflections from the tracking LED to enhance tracking. Each fingertip has a LED to track its position in the 3D space. The wooden support provides a rigid body reference to reduce the LED lateral motion due to the gloves' flexibility. 64

4.4 The setup involves a mix of Vive Trackers 3.0 with homemade tracking gloves, therefore mixing tracking solutions. Vive Trackers are wireless, reducing the user's constraints, while the tracking gloves are still wired to receivers that users keep in their pockets. 65

4.5 The middle plan that separates both hands is illustrated in cyan. Its location is computed locally to both hands referential (i.e., computed left-hand palm plan and computed right-hand palm plan) so that each hand model knows where is its contact surface. This pose also determines the fingertips' radius and local positions of extended fingers used in the animation stage to animate fingers' kinematic chains. 67

4.6 To calibrate a point, one must place the other hand's fingertips (the index by default) on top of the point of interest (e.g., joint) The order in which the user calibrates points is irrelevant, as all calibration points are first stored. The actual computation can be triggered later, hence avoiding flipping the hand several times to calibrate the wrist, surfaces, and finger base joint. 68

4.7 The foot calibration process expects the user to place their fingertips on the edges of his foot to calibrate the foot's planar surface in contact with the floor, with a projection applied (small oranges arrow) to ensure the measured position is on the floor. The user also places the fingertip on both sides of the malleolus to calibrate the ankle's joint position. 69

4.8 The linear bone b_k has a structure containing an origin (r_k) and a bone vector $\overrightarrow{b_k}$ that links the proximal joint to the distal joint e_k. Each bone has its referential in which the bone axis is along the local forward direction $\overrightarrow{z}$. The bone consequently also has a local right ($\overrightarrow{x}$) and a local up ($\overrightarrow{y}$). The local right is set as the joint flexion axis (right-sided) for limbs. Constraining the axis direction and its local right (or up) is enough to constrain its world orientation fully. Setting the origin's position fixes the remaining degrees of freedom that fully constrain the bone's placement. 70

4.9 The position of the parent bone's tracker is recorded on a time window of 150 frames (this allows a sufficient average for measurements of the CoR when the user moves the limb). It generates a cloud of points illustrated in black, covering the history of the tracker positions, which is used to retrieve the unique normal corresponding to the joint direction's local right. N.B. Unity uses a left-handed referential. Thus, the output of the cross-product is the opposite of what is expected with a right-handed referential. 70

4.10 Except for the spine, which might contain a different number of joints compared to the user's skeleton model, the calibrated avatar contains the same structure as the user's skeleton. This means that each surface element has an equivalent in the source user model. 72

4.11 Example demonstrating the problem that can arise when the polygon count in the crude mesh is too low. In this case, the right hand is positioned correctly on the surface of the crude mesh, but since the crude mesh represents is a chord of the belly's rounded shape, the hand interpenetrates with the belly. 74

4.12 Capsule bones are calibrated to represent the user's limb shape, and finger digits bones, while the crude mesh represents the shape of the torso, hand and feet, palm surfaces, jaw, and a sphere approximates the back of the skull's surface. . 75

4.13 Flexing finger based on the effector-finger base joint distance. Flexing the finger is insufficient to enforce the effector's position to match the finger's kinematic chain extremity. 76

4.14 In the realignment process, the algorithm measures the difference in yaw and pitch between the position of the user's fingertip and the only flexed finger model. Then, the rotation required to align the reference axis with the user's finger is computed as the rotation that rotates the reference axis by the differences in yaw and pitch. This rotation is then applied to all the joints of the finger, and the position of the distal end of each digit segment is computed to update the origin of the next proximal digit side. 77

4.15 Illustration of the process of linking bones on a limb 78

4.16 The spine animation process comprises a first resolution of the spine using the IK method from Unzueta et al. (2008), followed by a realignment and a second pass. 79

4.17 Illustration, in black, of the shortcut bones used to skip intermediate bones for the normalization computation process. 80

4.18 Illustration, adapted from Molla et al. (2017), representing the decomposition of a point's position into a surface contact point and a vector. 81

4.19 In the egocentric coordinate system, each point position p_j is decomposed into a sum of contributions from each element surface s_i. Those surface elements can either be a sphere (on the left), a mesh triangle (middle), or a cylinder (on the right). The contribution for each surface element to the jth target point is denoted $\overrightarrow{v_{i,j}}$ (in red) and represents the vector between p_j and the closest projection point of p_j on s_i that is noted $x_{i,j}$ (in green). Finally, a normalization factor $\tau_{i,j}$ is computed as the sum of the dot product of each bone's length and $\overrightarrow{v_{i,j}}$. 82

4.20 Illustration of the set of vector contributions $(\overrightarrow{v_{i,j}})_{i\in\mathbb{S}}$ for a single target point p_j. N.B. The thumb wireframe layer overlaps the extremities of the contribution vectors. 84

4.21 Illustration of the same pose on different avatars with and without the normalization enabled. The structure on the left is the modelized user's skeleton. On top, the non-normalized animation produces flexions in the characters with long arms (the two avatars on the right), while for the child, the arm is completely extended, although this was not the case on the source skeleton. On the bottom, the normalization of effector positions straightens the arms of the two characters on the right and reduces the extension of the child's arms. 86

4.22 Illustration of the kinematic chain used to compute the normalization factor τ. The spine model was simplified to reduce the computational cost. 87

4.23 Performance measurements between parallelized (CPU and GPU) and sequential pipeline processing for egocentric computation. We can observe that the largest time consumption is drawn by the computation of $x_{i,j}$, $\hat{x}_{i,j}$, $v_{i,j}$ and $\Lambda_{i,j}$. Consequently, we implemented several pipelines using CPU and GPU parallelization to reduce the latency. Values measured on a desktop equipped with an Intel Core i7-4790 CPU (4 Cores at 3.6GHz, boosting at 4GHz) with 8GB of DDR3 RAM and an NVIDIA GeForce GTX 970 GPU with 4GB of DDR5 SDRAM 88

4.24 The posture adaptation pipeline is applied for each frame. It resets the placement of the avatar and pre-orient limbs using direct kinematic forward. Limbs are then progressively attracted toward their retro-projected target points (p'_j). Fingers are then initialized using direct kinematics and are also progressively attracted toward their retro-projected target points to produce the final avatar pose. . . . 89

4.25 Illustration of the computation of the contribution vectors and their normalized re-application on the destination avatar to compute the avatar's position. 92

4.26 Three target points are assigned to each effector used as a rigid body referential that allows the expression of the wrist location in this referential and to measure a rotation. 93

4.27 The process computes limb flexion followed by limb realignment. Once the flexion is determined and the realignment applied, the twist is adjusted. For a large eccentricity of the intermediate joint reprojected position from the anchor effector axis, the swivel angle is determined by the radial direction of the retro-projected joint (not necessarily on the same disc as the current bone's right). Conversely, when the retro projection of the intermediate joint is close to the effector anchor axis, the swivel retained is the one from the source skeleton to avoid instabilities in the animation. In between, a linear interpolation is performed. 95

4.28 This figure shows a set of poses in pairs of two rows, with the upper row representing animations using forward kinematics and the lower row (the one with the source user skeleton model on the left) representing the retargeted animations. 96

4.29 Example demonstrating a missed ground contact due to the target point for the right ankle being out of reach. 98

5.1 Example of semantic poses conveying different types of interactions 102

5.2 The participants were seated comfortably in front of a large screen that displayed the videos. They were given the option to take breaks, and to avoid any potential issues with light reflections, the room lights were turned off (unlike what is depicted in the picture). 104

5.3 Screenshot of the displayed interface to the user on the full right: the original action performed by a person, on the two middle and right: the animated avatar using either the approach with the retargeting pipeline enabled or with only raw angles provided by the user's skeleton input. The order between the two animation methods is randomized, hence unknown to the participant. Two continuous sliders are displayed below the videos to allow participants for individual video evaluations. Finally, the participant can replay the videos as much as they want and validate their choice using buttons from the interface. 105
5.4 Normalized boxplots of the participants' evaluation scores. 107

B.1 The CPR mannequin BraydenManikin (2019) is tracked through the HTC-Vive tracker mounted on the wooden support to align the virtual victim's body with the mannequin. The same type of tracker is attached to the dominant hand. Finally, an HTC controller is only used to launch the application. 120
B.2 Cardiopulmonary Resuscitation training in VR: setup with the tracked mannequin device and tracked hand (left) and 1PP view with performance feedback provided in the HMD (right) 121
B.3 Hand height and allowed interaction area are deduced from the tracker's data . 124
B.4 Conceptual illustration of the frequency count without (a) and with (b) a hysteresis filter 124
B.5 Calibration of the internal probe covering the whole set of possible compression for this mannequin 126
B.6 System architecture 127
B.7 Illustration of the pre-processing applied on the raw data from the analysis of the results of the experiment 128
B.8 Illustration of artifacts on the tracker curve 129
B.9 Hand-torso interaction without (left) and with the simplified geometric deformation (right). Note that the displayed hand is on top of the (undisplayed) other hand for performing the cardiac massage. See the video for the first PV. 130
B.10 Experimental main timeline and conditions 132
B.11 Distributions of normalized scores from questionnaires (labels from Figure B.10b) 134
B.12 Plot of scores obtained by group (labels from Figure B.10b). Horizontal green lines represent the targeted value for the best CPR, and noches represent the 95% confidence interval of the median (the red line, dashed blue one represents the mean). 135

List of Tables

2.1 Full list of layers in the marker regressor. 31
2.2 Full list of layers in the joint regressor. 33
2.3 Evaluation of the error on the marker neural network (error units=centimeters, lower=better). Legend: **LK** last known position, **MA** moving average, **AC** affine combinations, **NN** neural network. 38
2.4 Evaluation of the error on the final joint positions (error units=centimeters, lower=better). Legend: **IK** inverse kinematics, **FT** fine-tuned, **NN** neural networks. 39

3.1 Experimental conditions: Hatched areas indicate which finger is (sometimes wrongly) moved by subjects while cyan areas indicate the finger which is actually animated by the system. Illustrations represent an example case when a subject should move the index to validate the button (in the game, buttons can arrive on any vertical line). The arrow in the swap conditions represents the swap count. Here its value is one as swapped fingers are next to each other. 52
3.2 Detailed count of each case occurrence. The No Swap cases (blue) do not count for the experimental condition evaluation as they represent the large majority of ’normal’ events when playing the game. The `pp` columns represent the sub pedal pressed count per condition while the % `pp` represents its percentage share. In total, on the experimental conditions, 24% of the swaps were noticed with a pedal press (sum of `pp` over totals from red and green cells from the last line). . 55
3.3 Predictors values for the fitted mixed model. We can observe the very significant impact of the `se` predictor on the observed outcome. 55

4.1 Performance comparison between parallelized and sequential pipeline processing for egocentric computation 89

5.1 Illustration of the targeted poses used for the evaluation 103
5.2 p-Values for Shapiro tests on evaluations scores; we can observe that all data were normally distributed 106
5.3 Test and measure results for the scores of the evaluations between the Direct Kinematic and the Retargeting evaluation scores 107

1 Introduction

Jaron Lanier is credited with introducing the term VR (Virtual Reality), which refers to a real-time simulation that creates the subjective illusion of a virtual world for the user. Lanier advocated that VR had the potential to let you "being you in the Virtual Reality" with a "virtual version of your body" allowing users to interact with virtual environments using their entire body. Back at the advent of VR, he collaborated with Thomas G. Zimmerman, with whom they built a hand gesture interface device that was not only able to acquire hands and fingers motions but also to provide haptic feedback to the users wearing the gloves. Those devices, the DataGlove™ and the Z-Glove™ (Zimmerman et al., 1986) were connected to a computer and were intended for object manipulation in 3D as, e.g., a clinical tool to evaluate hand function, a music controller, or a finger spelling interface. This consideration for the hands, fingers, and overall body tracking system was seen as a clothing required to be put on in order to enter VR (Lanier, 1988). Due to their technological limitations, limited accuracy in determining the rotation (Quam et al., 1989), and combined with high cost, data gloves, however, remained limited to specific application cases of VR.

Compared to gloves, hand-held controllers are typically simpler in design, more cost-effective, more robust, and provide greater accuracy and reliability when tracking positions and orientation in 3D space. Additionally, holding a controller provides a resting position for the hand, which can help prevent fatigue, whereas not holding an object while maintaining a virtual object in front of us can often lead to hand fatigue (Falcao et al., 2015). Furthermore, the metaphor presented by using controllers is quite similar to the computer mouse: the selection and movement of an object can be as straightforward as placing the controller near the object, pressing the trigger with firm click feedback, and releasing the object where desired. In spite of the lack of realism, controllers have inherited the benefits of the widespread familiarity with mouse usage, whereas the inaccuracies in finger tracking or in the detection of a grasp for virtual objects reduced the overall effectiveness of gloves compared to controllers (Boban et al., 2020).

Thanks to technological advancement in computer vision, allowing for real-time finger tracking, there has been a recent change resulting in shifting away from the use of controllers. Nowadays, with devices such as the Oculus Quest (Oculus, 2019), LeapMotion (LeapMotion, 2019) providing finger tracking capabilities, or Vive Trackers (Vive, 2022) capable of tracking user's limbs and hands, we can see a regain of interest for the original idea to providing users with direct hands and fingers interactions. This is an additional clue relating to the importance of the role of hands and fingers in human interaction with the real world and, most importantly, the lack induced using only controllers.

1.1 VR use and limitations

1.1.1 VR field of applications

The review from Radianti et al. (2020) describes the main trends in using VR for skill training applications. Among those fields, they identified the training for the military; the military was the leading actor in the development of VR at its dawn, First Responders (e.g., police officers, firefighters, and emergency medical services), Transportation, Workforce Training, Interpersonal Skills Training, and Medical Training. Simulation training in surgery is an excellent illustration of how VR technology allows the development of surgical skills in a controlled environment, reducing risks to patient safety, optimizing the use of operating theaters, and minimizing financial costs (Aggarwal et al., 2010). However, using simulations involves raising the question of the fidelity of the proposed experience and its influence on the quality of the training. Consequently, it was shown that the level of realism and fidelity in the training experience would enhance skills acquisition compared to low-fidelity systems (Sidhu et al., 2007).

In 1999, Gallagher et al. investigated ways to address simulation training for laparoscopy. This type of operation involves placing a tool equipped with a camera at its tip in a patient's abdomen. Here, a hardware simulator (MIST VR simulator) was replicating the real tool used during the surgery, and the goal was to train the fulcrum effect of the abdominal wall on the manageability of the instrument with inexperienced subjects (Gallagher et al., 1999; Seymour et al., 2002). With advancements in the level of immersion, the metaphor of controllers heavily contributed to the reduction of the hardware costs, and new elements were introduced to enhance the realism and the involvement of trainees. For instance, Papagiannakis et al. (2018) introduced an environment that is easily and broadly accessible, including scenarios, tracked virtual characters, and interactive 3D medical simulation training, hence relating the importance of the whole environment rather than not only focusing on the surgery task in itself. Furthermore, tracking users' full-body movements while displaying a plausible scenario within a virtual world might help participants behave as if they were experiencing the actual situation (Manganas et al., 2005). In line with this vision, Pfeifer and Bongard (2006) argued that "the body is required for intelligence" (intelligence in the sense of the ability to think). Computed-mediated interactions evolved from the traditional desktop metaphor to integrate embodied interaction (Ullmer et al., 2022) (i.e., the interaction involves the user's virtual body) as a powerful means to achieve new classes of tasks leveraging our full-body synergies and skills (Dourish, 2001).

1.1.2 Training in VR with tangible haptic elements: When controllers become a limitation

In this context of tangible applications with objects (applications with the integration of real elements from the real world into the VE), many interactions may occur and produce haptic feedback while using hands to interact with tools, systems, or bodies. Thus, it becomes impossible to use regular controllers as those would hit the tangible elements and prevent hand interactions.

In the study Delahaye et al. (2021) (Appendix A), we investigated the importance of providing tangible haptic feedback on the quality of CPR (Cardiopulmonary Resuscitation) training in VR. Providing CPR simulation in VR is useful for training individuals to react correctly to stressful situations, e.g., an emergency requiring to perform first aid in case of sudden cardiac arrest (Lemaire, 2018). To that aim, we developed a VR scenario with a physical dummy mannequin (BraydenManikin, 2019) equipped with an electric probe (only used for control) that was successfully integrated into the VE using a tracker to locate it in the VE where a virtual body was laying down on the floor at the same location Figure 1.1.

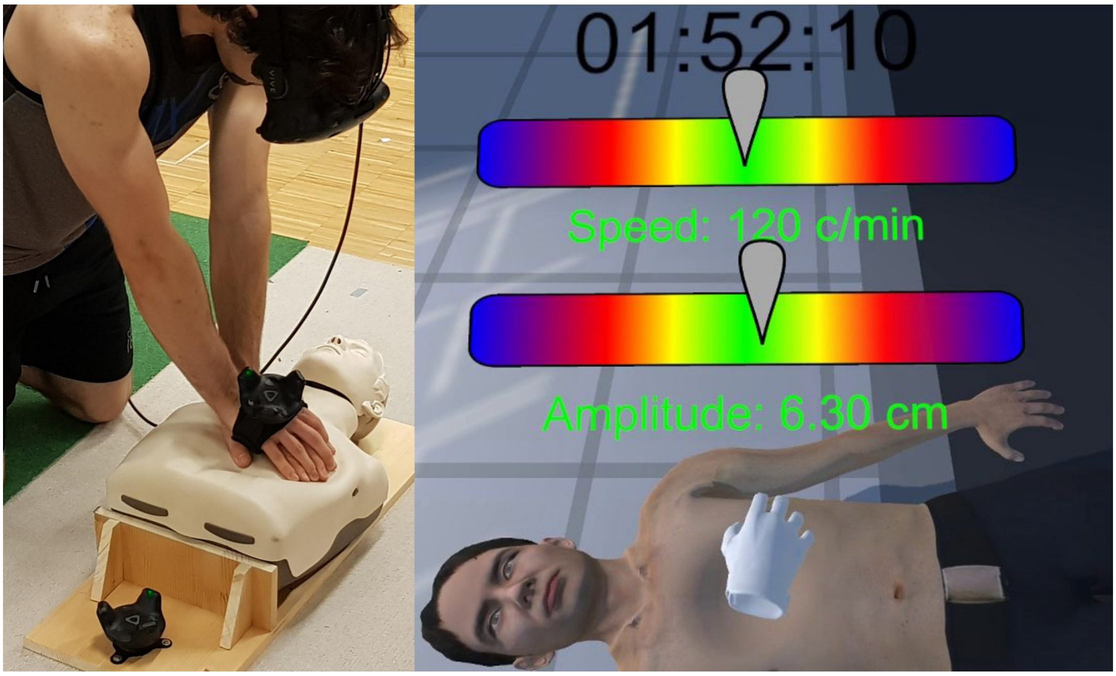


Figure 1.1 – *Cardiopulmonary Resuscitation training in VR: setup with the tracked mannequin device and tracked hand (left) and first PV (Person Viewpoint) with performance feedback provided in the HMD (right)*

Due to the presence of a tangible haptic surface (the dummy), controllers were replaced with a single tracker. The minimal immersive setup used was chosen for its previously proven sufficiency at eliciting presence (Cummings and Bailenson, 2016). The presence of the dummy mannequin providing a tangible haptic response was a factor shown to significantly increase the quality of the amplitude target to perform the correct movement. It was concluded that the sole visual immersion is insufficient for the correct skill training and relating the importance of the haptic component to be provided, hence the limitation of using controllers in such training.

1.2 Immersion, presence, and embodiment in VR (Virtual Reality)

1.2.1 Immersion: the characteristics of the display to the Virtual Environment

Before discussing interactions within a VE (Virtual Environment), one must first allow users to perceive this virtual world. An interface device is necessary to display a virtual world to the user, and a wide range of devices and configurations can immerse users in VEs. For instance, in the past, when the DataGlove ™was released, a typical interface that users could use was a regular 2D screen, as shown in Figure 1.2.

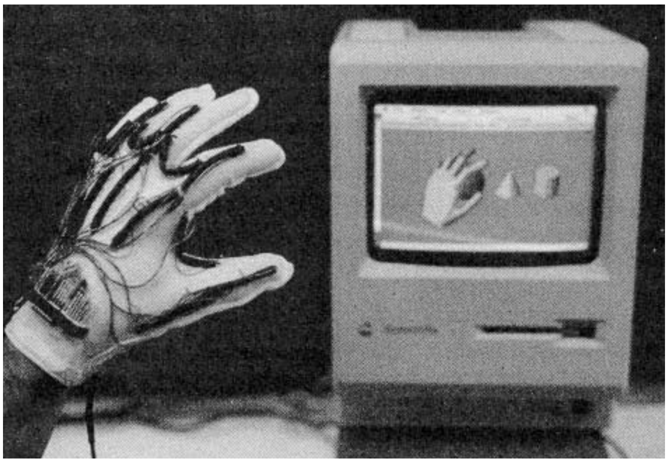


Figure 1.2 – *Illustration of the DataGlove ™from Zimmerman et al. (1986) used to move 3D objects in a virtual scene.*

The characteristics of these interfaces determine the level of immersion, such as the "visual fidelity" of the devices in rendering a VE (Sanchez-Vives and Slater, 2005). These characteristics include the frame rate of the screens in the device, its field of view (FoV), and color fidelity, among others. These differences in characteristics provide a continuum of the level of immersion, known as Milgram's Virtuality Continuum (Milgram and Kishino, 1994), illustrated in Figure 1.3.

For instance, when Ivan Sutherland designed the first HMD (Head-Mounted Display) in 1965 (Sutherland et al., 1965), the display could only output monochromatic lines, resulting in a lower level of immersion compared to modern HMDs that offer high refresh rates and a larger field of view. In that way, a static display is less immersive than a configuration of a set of displays (e.g., (Manjrekar et al., 2014)), themselves, less immersive than HMD.

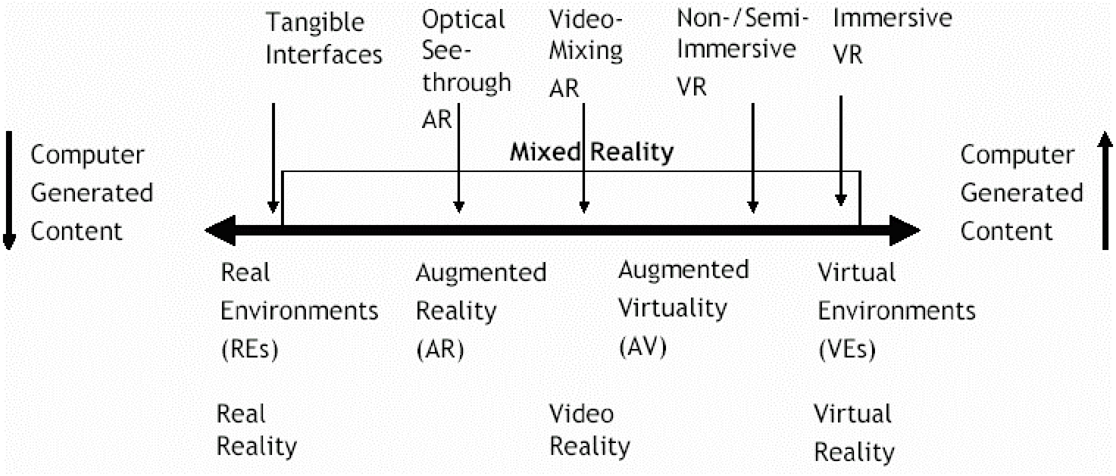


Figure 1.3 – *Milgram's Virtuality continuum with on the left the real world as we perceive it, and on the full right the immersive VR where the user can believe he is in another world. Illustration adapted schema from Zlatanova (2002)*

1.2.2 Presence

In reaction to the technical immersion provided by the system, one may experience the subjective feeling of "being there" in the virtual world. When such a feeling occurs, we talk about presence illusion, also commonly shortened as presence. This term was progressively anchored in the literature by Slater et al. (Slater and Wilbur, 1997). It is widely accepted in the scientific literature (Heeter, 1999; Slater, 2003) and can be measured through questionnaires (Schwind et al., 2019). When this feeling is disrupted, we talk about a BiP (Break in Presence) (Slater and Steed, 2000). Common factors leading to those disruptions are breaks in the PI (Place Illusion) (defined as "the illusion of being in the place depicted by the VR") or breaks in the PSI (Plausibility Illusion) (defined as "the illusion that the virtual situations and events are really happening"). These definitions were introduced to extend the definitions of presence (Slater et al., 2009). For instance, those breaks can occur when a glitch moves us through the boundary walls and show a sky box that was not intended to be seen by a player. It is consequently crucial to avoid such non-plausible situations to maintain the Place Illusion.

1.2.3 The SoE (Sense of Embodiment) and the role of the body

The construction of our body is deeply engraved into us; it serves as an anchor that connects us to the environment and enables us to interact with it (Slater et al., 2022). In particular, a prominent way to interact with the real world is through our hands and fingers.

In the case of fully immersive devices, such as HMDs, the headset hides the real world and the user's body, only to allow the user to see what is displayed on the virtual screens. If the body of the user is not represented in the scene, a conflict occurs between the user's expectation of having a body and the absence of a body in the virtual environment, negatively affecting the user's experience (Porssut et al., 2019; Gao et al., 2020).

Self-Location, body ownership and Sense of Agency

To solve the conflict arising from the lack of the body, it is required that the system provides the user with a virtual body toward which, provided that some criteria are met, the user could experience a strong SoE (Sense of Embodiment). According to Kilteni et al. (2012), the SoE requires the conjunction of three components: the sense of self-location ("refers to one's spatial experience of being inside a body" (Kilteni et al., 2012)), the sense of body ownership ("refers to one's self-attribution of a body" (Kilteni et al., 2012)) and finally, the SoA (Sense of Agency) (the "global motor control, including the subjective experience of action, control, intention, motor selection and the conscious experience of will" (Blanke and Metzinger, 2009)). Any disruption of any of the three components would be sufficient to induce a BiE (Break in Embodiment) (Kokkinara and Slater, 2014), therefore reducing the quality of user experience.

Body ownership The "rubber hand illusion" experiment (Botvinick and Cohen, 1998) and its virtual counterpart (Slater and Wilbur, 1997) provide a clear example of what body ownership is. In those experiments, researchers investigated the multi-sensory integration of bodily perception by placing a dummy/virtual limb nearby the location of the real limb of the user. Despite knowing that the limb is not their own limb, participants experienced a sense of ownership toward the virtual limb when it is stimulated synchronously with a sequence of successive strokes on both the real and virtual limb. This subjective experience is measured with questionnaires, and the questionnaire proposed by Gonzalez-Franco (Gonzalez-Franco and Peck, 2018), inspired by this experiment, aims to provide a standardized version to allow balanced comparison between studies.

Self-location According to Kilteni et al., "Self-location is a determinate volume in space where one feels to be located" (Kilteni et al., 2012). This subjective experience is usually perceived within the limits of a physical body Blanke (2012); however, this sense could be manipulated through experimental setup or illness, resulting in the out-of-body experience Blanke and Mohr (2005). Under normal circumstances, the full-body illusion in immersive VR was shown to be higher at the first-person viewpoint Galvan Debarba et al. (2017). This subjective experience is commonly measured using questionnaires, but more recent approaches can use a mental imagery task to measure changes in self-location (Nakul et al., 2020).

Sense of Agency When one performs a voluntary movement, the brain makes a copy of the planned movement (known as the efference copy) and compares it to the actual movement observed (namely, the afferent copy). This is known as the comparator model (Wolpert et al., 1995), and the comparison result is useful to adjust the current intended movement. If both copies yield similar information, one will self-attribute the authorship of the performed action (Jeannerod, 2003; Blakemore et al., 2000): the SoA (Sense of Agency). Conversely, here, if the arm remains static while the real arm moves, there is a significant difference between both copies, and a loss of SoA can occur (Engbert et al., 2008; Jeannerod, 2009b). This is even stronger when the discrepancy appears suddenly, in which case a violation of agency can be observed (Haggard, 2017; Jeannerod, 2009a). Among the factors that influence agency, latency beyond hundreds of milliseconds was shown to significantly reduce the SoA (Farrer et al., 2008; Wen, 2019). Formulated differently, the SoA is the subjective feeling that one is controlling his body and that the body reacts reasonably quickly to his commands.

Within this framework, the subject of this thesis aims to provide to the users, regardless of their morphology or the morphology of the avatar, an animated avatar respecting their movements down to the finger level Figure 1.4.

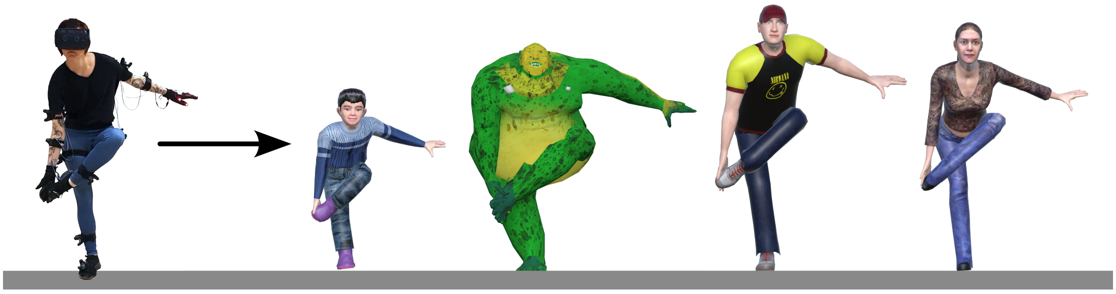


Figure 1.4 – *On the left is the source position of the user, and on the right, the pose retargeted on different avatars produced respecting the self-contact with the body.*

1.3 Providing a virtual body in immersive VR

Providing a virtual body to a user leverages the user's immersion level, but this does not necessarily imply that the user will embody the avatar. To allow someone to embody an avatar, we previously observed in particular that the avatar's body should be co-located with the user's body and react to the user's movement in real-time.

One way to achieve this consists of 3D scanning in real-time the user's body and displaying the scan in the VE (Albert et al., 2019). This technique is straightforward and very effective, but it does not provide a direct representation of the skeleton structure. The difference between providing the streamed surface and knowing the internal structure is the same as the difference between a PNG image (array of pixels) and an SVG image (vectorial description of the image): in one case, the visible external shape is displayed to the user, while in the second case, the structure is first rendered using an avatar, and then displayed to the user, hence allowing automated manipulations of the output based on the modification of the underlying structure, while in the first case, all manipulations must be done manually. Furthermore, having a body scan is likely to induce a different level of detail compared to the one present in the VE, or the light exposure might differ from the one from the scene, for instance.

1.3.1 The virtual body as an animated 3D character

To ensure uniformity in the rendering and to support interactions with the VE, it is common to rig 3D models of characters. Those are structures composed of a skeleton and a mesh. The mesh is a 3D structure rendered and displayed to the user by the rendering pipelines of game engines. The mesh is attached to the skeleton through the process of rigging. This process corresponds to attaching vertices (points constituting the mesh) to the bones of the skeleton. Usually, vertices can be attached to up to four different bones at the same time. The rigging is usually performed only once when designing the avatar.

3D models can be manually drawn or scanned using pictures of the different sides of the body with cameras, controlled lighting, and triangulation.

In the case of a 3D scan, the texture can be directly applied to the created mesh, and pseudo-automation can accelerate the rigging of the avatar (Shapiro et al., 2014; Feng et al., 2015; Baran and Popovi, 2007). Using a scan provides the advantage that the generated model corresponds to the one from the user in terms of dimension, appearance, or skin details (vein location, mole, tattoos, skin tone), on which the user can rely to recognize their own body. As a result, in terms of animation, the user's skeleton motion can mostly be directly applied to the avatar with a limited risk that the final avatar's pose would present animation errors due to a difference in limb lengths, for instance. However, in most situations, VR users are not offered the choice to have their body scanned and therefore have to use or choose an avatar among a pre-defined set of characters, and users can have the body of someone else (Banakou et al., 2018; Osimo et al., 2015), change their body size (Banakou et al., 2013), their skin tone (Maister et al., 2015), their gender (Neyret et al., 2020) or even have a supplementary limb (Steptoe et al., 2013; Hoyet et al., 2016).

When the user's morphology differs from the avatar's in terms of volume and/or proportions, it is no longer possible to remap the raw movements from the user directly. Let's assume one is placing the hand on the belly and that one's body and avatar share the same skeleton and morphology. Applying the same angles from the raw user's skeleton onto the avatar will result in the same skeleton pose, which, given the same morphology, will ultimately result in the same self-contacts between the hand and the belly. Now, from this state, let's progressively increase the belly size of the avatar without changing anything else. Then, the avatar's belly will progressively overlap, hiding the virtual's hand. Such a phenomenon is so-called inter-penetrations. Similarly, in the opposite direction, this may result in gaps that are known to break the SoE (Bovet et al., 2018).

The computer must first acquire the user's motion through MoCap (Motion Capture) to generate the avatar's animation. Those systems are interfaces allowing to capture the motion of objects or living beings. Different methods exist to achieve this goal.

1.3.2 Motion Capture technologies

Mechanical capture uses physical devices to capture the motion. The DataGlove ™from Zimmerman et al. (1986), the Manus VR (Manus-VR, 2018) or VR Free (Sensoryx, 2019) are devices that illustrate this for the tracking of local finger's motion in the hand's referential. Full body tracking can also be performed with this technique; however, those devices can be bulky and suffer poor accuracy, as this was the case for the Datasuit from (Sturman and Zeltzer, 1994). By construction, such devices provide consistent output over time. The measurement is often performed over the modulation of flexible PCBs (circuit boards, here used as gauges) resistance, the time required for the light to pass through a fiber, or through mechanical measurement of angles using potentiometers/coders attached to an armature, which are sensors providing continuity in their measurements. Due to the nature of resisting materials (potentiometers or gauges), those devices are often subject to drifts and might also present a sensitivity to the heat in the measurement. Consequently, those devices should commonly be calibrated to get accurate output.

IMU uses a set of multiple accelerometers attached to critical locations of the structure to be tracked. The Perception Neuron suit (NeuronMocap, 2018) is a typical example of this technology applied to full body tracking. Accelerations only provide relative information on the movement of the accelerometer; a null acceleration could either be attributed to a continuous displacement at a consistent speed and direction or as a static position in the world coordinates. Hence, to be able to locate a position, calibration is first required, and then the acceleration measurements are accumulated to compute the traveled displacement since the calibration point. Due to the nature of the integration of measurements, a drift easily occurs over time, making the measurements no longer viable (Tian et al., 2015), despite the fact that filters, like the Kalman filter (Kalman, 1960), are present to reduce the drift.

Proximity Sensors is a technique using distance sensors to locate the proximity of an object. Valve uses this technique in the Valve Index controllers (Valve, 2019) to give users finger-tracking capabilities. It is, however, less common to use it to measure the full body motion, and a controller has to be held at hand for the motion to be recorded, which can become an issue.

Electromagnetic sensors works by using sensors to detect changes in magnetic fields generated by an electromagnetic field generator. These sensors provide the position and the orientation of the object being tracked, as this was the case for the DataGlove ™. As it was observed in Bodenheimer et al. (1997); Molet et al. (1999), this technology suffers from a high sensibility to electromagnetic noises, while their range of action may be limited, making it impractical to be used for getting reliable positions.

Markerless optical tracking does not require the user to wear a specific device to acquire its motion and instead relies on external cameras. It is, as a consequence, more convenient for the users but comes at the cost of a sensibility to occlusions, a phenomenon occurring when the system cannot see the tracked item. The Kinect (Microsoft, 2019) (which also embeds a depth camera) was probably the most famous example of the implementation of this technique before the recent arrival of the Oculus Quest (Oculus, 2019) that is a standalone HMD that directly embeds the cameras on it. This technology can easily be mixed with prior knowledge, such as a silhouette, or trained models, to track users' movements (Ballan and Cortelazzo, 2008; Mathis et al., 2018), such as fingers, as this is the case for both the Oculus Quest and Leap Motion (LeapMotion, 2019). However, Shao (2016) showed that the LeapMotion, at that time (the Oculus was not released yet), had trouble dealing with two hands when those were close to each other. An advantage of this technology is that it can be applied through transfer learning to objects on which it is difficult to attach trackers, such as mice (Yosinski et al., 2014; Insafutdinov et al., 2016). It is still observed that this technology struggles to acquire fast movements in real-time, such as finger pinching (Li et al., 2022).

Passive optical tracking is a solution that also uses external cameras to acquire motion. The difference is that some markers (usually made of reflective materials for infrared) are placed on the structure to be tracked. This enhances the quality of the tracking in both refresh rate and precision, but there, only the marker's positions are captured; there is no overall knowledge of the structure status that could be used to recover the input in case of an occlusion occurring. One of the main actors in this field is Vicon, which also sells software for their technology that provides a model of full body skeleton tracked, that also includes fingers (Vicon, 2019).

Active optical tracking is similar to passive optical tracking; however, the reflective markers are replaced with active LEDs. On the one hand, this requires cables to be placed on the user, while on the other hand, this allows the system to put a unique ID to each of the markers, which cannot be mixed if one is about to disappear and reappear later on. This technology is commonly used in scientific research (Holden, 2018; Aristidou and Lasenby, 2013; Herda et al., 2000) It is to be noted that this remains insufficient to alleviate the missing data induced by the occlusions.

Phasespace Inc. implements this technology in their Impulse X2 (PhaseSpace, 2019) and provides an acquisition speed of up to 480Hz with an accuracy in the millimeter range.

Overall, none of the technologies presented here can be used without a drawback, and the perfect tracking system does not exist yet, and some mitigation techniques must be employed to track movements reliably.

1.3.3 Mitigation techniques for unreliable MoCap input

To mitigate the risk of tracking losses, a combination of several technologies with different weaknesses can be used to enhance the tracking reliability of the system. For instance, an IMU system suffering from drift can be coupled with an absolute system suffering from occlusion (Tian et al., 2015). By coupling technologies suffering from different issues, the input source can alternate based on the available information, which can be used to re-calibrate the other system failing to provide the position: The IMU drift is then corrected each time there is no occlusion. This is typically used in the Vive Trackers (Vive, 2022) that embeds a large set of active LEDs combined with internal IMUs to mitigate the risk of losing the tracking. Furthermore, the numerous LEDs create a referential from which the tracker's orientation can be retrieved, constituting valuable information for the animation. However, those devices are a bit bulky and cannot be used to track finger movements.

To recover the missing information due to occlusions on optical tracking solutions, one can interpolate the missing positions from the last known position and the position from the first frame where the tracker came back (Wiley and Hahn, 1997; Rose et al., 1998; Nebel, 1999). This provides good precision on the recovered points, but this cannot be used in real-time as we don't know in advance the position of the marker when the occlusion stops. To make the approach real-time compliant, a prediction on the missing position needs to be made. Predictions can be extrapolations of a mix between linear and circular motion (Piazza et al., 2009), but the observed coherence of the predicted markers' positions difficulty goes over 150ms. In their work Li et al. (2010); Herda et al. (2000), the authors took advantage of the human skeleton to infer constraints into a model used to improve the prediction quality. In Li et al. (2010), Junlei Li et al. use the length of the bone to define hard and soft constraints and deal with black-outs (when many markers are occluded at the same time) by extrapolating the current moving trend. Soft constraints allow the system to deal with the fact that markers slightly move in the joint referential; thus, the soft constraints can be violated to improve the quality of pose reconstruction. Aristidou et al. also exploited the fact that the distance between markers on the same segment is approximately constant (rigid body) to estimate in real-time the joint CoR (Center of Rotation) (Aristidou et al., 2008) and used a constant velocity model in the Kalman filter (Kalman, 1960) to predict the occluded state. Observed results show that this approach can run up to 350 frames per second and yield an error on the CoR position of approximately 6.5mm with one missing marker out of three or 9mm with two missing markers after 500 frames with occlusions. This was even more enhanced in Aristidou and Lasenby (2013), where the pipeline was improved using partial camera information when a marker is seen from only one camera. Those results show that occlusions do not necessarily lead to the impossibility of animating an avatar.

1.3.4 From MoCap data to the animation of a 3D model with inverse kinematics (IK)

Having all the markers' positions is already a challenge, but this is insufficient to animate a rigged skeleton of a 3D avatar with many DoF (Degree of Freedom). The inputs from the MoCap need to be processed, considering the kinematic description of the joints and the position of the markers to generate the joints' rotations. In the engineering field and industry, this is a common problem for the control of robotic arms known as IK (Inverse Kinematic), but the solutions proposed are mostly transferable to the domain of animation. It is to be noted, though, that the whole human skeleton presents the singularity of having a complex structure composed of 206 bones (Kamina, 2009), and some techniques might struggle to be real-time compliant. A good description of the state of the art of IK is provided in Aristidou et al. (2017), highlighting the pros and cons of the different methods. Among them, one of the most established methods is using Jacobians to reverse the kinematic chain thanks to its ability to easily handle the high number of DoFs of animated avatars (usually around 70) while highlighting the risk of having some instabilities. It is also described how dampening the least squares helps stabilize the pseudo-inverse kinematic solution in the neighborhood of singularities (Baerlocher and Boulic, 2004) at the cost of a slower convergence rate, which could be palliated using a GPU implementation to obtain the best damping factor (Harish et al., 2016). With the rise of machine learning, the review Aristidou et al. (2017) highlights the new trend for data-driven IKs. Robotics arms were successfully controlled using neural networks (Waegeman and Schrauwen, 2011; Hasan et al., 2010; Das and Deb, 2016; Vladimirov and Koceski, 2019), with sometime millimeter error precision for simple kinematic structure with only two degrees of freedom (Vladimirov and Koceski, 2019). One of the pros of neural networks is their ability to take extra input to improve their output, as this was the case in Das and Deb (2016), where the input was extended with the current joint rotations, whereas, in the traditional methods, only the targeted position was given as an input of the model trained for the specific kinematic chain. Some solutions are also designed on purpose for character animation, such as for the spine (Unzueta et al., 2008) or the hands (Aristidou et al., 2017; Aristidou, 2018; Kim, 2014). It is noted that some simplifications can also be made in the structural representation of the human body. For instance, hands can be reasonably represented with a structure containing only 24 DoFs as a good compromise (Cobos et al., 2008).

1.3.5 Retargeting: the adaptation of one's body pose to a different virtual body

Once the user's skeleton model is animated, the final step is to apply the animation to the target character. When the user shares the same skeleton structure, volume, and proportions as the animated avatar, the mapping is mostly the direct application of the skeleton model onto the avatar. However, when any of these characteristics differ, applying the raw angles from the model to the avatar might lead to gaps when there should be contacts or interpenetrations when there should be self-contacts, thus hampering the SoE for the user (Bovet et al., 2018). Additionally, it was observed that differences in self-contacts are considered as different poses by third-person viewpoint observers in most cases, this being even stronger for hand contacts compared to arms contact (Basset et al., 2022) relating the importance of hands in the animation.

In consequence, an additional process is required to transform the original motion to adapt it to the destination avatar when a difference exists in morphologies/skeleton between the user and the 3D character. This process is called retargeting or sometimes remapping, and the literature offers a large panel of techniques on this topic (Guo et al., 2015; Mourot et al., 2022b).

In the seminal work from Gleicher (1998), the authors considered the discrepancy that can occur from the difference in limb lengths; however, the approach required some pre-processing where constraints needed to be specified to get the correct animation. Furthermore, the body surface was not considered in the approach; hence the self-interactions with the body were not addressed, leading to possible interpenetrations. In their approach Choi and Ko (2000), Choi et al. investigated an online approach, without the need for constraints to be set, to retarget the animation from one character to another. However, this method relying on a closed-loop control was shown to present some instabilities near singularities. Shin et al. also proposed a real-time approach used in television to animate characters, which had an essential role in preserving the semantics of the posture (Shin et al., 2001); however, as for the work from Gleicher et al. or Choi et al., those method does not address changes in morphology other than limb length (e.g., large belly vs. small belly with the same skeleton)

A higher level abstract method arose with the work from Kulpa et al. (2005) in which the authors used an internal representation of the structure of the limbs to adapt a limb motion onto another character's motion easily. This representation represents a limb as a combination of a half-plane (whose origin is the root's joint of the limb, its leading axis, the axis passing by the root and the effector's joint, and the second axis orienting the plan so that the intermediate joint is contained), and the normalized distance, concerning the bones length, between the effector and the root position. The authors also allowed the user to add constraints such as orientation, exclusions area, position, or distance constraint to ensure self-contact when clapping hands. Special care was provided to address the foot contact with the floor surface. Given the nature of the implementation, their approach can be used with elements that can evolve on-the-fly (e.g., the floor can bend, the limb can shrink or extend, etc.). The approach for the constraints was then layered in Multon et al. (2009) and featured an additional response to external forces applied by the environment. However, not all of the constraints are known in advance, and this work focuses more on the interaction with the rest of the environment than on the interaction with the body itself. Al-Asqhar et al. introduced in Al-Asqhar et al. (2013) an approach based on surface descriptor to tackle the problem of maintaining contact congruency with close proximity of the skeleton with mesh surfaces. Mesh surfaces are discretized in triangles, and contribution weights are computed for the vectors separating the body joints and their projection on the discretized mesh. Then, those vectors are iteratively reapplied on the avatar as a sum of forces to adjust the positions of the body joints. In their work Molla et al. (2017), Molla et al. combined this approach with elements from Kulpa et al. (2005); Multon et al. (2009) to provide a body-independent retargeting animation pipeline that can handle self-contact congruency in real-time.

More recently, several new approaches were presented with the new rise of machine learning. In the method from Celikcan et al. (Celikcan et al., 2015), equivalent poses between the user and the avatar needed to be calibrated to train the correspondence of posses, and the animation of the avatar was performed through a mesh deformation rather than applying rotation on a skeleton. This is particularly useful for facial animation or motion retargeting (Zhang et al., 2022), where the animation through a skeleton is complex; it broadens the method's applicability on a larger set of non-rigged avatars but drops the internal structure representation. Mesh deformation was also investigated in Basset et al. (2020) to produce convincing avatar poses robust to body shape differences; however, the proposed approach is not real-time compliant making it impossible to be used in VR for avatar animation. Machine learning was also applied for rigged virtual character motion retargeting with approaches such as the ones from Villegas et al. (2018); Aberman et al. (2020) using neural networks to animate target avatars; however, only the skeleton is considered; hence the difference in morphologies remains not fully covered.

Overall, to address the issue of the importance of self-contact congruence between the users' movement and the avatars (Bovet et al., 2018; Basset et al., 2022), an additional process must be put in place to transform the modelized source motion of a person into the avatar's motion. We observed that some approaches address interactions with the ground (Shin et al., 2001; Gleicher, 1998; Kulpa et al., 2005; Multon et al., 2009), other interactions with objects (Kim and Park, 2016) or self-contact with the body interactions (Molla et al., 2017); however, none of these methods appear to address finger-level interactions, which play a crucial role in interacting with the virtual world.

1.4 Research plan

We interact with the world through our body, hands, and fingers. In an immersive Virtual Environment, this physical body is no longer visible to the user, which conflicts with the expectation to see our body. Consequently, presenting a virtual body, called an avatar, is an important step to improve the user experience in VR. When the avatar is a 3D character, its skeleton and morphology might differ from the user, and the direct application of the captured user motion on the avatar's skeleton might induce self-contact conflicts. This is why it is essential to edit on-the-fly the animation to be applied on the avatar to prevent mismatches that would break the embodiment. Currently, the literature only addresses part of this problem separately, but without addressing the subjective user experience, especially regarding providing both finger-tracking capabilities and self-contact congruency in real-time. Therefore, This thesis focuses on providing the user with a real-time animated avatar, with both finger and body-level animation, whose morphology might differ from the user, while placing the accent on the subjective experience of the user.

In order to accomplish this objective, we followed the following research plan:

- First, in line with our observations on the limitation of controllers and rigid hands in Delahaye et al. (2021), we aimed to provide a way to animate hands and fingers in real-time. Rather than prioritizing the pursuit of a solution that is exact and accurate under occlusions, we designed a system focusing on producing plausible hands and finger poses learned from a recorded dataset in chapter 2.
- Then, we performed a user study that examined the thresholds of human motion perception and embodiment at the finger level in chapter 3. This characterization was investigated through an experimental paradigm involving a finger-based task in which participants had to validate buttons using only their fingers. During the game task, the machine introduced finger swaps to correct or impede the user's actions, and the experimental task for participants was to press a pedal when noticing those introduced swaps.
- The observations from the outcome of the previous study, combined with the knowledge from recent literature on embodiment at the body level, highlighted an opportunity to introduce controlled distortions to ensure self-contact consistency at the finger level. Here, using the knowledge of the limits of embodiment, we developed an approach to provide users with a virtual body, not necessarily with the same morphology as the user. This approach took advantage of movement distortions to provide body animation, at both finger and body levels, and self-contact congruency (chapter 4).
- The contribution from this technique was evaluated against the direct forward kinematic animation, with various avatars presenting different morphologies and sizes in chapter 5.
- Finally, a synthesis of this thesis is discussed, and a conclusion is drawn from the observations in chapter 6.

2 Providing users with finger animation in VR

2.1 Introduction

VR is becoming increasingly popular owing to a new generation of affordable HMDs and machines to run and render VE in real-time. However, interactions with the environment can be challenging due to the overly simplified avatar representation leading to a sub-optimal experience, as observed in Delahaye et al. (2021). When present in the VE, the avatars used to be static meshes or animated characters that used to move according to predefined patterns or actions.

In recent years, trackers such as the Vive trackers Vive (2022) (relying on active optical tracking and on IMU to mitigate the risk of occlusions) have become accessible to the public, allowing MoCap to be integrated for consumer-grade oriented setups, hence, making it compelling for leveraging the level of immersion to the user: Thanks to IKs, such as the bundle FinalIK RootMotion (2020) regrouping different types of IK, it is now possible to have avatars animated by the captured user's movements at the body level. However, those device remains too bulky to be placed on fingers to allow them to track finger movements, and no real equivalent is offered to replace those for tracking finger movements.

With the speed limitations from computer vision (Li et al., 2022), the drift and loss of precision requiring regular re-calibrations of IMU and mechanical tracking (Sturman and Zeltzer, 1994; Tian et al., 2015), we investigated an approach using an active optical tracking solution. This tracking technique still suffers from occlusions, and increasing the number of cameras is not always feasible and does not solve the problem entirely. However, studies have shown that it is possible, to a certain extent, to recover and predict marker positions during those occlusions Piazza et al. (2009); Li et al. (2010); Herda et al. (2000); Aristidou et al. (2008); Aristidou and Lasenby (2013).

In this chapter, we discuss an approach using this MoCap technique in combination with neural networks to mitigate occlusions and provide IKs to animate hands and fingers in real-time. In the first part (section 2.2), we discuss the state-of-the-art and analyze how our method relates to other approaches. In subsection 2.3.1, we describe the dataset's features for training our model. In subsection 2.3.2, we recall three baseline methods for correcting occlusions, and we introduce a more complex model based on neural networks for handling both occlusions and inverse kinematics. The section 2.4 describes our experimental methodology while subsection 2.3.5 details the context of bimanual tracking. Finally, we present our results in subsection 2.4.2 before the concluding discussion.

2.2 Related work

Occlusion robustness The most common approach for correcting occlusions is to use interpolation algorithms. In this regard, some data-based interpolation techniques have been specifically designed for human body tracking and skeleton animation Wiley and Hahn (1997). However, interpolation algorithms require knowledge of past and future data and can only be applied in post-processing. More recently, denoising neural networks have been proposed for offline cleaning of motion capture data Holden (2018), producing results comparable to hand-cleaning.

Aristidou et al. proposed an approach based on Kalman filters for estimating the positions of occluded markers in real-time Aristidou et al. (2008). Their method does not require prior knowledge of the skeleton but assumes that the distance between neighboring markers is approximately constant. The algorithm builds a skeleton model by estimating the centers of rotation between two sets of points. When an occlusion occurs, the marker position is predicted using a Kalman filter, which considers velocity and the positions of neighboring markers. Piazza et al. developed a real-time extrapolation algorithm that assumes that motion can be either linear, circular, or a combination of both Piazza et al. (2009). As before, it does not rely on a predefined skeleton model. The prediction is performed through a moving average of the marker's velocity to minimize the effect of noise. An interesting optimization employed in this approach is the so-called constraint matrix (CM), which stores the minimum/maximum pairwise distances between all markers. At inference, the estimates are adjusted according to the constraints described in the CM. Both Aristidou et al. (2008) and Piazza et al. (2009) focus their work on limbs and do not address the particular case of fingers. Finally, a large portion of research in this field exploits the assumption that an underlying skeleton model is available, thereby allowing the algorithm to put some constraints on the solution. Recently Alexanderson et al. addressed the problem of labeling markers in a passive system for the fingers and the face Alexanderson et al. (2017). Instead of tracking markers in the temporal domain, it estimates the most likely assignments using Gaussian Mixture Models (GMMs). This allows fast recovery from occlusions and avoids the so-called *ghost markers*, i.e., detection of markers that do not exist. However, this approach does not address the problem of predicting the marker positions during occlusions.

Current real-time machine-learning-based approaches for handling occlusions are restricted to the sub-problem of posture and gesture recognition Mousas and Anagnostopoulos (2017). In our case, we do not perform such classification tasks; rather, we aim to achieve a complete reconstruction of the hand posture.

Tracking and reconstruction A common framework for capturing movement is to perform body or hand reconstruction through images and depth cameras. These approaches leverage computer vision and machine learning algorithms Moeslund et al. (2006) and aim at providing an affordable consumer-ready alternative to complex motion capture systems. Solutions focusing on hand and finger movements have made significant progress in tracking isolated hands in free space. These techniques are designed for the context of desktop-range interactions using specialized devices (a noteworthy example is the Leap Motion controller). When mounted on a HMD, these types of finger-tracking devices can offer an interesting compromise for immersive VR Rafferty et al. (2017). Nevertheless, their field of view is still limited when compared to the range of motion of the hands, and they present weaknesses when the hand palm is not facing the head of the user, thereby resulting in self-occlusions. Previous work has tried to address this problem. For instance, Tkach et al. fit the hand posture using a combination of sphere meshes Tkach et al. (2016) while Mueller et al. use a cascade of convolutional neural networks (CNNs) to first localize the hand center and then regress 3D joint locationsMueller et al. (2017). They also employ a synthesized dataset that simulates cluttered environments via a merged reality approach, allowing the model to generalize better. These approaches, however, are still very limited in terms of the range of motion as they are optimized for user-facing the camera.

As for tracking with motion capture, Han et al. frame the problem as a key point estimation task, which is tackled with CNNs Han et al. (2018). While their approach allows using a passive system, the authors highlight some shortcomings with multiple occlusions. Other frameworks based on motion capture typically employ some sort of sensor fusion from multiple data sources. Andrews et al. propose a tracking system that uses IMUs and a physics model to recover from sensor dropout Andrews et al. (2016). Our approach is related to Andrews et al. (2016) in the sense that we combine IMUs with motion capture to record a robust dataset, but differs in the fact that we use only motion capture at inference.

Machine learning for inverse kinematics As an improvement over existing techniques, Zhou et al. (2016b,a) proposed a deep learning framework in which a forward kinematics layer is added to a neural network to constrain the output to feasible postures. Specifically, Zhou et al. (2016b) focuses on hand pose estimation and Zhou et al. (2016a) on full-body pose estimation. As with the MS Kinect, these techniques rely on regular cameras and computer vision algorithms. As such, they are unable to exploit the potential that a full motion capture system has to offer in terms of both precision and range of motion.

A recent survey classifies inverse kinematics techniques into several categories, which can be summarized as “traditional” (analytic, Jacobian-based) and “data-driven”, often based on machine learning, and recently, on deep neural networks Aristidou et al. (2017). Most related work focuses on inverse kinematics in the most general setting, which consists in defining the desired positions of the end effectors and having the model find a configuration that achieves the desired result. This particular problem has already been tackled with machine learning in an industrial control setting, i.e., robot arm Almusawi et al. (2016); Waegeman and Schrauwen (2011), and in humanoid fingers Kim (2014).

2.3 Our Approach

We propose a compromise between skeleton-less methods (i.e., zero knowledge) and those with skeletons. Despite being closely related to *data-driven* inverse kinematics, it is more correctly referred to as “reconstruction”. We map the captured markers to the transformations of a virtual hand, but the end effectors do not need to be aligned with the corresponding markers. In fact, markers and joints belong to two different sets whose correlation is exploited by the model. Instead of defining constraints manually (as in IK systems), all the necessary information is automatically inferred from the data so as to obtain the most precise and naturally-looking prediction. Our work focuses on motion capture with active markers and proposes a machine-learning-based alternative to analytic IK algorithms, as well as a method for correcting occlusions whose pipeline is illustrated in Figure 2.1.

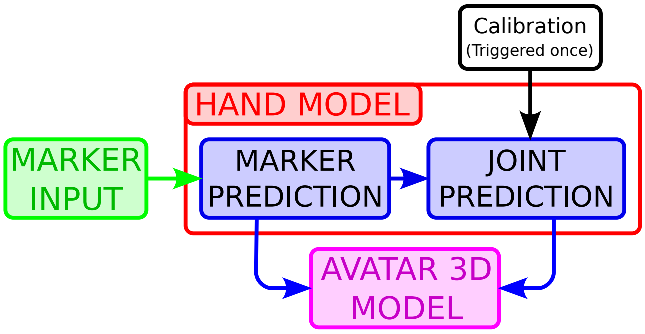


Figure 2.1 – *Conceptual full prediction pipeline.*

In this chapter, we show how we acquired a dataset from a number of subjects and devised an efficient two-stage pipeline that first corrects occlusions in the motion capture stream and then reconstructs all the transformations of the hand joints. Both stages are based on neural networks, which are trained on the aforementioned dataset. We evaluate our model at different levels: reconstruction error of occlusions, end-to-end reconstruction error of joint positions, and computational cost in terms of CPU and memory usage, crucial factors for real-time applications. The preliminary version of this approach Pavllo et al. (2018) has been extended as follows. First, we add a calibration process to increase the fidelity of the reconstruction of the model. Second, we handle bimanual occlusions in real-time. The system is also re-evaluated with a broader range of use cases.

2.3.1 Training dataset acquisition

We start by briefly introducing the characteristics of the motion capture pipeline and the final goal that our model is expected to meet. The user wears a pair of gloves equipped with LED markers. The positions of these markers are collected by a motion capture system and passed to the pipeline, which outputs the transformations necessary for animating the virtual hands in a VE. The mapping is depicted in Figure 2.2. More formally, the pipeline comprises the following inputs and outputs:

Inputs The absolute positions (i.e., 3D points) of the markers. Given that we employ an active motion capture system, each marker is tagged with its own unique ID. Some positions can be missing from the data stream if the corresponding markers are not visible from a minimum number of cameras, i.e., they are *occluded*.

Outputs The angles of each joint within the hand and the absolute position and orientation of the latter.

As mentioned, this mapping is learned from a dataset that we have collected for this specific task. The next section shows how we acquired the data and how we built our ground truth for the purposes of training and evaluating the model.

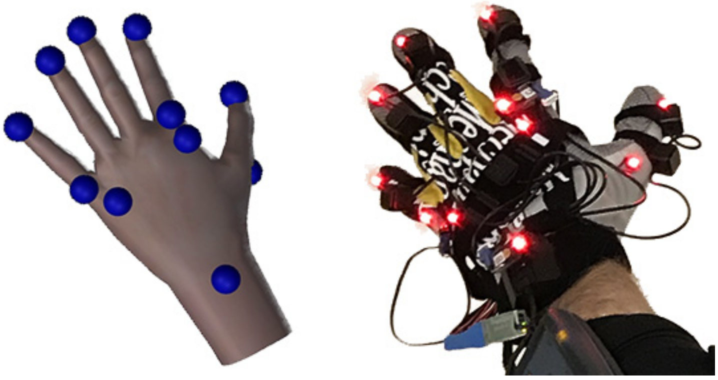


Figure 2.2 – *The virtual hand (left) and the mocap glove (right).*

Motion capture

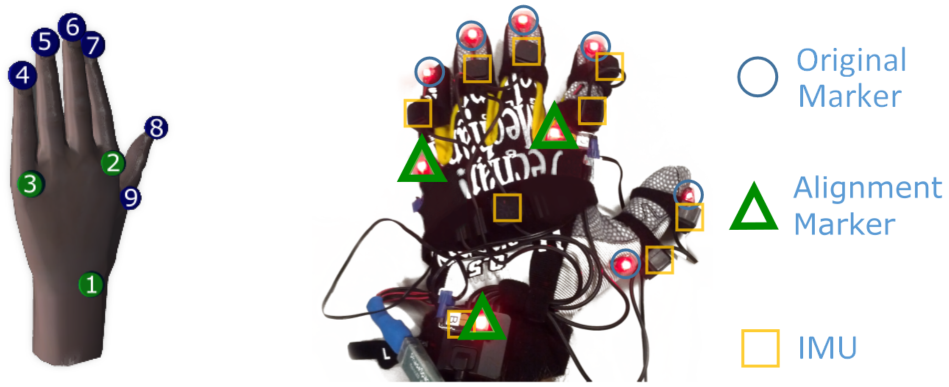
We acquired the data using two devices:

- PhaseSpace ImpulseX2 motion capture system, based on active LED markers. Each marker is tagged separately with a unique ID via a frequency modulation mechanism. This system can track the entire body by means of a suit equipped with markers.
- Noitom Perception Neuron, a low-cost hand-tracking device based on IMUs.

The two were combined on the custom glove shown in Figure 2.3 and used simultaneously for computing the ground truth through a sensor fusion algorithm, which we describe in Figure 2.3.1.

The original PhaseSpace glove, prior to customization, comprised eight markers (denoted as "Original markers" in Figure 2.3). The positions of some markers are not optimal, especially due to the lack of a marker at the wrist level, a crucial location for estimating the orientation of the hand. For this reason, we discarded two markers on the original glove and decided to add three additional markers (referred to as "Alignment Marker" in Figure 2.3). The secondary system (Perception Neuron) served the role of collecting the mapping between marker positions and joint positions/angles. It consists of a flexible glove with several 9-axis IMUs placed on top of the fingers. Due to its nature, this system is immune to occlusions but presents the issue of drifting over time. This is an inherent problem of inertial tracking; it cannot be corrected without an absolute reference. Additionally, although the IMUs can, in theory, detect all degrees of freedom, their particular finger reconstruction algorithm can sense only one axis: the finger flexion-extension. As a consequence, finger spread/crossing cannot be detected.

We solved these issues by combining the readings from the Perception Neuron with the ones from the PhaseSpace, in a process known as *sensor fusion*. The details are explained in Figure 2.3.1. Accordingly, we also moved the Perception Neuron's sensors to our custom PhaseSpace glove (Figure 2.3). The IMUs were used only during the dataset recording phase, and they were removed afterward.



(a) *Hand template in a neutral pose, with marker IDs*

(b) *The glove with all the trackers (IMUs and LEDs).*

Figure 2.3 – *Illustration of the ground truth system (right), with its numerical model (left), used to create the training dataset.*

Sensor Fusion

The PhaseSpace motion capture system provides absolute tracking, whereas the Perception Neuron offers relative tracking. As the data streams between these sources differ, it is crucial to devise a sensor fusion algorithm that yields plausible results. From a high-level perspective, the algorithm is divided into a series of steps.

Setup Each marker is assigned to one of the joints of a hand template (the rigged 3D model in Figure 2.3) with the possibility of specifying an offset relative to the joint (i.e., in object space). The offsets are static and must be known in advance because they depend on where the markers have been physically placed on the glove. Three extra markers, visible on Figure 2.3, are tagged as *alignment markers* as they are used for estimating the location of the hand in space. Our choice was to form a triangle on the back of the hand, namely the markers corresponding to the wrist, the base of the index, and the base of the pinky.

Estimation of hand position/orientation We compute an optimal *rigid motion* transformation (which comprises only a rotation and a translation) from the hand template in local space to the hand in world space. More formally, we denote the positions of the joints in the hand template (Figure 2.3) as $\mathbf{U}$ and the positions of the markers as $\mathbf{V}$. The algorithm takes the two lists of points $\mathbf{U}$ and $\mathbf{V}$ (which have the same number of points and the same dimension) as input and returns a transformation $\mathscr{T}$ such that:

$$\mathscr{T}(\mathbf{u}) = \mathbf{u}\mathbf{R} + \mathbf{t} \tag{2.1}$$

where $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ is an offset. We assume that all vectors are in row-major order. This transformation minimizes the *mean squared error* (MSE) between the source positions and the target positions, defined as:

$$\mathrm{MSE}(\mathbf{U},\mathbf{V}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{V}_i - \mathscr{T}(\mathbf{U}_i)\right\|^2 \tag{2.2}$$

Fortunately, there exists a closed-form solution for this problem, which is also very efficient. It is based on the singular value decomposition (SVD) and can be computed using Kabsch's algorithm Kabsch (1976). In our case, the transformation is calculated using only the alignment markers (i.e., the three markers on the back of the hand).

Joints calibration An additional calibration stage is necessary to make the proposed approach more robust to hand variety (Figure 2.5). The user has to adopt an occlusion-free flat pose corresponding to the default pose of the animated avatar with the identity transformation for all the joints (Figure 2.4). While performing this pose, we record all the predicted joint transformations and store their inverse as the (constant) calibration offset transformation. By construction, combining each joint prediction with its calibration offset produces the desired identity transformation for that pose (Figure 2.5). This calibration offset is then applied systematically to the prediction at run-time to partially handle the variety of users' hands. It is completed with the post-processing stage described below.

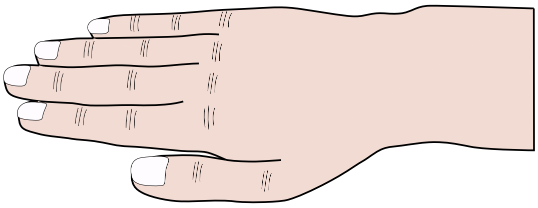
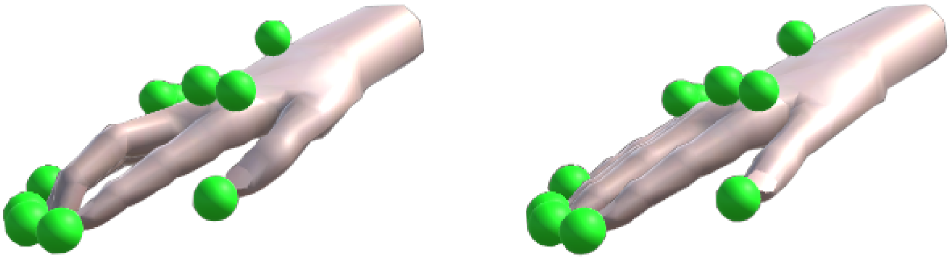


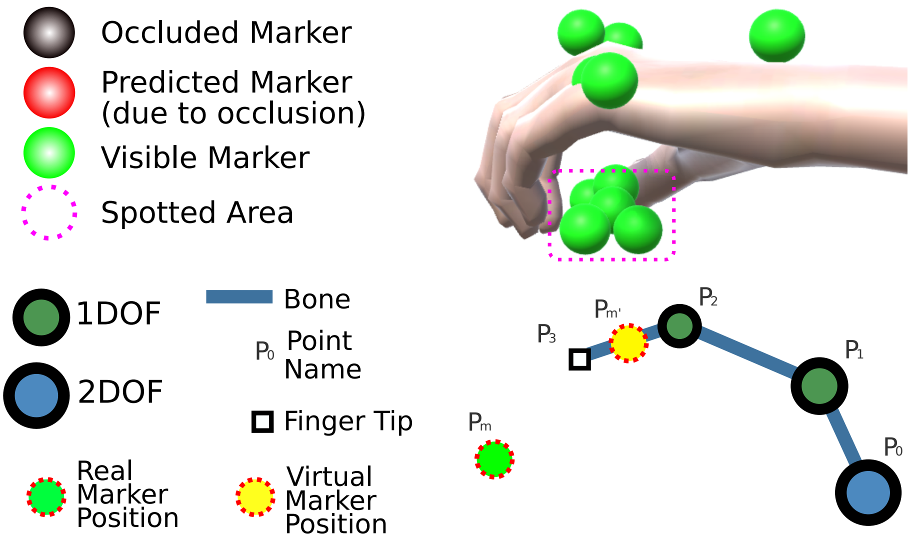
Figure 2.4 – *Illustration of the hand calibration pose*



(a) *Before* **(b)** *After*

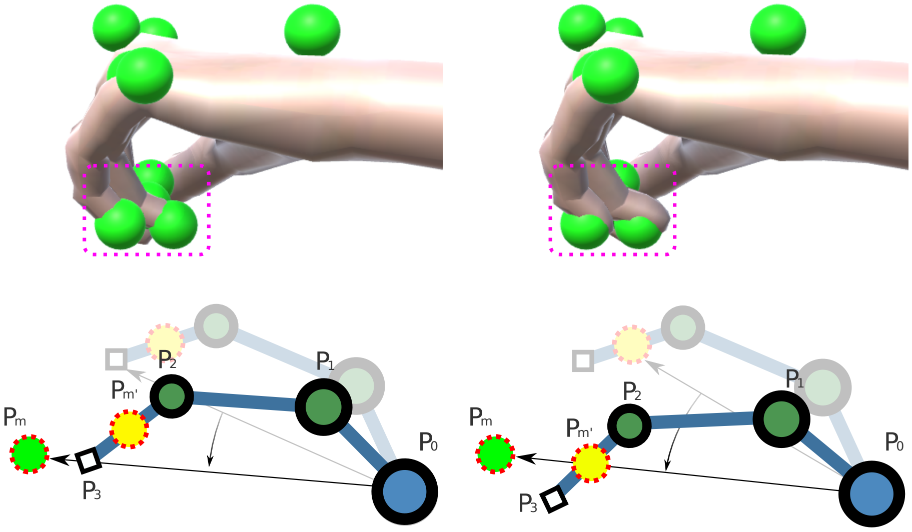
Figure 2.5 – *Illustration of the simple calibration*

Post-processing In Figure 2.6b, one can notice a gap between the fingertips and their associated markers. This is caused by inaccuracies in the Perception Neuron. We mitigate this issue by applying an artificial rotation to every finger so that, after the transformation, every finger points in the direction of the corresponding marker. Specifically, we denote with $\mathbf{p}_0$ the position of the base of the finger (metacarpophalangeal joint), with $\mathbf{p}_3$ the position of the fingertip (returned by the Perception Neuron), and with $\mathbf{p}_m$ the position of the marker (returned by the PhaseSpace). We are not interested in modifying intermediate joints as they have no associated marker. However, of course, the position $\mathbf{p}_3$ depends on the orientation of $\mathbf{p}_2$ and $\mathbf{p}_1$ (the intermediate joints, see Figure 2.6). Ideally, we would want $\mathbf{p}_3 = \mathbf{p}_m$, and this is what a traditional IK (Inverse Kinematic) solver achieves. However, this constraint is too strong since it would force unnatural postures in certain cases. On the other hand, our aim is just to apply a small correction to the data already obtained from the Perception Neuron, and therefore a simple rotation is sufficient. The approach adopted in Pavllo et al. (2018) limited our ability to reproduce contact between the thumb and other fingertips. Figure Figure 2.6d illustrates the principle of the proposed approach to reduce such a gap. We first compute the position of the last finger mid-segment $\mathbf{p}_{m'} = (\mathbf{p}_3 + \mathbf{p}_2)/2$ to better reflect the marker location. Then we rotate the finger by the shortest-arc quaternion rotation from vector $\mathbf{p}_{m'} - \mathbf{p}_0$ to vector $\mathbf{p}_m - \mathbf{p}_0$. At this point, all the limitations of the Perception Neuron have been overcome: all possible gestures/postures can be detected, including the most problematic ones (e.g., finger spread and finger crossing).



(a) *Legend*

(b) *No finger alignment; a gap may exist between the marker position and its virtual position on the virtual finger*



(c) *Alignment from Pavllo et al. (2018)*

(d) *Proposed alignment*

Figure 2.6 – *Finger alignment post-processing.*

2.3.2 Occlusion recovery model

Pipeline

We adopt a two-stage model: the first step (marker predictor) predicts the positions of the occluded markers, and the second step (subsection 2.3.3) infers the angles of all joints from the output of the first step, assuming that there are no occlusions. We train the two models separately and not in an end-to-end fashion, as our approach for enforcing temporal consistency (described in Figure 2.3.2) is not differentiable. Having two stages presents some advantages from a flexibility standpoint. If occlusion correction is not required by a particular task, the joint predictor could be used out of the box as if it were an IK solver. Moreover, a potential developer could use different algorithms for each system: the occlusion manager could be based on neural networks, linear models, or anything else, and it would not affect the behavior of the second model. Similarly, the marker prediction model could be used solely for the purpose of handling occlusions, and a traditional IK solver could be added on top of it. Note, however, that this requires a one to one correspondence between markers and joints, which is not a requirement of our system. We show a block diagram of our full pipeline in Figure 2.7.

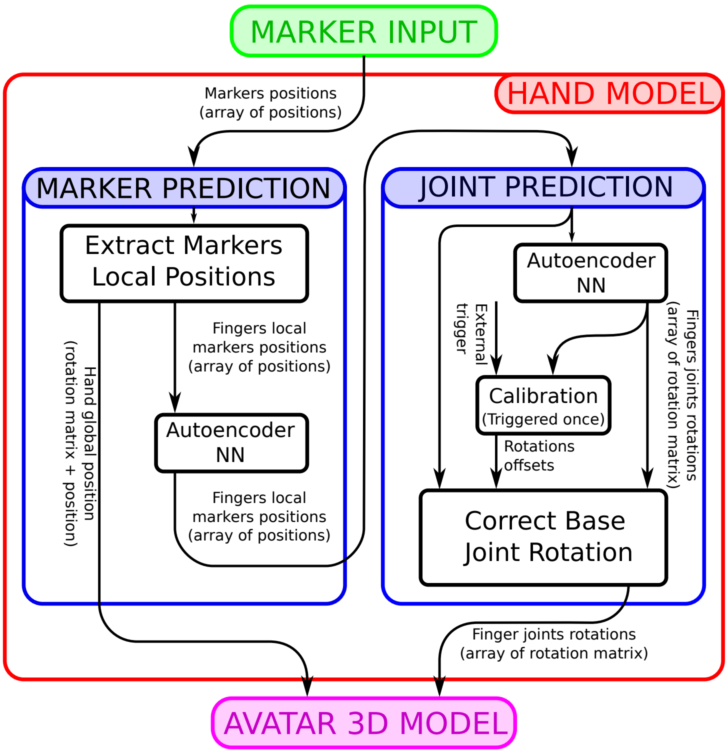

MARKER INPUT
Markers positions (array of positions)
HAND MODEL
MARKER PREDICTION
Extract Markers Local Positions
Hand global position (rotation matrix + position)
Fingers local markers positions (array of positions)
Autoencoder NN
Fingers local markers positions (array of positions)
JOINT PREDICTION
Autoencoder NN
External trigger
Fingers joints rotations (array of rotation matrix)
Calibration (Triggered once)
Rotations offsets
Correct Base Joint Rotation
Finger joints rotations (array of rotation matrix)
AVATAR 3D MODEL

Figure 2.7 – *Conceptual full prediction pipeline*

Marker Predictor

Before presenting our model based on neural networks, we recall the three simple baselines that are evaluated against our method.

It is worth mentioning an important property that this step must implement: *temporal consistency*. The model should enforce a "smoothness" condition between subsequent frames so as to avoid *discontinuities* (sudden jumps in the joint transformations). We can identify two types of discontinuities:

Discontinuity on occlusion: when a marker is visible at time t and becomes occluded at time $t+1$.

Re-entry discontinuity: when a marker that was previously occluded at time t becomes available again at time $t+1$.

While discontinuities on occlusions can be corrected explicitly by enforcing temporal consistency in the model, re-entry discontinuities cannot be solved without having future knowledge of the data. In a real-time system like ours, this means that they must be smoothed manually as a post-processing step.

Baselines

We now introduce the three aforementioned baselines in order of increasing complexity:

Last known position The simplest baseline consists in keeping the last known position of an occluded marker. With regard to discontinuities on occlusions, this method is temporal consistent.

Moving average Inspired by Piazza et al. (2009), we take velocity into account. We keep a moving average of the velocities of each marker over the last k frames (we use $k = 20$, i.e., a third of a second) to minimize the effect of noise. When a marker is occluded, this baseline simply moves the marker along the trajectory defined by the average velocity.

Affine combination model Finally, we propose an improvement over the previous baselines. Another simple (yet effective) method consists in expressing an occluded marker as an affine combination of the other available markers, i.e., a linear combination with weights that sum up to 1. The computation is performed using the data from the previous frame, where the occlusion was not present. In order to enforce the affine property, it is sufficient to add a homogeneous coordinate to each point and fix it to 1. More formally, we denote with $\mathbf{X_i}$ ($i = 1..N$) the set of all known positions ($\mathbf{X}$ is a $N \times 4$ matrix), $\mathbf{Y_j}$ ($j = 1..M$) the set of occluded points that must be predicted (the result $\mathbf{Y}$ would be a $M \times 4$ matrix), and $\mathbf{W}$ the weight matrix of size $M \times N$. It must follow that:

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \tag{2.3}$$

$$\sum_{i=1}^{N} \mathbf{W}_{j,i} = 1 \quad \forall \mathbf{j} \tag{2.4}$$

where $\mathbf{X_i} = [X_{ix}, X_{iy}, X_{iz}, 1]$ and $\mathbf{Y_j} = [Y_{jx}, Y_{jy}, Y_{jz}, 1]$. We are again assuming a row-major vector notation. This problem can be solved using exactly four non-coplanar markers. If more markers are available, the problem is underdetermined, as there are infinite solutions to the linear system. Therefore, we apply L2 regularization, which means that among all possible solutions, we choose the one that minimizes the squared norm of the weight vector. In other words, the predicted position should depend on all the other markers, each of which has a small weight; this leads to better robustness to noise. We minimize the loss function:

$$\mathscr{L}(\mathbf{W}) = \sum_{j=1}^{M} \left(\left\| \mathbf{Y}_j - \mathbf{W}_j \mathbf{X} \right\|^2 + \lambda \left\| \mathbf{W}_j \right\|^2 \right) \tag{2.5}$$

$$= \left\| \mathbf{Y} - \mathbf{W}\mathbf{X} \right\|_F^2 + \lambda \left\| \mathbf{W} \right\|_F^2 \tag{2.6}$$

where $\|\mathbf{M}\|_F$ denotes the Frobenius norm of $\mathbf{M}$, and λ is a small positive regularization constant ($\lambda = 10^{-8}$ is suitable in our case; in general, one should choose the smallest value that does not cause numerical precision issues). Fortunately, the function is convex, and there exists a closed-form solution for its minimum. We derive the gradient with respect to $\mathbf{W}$ and equal it to zero:

$$\nabla \mathscr{L}(\mathbf{W}) = -2\left(\mathbf{Y} - \mathbf{W}\mathbf{X}\right)\mathbf{X}^T + 2\lambda\mathbf{W} = \mathbf{0} \tag{2.7}$$

Solving for $\mathbf{W}$ we obtain:

$$\mathbf{W} = \mathbf{Y}\mathbf{X}^T \left(\mathbf{X}\mathbf{X}^T + \lambda\, \mathbf{I_N}\right)^{-1} \tag{2.8}$$

where $\mathbf{I_N}$ is the $N \times N$ identity matrix. This approach is closely related to *ridge regression* Hoerl and Kennard (1970).

As before, discontinuities on occlusions are avoided by design. Furthermore, this baseline is intrinsically invariant to translations and rotations. Since $\mathbf{Y_j}$ is expressed as an affine combination of all $\mathbf{X_i}$, any rigid transformation applied to $\mathbf{X}$ would be applied to $\mathbf{Y}$ as well, i.e., $f(\mathscr{T}(\mathbf{X})) = \mathscr{T}(f(\mathbf{X}))$ (where $\mathscr{T}$ is a rigid transformation). From a practical standpoint, if the hand is kept in a static posture and moved around the capture space, the occluded markers are reconstructed perfectly. We also found this baseline to perform relatively well on gestures that do not involve complex movements.

Marker Regressor

In theory, neural networks (NNs) can approximate any function (provided that a sufficient number of neurons is available) Hornik et al. (1989), but, in practice, the result is strongly dependent on how the data is pre-processed.

Similarly to the affine combination model, we want our prediction to be spatially invariant in the sense that any translation/rotation transformation applied on the input points should not affect the output of the neural network. Therefore, we enforce, for this step, a pre/post-processing scheme that allows the network to learn proper mapping thanks to the reduced search space. These are named:

Marker position extraction in hand referential (registration): The rotation and translation of the hand in space are removed. This can be achieved by aligning the hand to the standard template, which is centered on the world's origin and is oriented toward a predefined axis. The alignment can again be performed by finding the lowest-error rigid motion transformation. This process can be regarded as the inverse operation of the hand position estimation presented earlier: instead of moving the hand template towards the markers, here, the markers are moved towards the hand template. The only difference is that here we use all available markers, and not only the three alignment markers (since they may be occluded). The hand template is kept in a neutral pose (see Figure 2.3), and therefore this step is dependent on the hand posture, but this does not represent a problem as the goal of this step is to perform spatial normalization.

Reconstruction of the positions of the markers (de-registration): The inverse transformation is applied to the predicted points; that is, the markers are put back to their original positions in world space.

With regard to how occlusions are handled, it is important to note that neural networks cannot operate on missing data. Hence, a special architecture and/or training procedure is required. A thorough approach consists in building an ensemble of different models Jiang et al. (2005), one for each possible set of available markers, and training them independently from each other. It is clear that this method presents a severe limitation: the number of models to train increases exponentially as more markers are added, not to mention the tremendous computational (and memory) cost both at training and inference.

Instead, we employ a single feed-forward neural network configured as an *autoencoder*, i.e., a topology that maps the identity function $\mathbf{x} \longrightarrow \mathbf{x}$, as depicted in Figure 2.8. The network comprises $3N$ input neurons and $3N$ output neurons, where N is the total number of markers (9 in our case). Each group of 3 neurons encodes the XYZ positions of a particular marker after the pre-processing step described above.

The structure of the neural network is shown in Table 2.1. All layers except the last one use ReLU (Rectified Linear Unit) activation functions Nair and Hinton (2010), defined as $y = \max(0, x)$, as they have been shown to yield the best results in a wide range of tasks Glorot et al. (2011); Krizhevsky et al. (2012). The output layer uses a linear activation function, thereby allowing an unbounded output range. All hyperparameters were chosen to minimize the reconstruction error on the validation set, also taking into account performance and latency constraints. We also experimented with varying numbers of layers and discovered that more layers lead to overfitting on this specific task (regardless of regularization).

Our mechanism for handling occlusions is closely related to *Dropout* Srivastava et al. (2014), a training technique traditionally used to avoid overfitting the training set. Dropout works as follows: during training, at each iteration, a random fraction of neurons are disconnected (which is equivalent to setting their output values to 0). At inference, all neurons are used. We apply a procedure similar to Dropout on the input layer. The model is trained using a data augmentation procedure: the dataset is generated in real-time by setting a random number of points (groups of 3 neurons) to 0 from frames containing exclusively all visible markers, according to the distribution observed in Figure 2.14 (with a number of occlusions between 1 and 4). The exact distribution is not crucial, but it helps with improving the error in realistic cases. It is worth noting that the pre/post-processing scheme still applies to this approach. The inputs must be disconnected *after* the positions are registered (i.e., are transformed into object space). The prediction algorithm is trivial: all available (non-occluded) points are registered and passed as inputs to the neural network, whereas the inputs corresponding to missing values are set to 0; the relevant outputs (i.e., the ones corresponding to the occluded markers) are extracted and de-registered.

Autoencoders learn a compressed representation of the data Bourlard and Kamp (1988), instead of just copying the input to the output. In our particular case, the bottleneck layer learns a *positional embedding*, i.e., a vector that encodes a particular posture. Our representation is overcomplete, meaning that the number of neurons in the bottleneck is greater than the number of input neurons. However, our training procedure acts as a regularizer, effectively forcing the model to learn a sparse representation that is suitable for reconstructing missing values. ReLU activations also contribute to sparsity Glorot et al. (2011).

Discontinuities Unlike the affine model discussed earlier, the feed-forward neural network approach tends to suffer from discontinuities because it does not enforce temporal consistency

Table 2.1 – *Full list of layers in the marker regressor.*

Type	Shape
Input	9×3
Flatten	27 neurons
Fully connected + ReLU	200 neurons
Fully connected + ReLU	150 neurons
Fully connected + ReLU	200 neurons
Fully connected + Linear	27 neurons
Reshape (output)	9×3

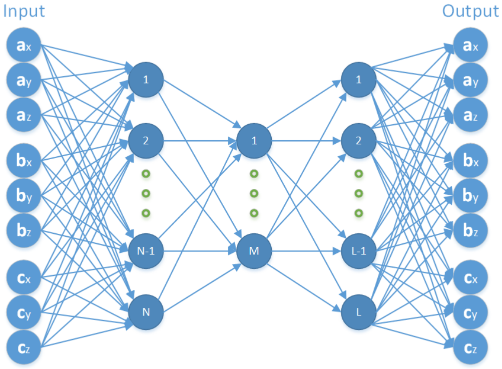
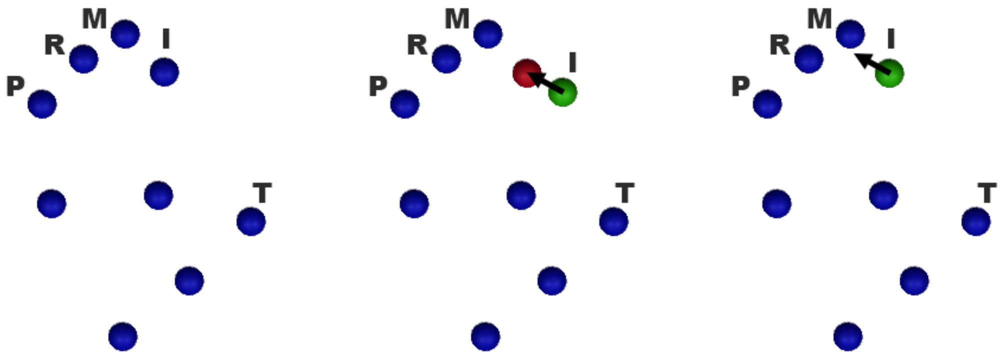


Figure 2.8 – *An autoencoder with three hidden layers. In a scenario where* **a** *and* **c** *are available, and* **b** *is occluded, we disconnect the inputs corresponding to* **b***, and we get the prediction of* **b** *in the output. Here, only three markers are depicted; in practice, we would have nine markers.*

explicitly. Since a feed-forward model does not contain any state information, it simply finds a solution that minimizes the error in the average case without being able to take into account any previous context. From the user's point of view, this results in a bad experience. Other neural network architectures, such as recurrent neural networks (RNNs), can exploit past information. However, even with them, handling missing values is a non-trivial task that could still result in discontinuities. Our preliminary experiments showed that this is indeed the case. Hence, we stick with feed-forward networks due to their lower computational cost and ease of training, and we adopt special measures to correct discontinuities. When a marker becomes occluded, we compute an offset term, and we apply it to all subsequent outputs until the occlusion is resolved. More specifically, given an occlusion at time t, we perform a prediction with the data from the previous frame $t-1$ (where the real position was known). Afterward, we calculate an offset that cancels out the discontinuity; this offset is retained as state information and is modified only if another occlusion happens or if the occlusion is resolved. The offset is applied to the output of the marker predictor network before the points are de-registered. To calculate the offset, we simply compute the difference between the predicted position and the actual position in local space. We also explicitly correct re-entry discontinuities using the same technique; the only difference is that the offset is decayed to zero over time (using a linear decay function) in order to remove the bias. We observed that a decaying speed of 25 cm/s offers a good compromise between reactivity and smoothness. Figure 2.9 depicts this process.



(a) *Frame* $t-1$ (b) *Frame* t (c) *Frame* $\geq t$

Figure 2.9 – *Handling of discontinuities.* ***(a)*** *At* $t-1$ *no marker is occluded.* ***(b)*** *At* t *the index marker is occluded. The NN predicts its hypothetical position at* $t-1$ *(green), which results in a small discontinuity from the true position (red).* ***(c)*** *From* t *onwards, the discontinuity is explicitly canceled by moving the marker by the offset vector (black arrow). For re-entry discontinuities, the process is reapplied in the opposite direction, but the offset vector is progressively shrunk to remove the bias.*

2.3.3 Finger animation model

The joint regressor predicts the angles of the fingers, given the marker positions as input. It solves a task similar to that of an IK solver, but instead of using a calibrated skeleton, it adapts to the user's hand according to a dataset of realistic motions. Furthermore, it does not need to handle missing values, as they are assumed to be predicted by the previous stage of the pipeline. We adopt a dense neural network for this task, which takes the marker positions as inputs ($9 \times 3 = 27$ neurons), and predicts the Euler angles of all relevant joints for a total of 26 Euler angles. The use of Euler angles instead of other representations, such as exponential maps Grassia (1998) or quaternions, is motivated by the observation that our fingers have limited degrees of freedom. We need to predict only certain angles, thereby obtaining a smaller neural network. Figure 2.10 shows the degrees of freedom that are modeled, while Table 2.2 shows the structure of the neural network. As before, all layers except the last one use ReLU activation functions Nair and Hinton (2010). Moreover, we used Dropout Srivastava et al. (2014) in the intermediate layers (with a probability of 0.1, meaning that 10% of neurons are randomly dropped at each training iteration) to avoid overfitting. This proved effective in improving the validation error.

During both training and inference, the inputs are registered (all rotations/translations are removed). Additionally, we found the same post-processing technique employed in the sensor fusion (i.e., artificial joint rotation, Figure 2.3.1) to be effective.

2.3.4 Training

For both models, we optimize the mean squared error (MSE) loss using the Adam optimizer Kingma and Ba (2014) with an initial learning rate $\eta = 0.001$. The learning rate is automatically

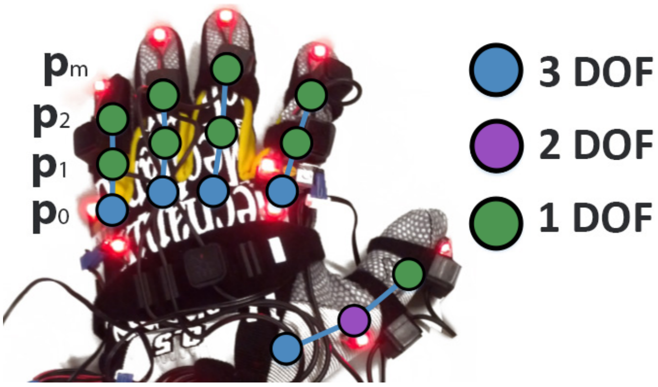


Figure 2.10 – *Degrees of freedom of the hand joints (26 in total).*

Table 2.2 – *Full list of layers in the joint regressor.*

Type	Shape/Notes
Input	9×3
Flatten	27 neurons
Fully connected + ReLU	200 neurons
Dropout	$p = 0.1$
Fully connected + ReLU	200 neurons
Dropout	$p = 0.1$
Fully connected + ReLU	200 neurons
Fully connected + Linear (output)	26 neurons

adjusted once the error reaches a plateau; more specifically, it is halved if the error has not improved over the last five epochs. The model is trained only on simulated occlusions, as they are the only ones for which a reliable ground truth can be obtained, and with a batch size of 32 samples.

2.3.5 Second hand pipeline mirroring

The reconstruction of the right-hand pose exploits the model trained for the left hand. For this, we simply transpose the behavior of the left-hand pipeline to the right hand, using the natural plan of symmetry of our skeleton to flip the coordinates of the marker according to this plan. The new pipeline handling both hands is a composition based on the pipeline described in subsection 2.3.2.

Then the set of mirrored coordinates is given as an input to the pipeline with the neural network for the prediction of the positions of the markers. The neural network trained on the left-hand dataset now sees an input matching a left hand and predicts the occlusions for this virtual left hand.

These data are stored in the hand model object in order to be accessed for the hand pose estimation and for the next step. The next step consists in filling these predicted markers to the neural network trained on the left hand to predict the joint rotations of each finger, and as above, it sees a left hand.

As for the marker position prediction, the output is also stored in the hand model object and is forwarded to the 3D model for its animation.

This design allows us to train only once the neural network with the dataset of one hand and use it as many times as required for the number of hands required in the simulation. Also, we were able to use four hands in our simulation environment.

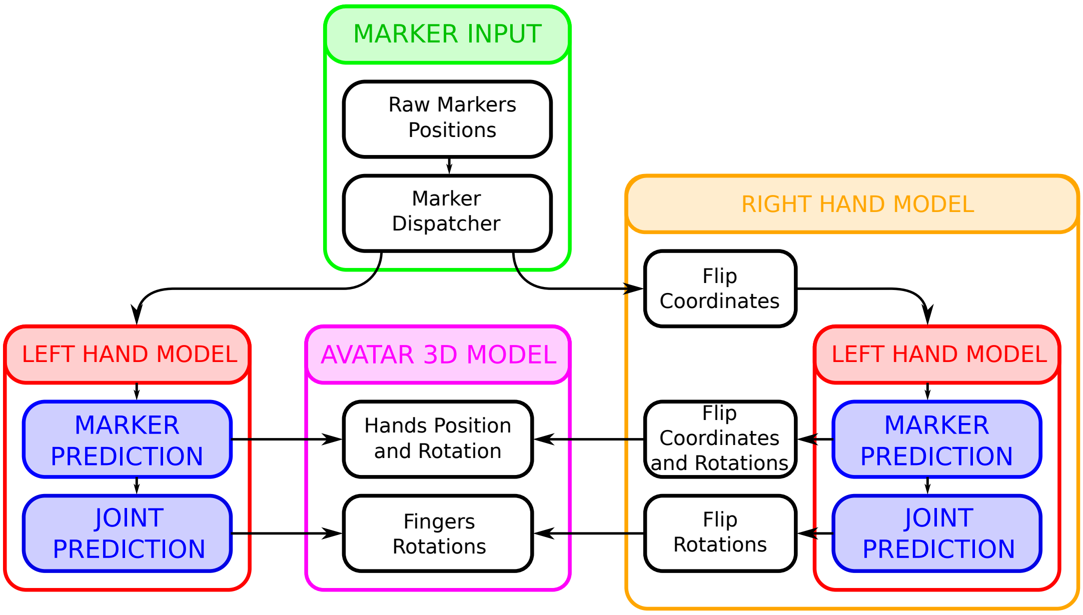


Figure 2.11 – *Main steps of the pipeline used to transform raw markers position from VRPN to the position of the avatar's body. Refer to Figure 2.7 for details of the left-hand model*

2.4 Approach validation

2.4.1 Evaluation dataset

First evaluation: Single Hands Dataset The dataset was recorded from four subjects (three males and one female, age range between 22 and 30), all right-handed and with different hand sizes. Every subject underwent eight recording sessions of approximately 60/80 seconds each, and the Perception Neuron was calibrated before each session (with a quick follow-up check). This approach ensures that IMU drifts do not degrade the dataset's accuracy. As for the movements, the subjects were left free to execute any movement but were also instructed to perform at least some key gestures. In order to evaluate the model, the dataset was partitioned into a *training set* (2 subjects), a *validation set* (1 subject), and a *test set* (1 subject). The validation set was used for tuning the hyperparameters and testing different architectures, whereas the test set was used only for the final evaluation.

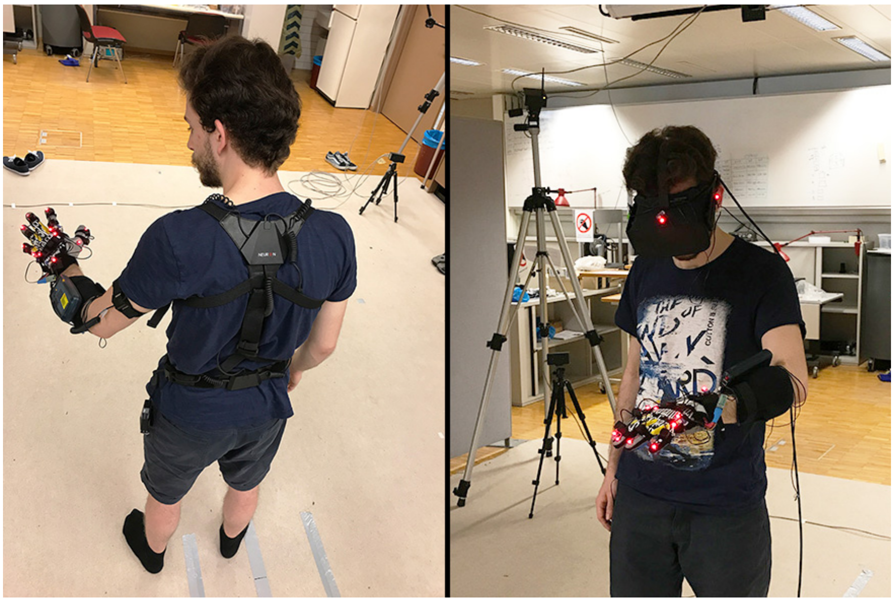


Figure 2.12 – ***Left:*** *a person wearing the recording equipment (PhaseSpace glove and Perception Neuron).* ***Right:*** *a person testing the application in a VR environment with an Oculus HMD.*

Second evaluation: Two Hands Dataset A second dataset was recorded from five subjects (four men and one female, age range 24-42), all right-handed with different hand sizes. Each subject spent 100 seconds wearing the two gloves but this time without the perception neuron system as it was no longer required. The subjects had to achieve three tasks (both hands finger crossing, palms in contact, fingers in contact) and were free to move the rest of the time. This dataset was used in order to compute the rate of occlusions per hand in comparison with a system with only one hand.

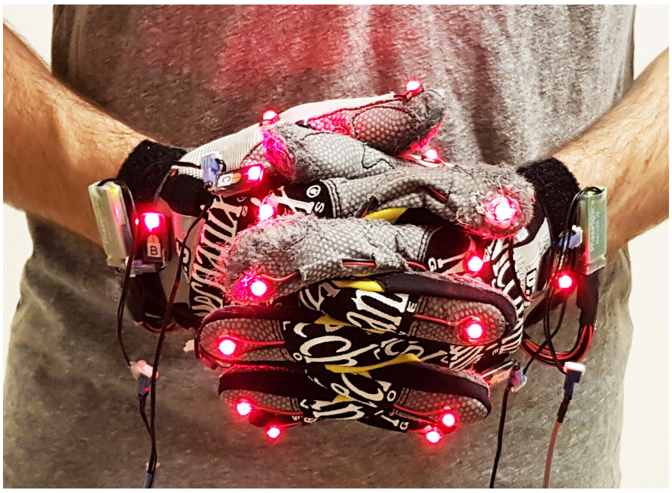


Figure 2.13 – *A subject wearing the two gloves during the recording phase of the dataset*

2.4.2 Results

Left hand dataset

The training set consists of ≈30 minutes of data recorded at 60 FPS. In theory, the PhaseSpace system can record at up to 480 FPS, but we limited the sample rate to 60 FPS to avoid collecting too many redundant samples. Figure 2.14 reveals some insights: most occlusions involve a small number of markers, that is, the probability that multiple markers are occluded at once is low. Moreover, the duration of an occlusion follows a heavy-tailed distribution (90% of occlusions last less than 0.36s).

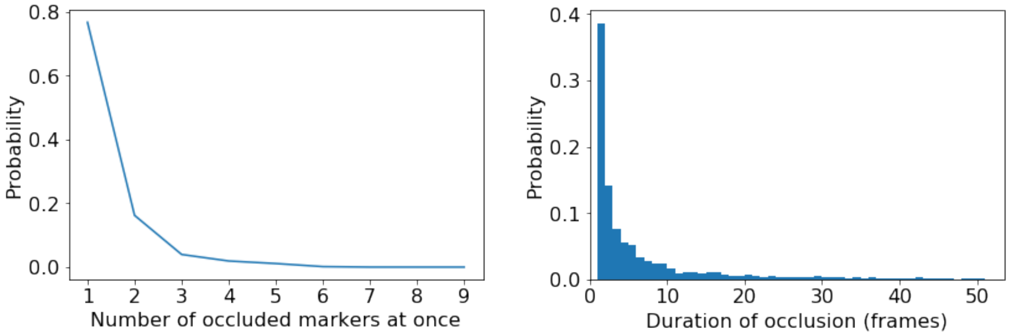


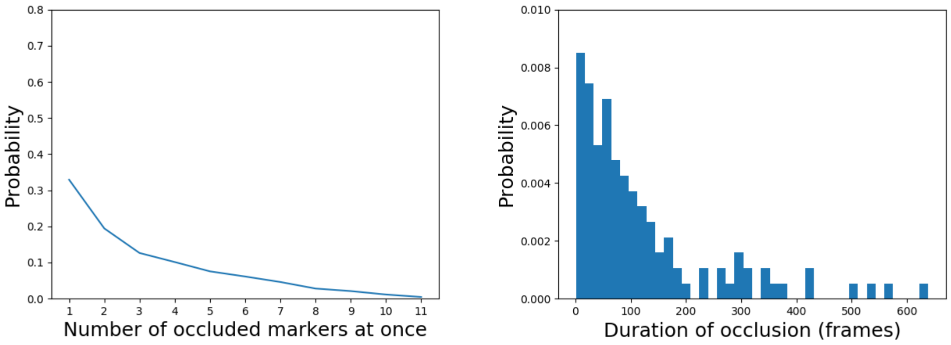
Figure 2.14 – *Probability of N occlusions at once (Left). / Occlusion duration histogram (Right).*

Two hands dataset

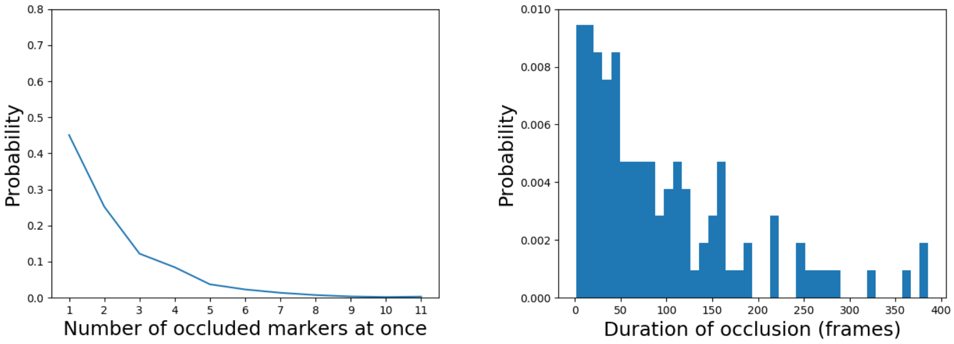
This dataset is used to compute the number of occlusions occurring with two hands instead of one hand. We used the same frame rate for this comparison. As we can see in Figure 2.15, the probability to have more than one or two occluded markers is higher than in the single-hand case. The occlusion duration is likely to be longer too. This can be explained by the fact that interacting hands may temporarily hide each other.

Reconstruction error

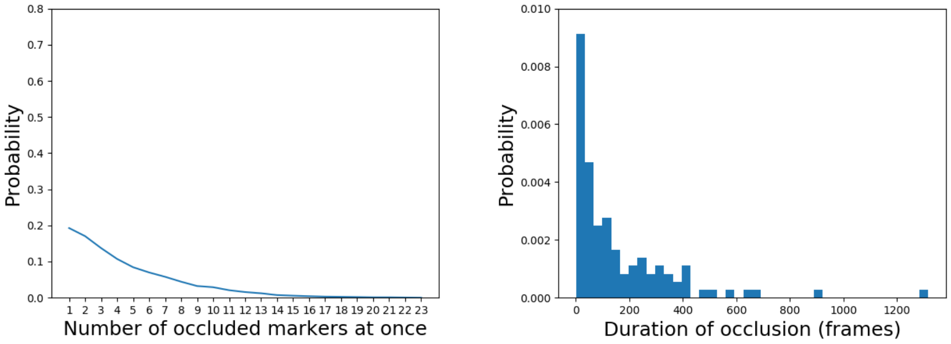
Marker predictor error Table 2.3 and Figure 2.16a reveal the error of the first stage of the pipeline (the marker prediction model). We compare the three baselines with our neural network approach, and we report statistics over varying occlusion durations and the number of markers occluded simultaneously. For each trial, we report the average Euclidean distance between the predicted position and the ground truth *on the last frame* before the occlusion is resolved, and only for the occluded markers. For instance, in the scenario “2 markers after 100 ms”, we occlude two random markers at once and measure their error after 100 ms. The errors are evaluated across the entire test set and repeated five times with different random seeds to smooth out their variance.



(a) *Left Hand*



(b) *Right Hand*



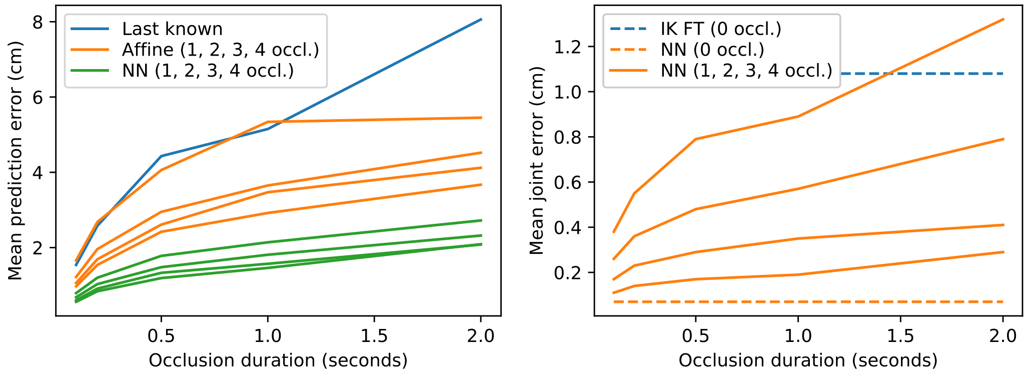
(c) *Both Hands*

Figure 2.15 – *Probability of N occlusions at once (Left). / Occlusion duration histogram (Right). Plots represent the same as in Figure 2.14*

Our evaluation methodology addresses both short-term occlusions (100 ms, 200 ms, 500 ms) and long-term occlusions (1 s and 2 s). Each method is tested on a number of occlusions between 1 and 4, except for the first two baselines (*last known position* and *moving average*), which are independent of this parameter. We observe that the moving average baseline exhibits the worst performance, which is caused by the markers drifting away on long-term occlusions. The affine combination model is better than the simplest baseline (last known position) except when many markers are occluded at once. Finally, our neural network approach consistently outperforms all the other methods.

Table 2.3 – *Evaluation of the error on the marker neural network (error units=centimeters, lower=better). Legend:* ***LK*** *last known position,* ***MA*** *moving average,* ***AC*** *affine combinations,* ***NN*** *neural network.*

	#	Occlusion duration (seconds)				
Method	Occlusions	0.1	0.2	0.5	1.0	2.0
LK	Any	1.54	2.58	4.43	5.15	8.06
MA	Any	2.23	4.28	9.99	20.19	44.28
AC	1	0.97	1.54	2.42	2.92	3.67
	2	1.06	1.69	2.61	3.47	4.12
	3	1.22	1.96	2.95	3.65	4.52
	4	1.66	2.68	4.06	5.34	5.45
NN	1	0.56	0.84	1.19	1.46	2.09
	2	0.60	0.91	1.33	1.57	2.08
	3	0.68	1.03	1.48	1.81	2.32
	4	0.79	1.20	1.78	2.14	2.72



(a) **(b)**

Figure 2.16 – *(a) Comparison between the baselines and our method for the marker predictor error. The moving average baseline is not included because of its excessive error.* ***(b)*** *End-to-end error: our approach at varying conditions versus an IK baseline.*

End-to-end error In Table 2.4 and Figure 2.16b, we report the error relative to the joint positions by running the entire pipeline. As in the previous section, we measure the average Euclidean distance between the predicted joint positions and the ground truth. The averages are computed only over the finger joints, i.e., $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ (as described in Figure 2.3.1). All errors are relative to the test set. We do not report angle errors because they would not be easily interpretable; errors in the first joints would accumulate along the kinematic chain.

We compare our work to an IK library, "Final IK" by RootMotion (RootMotion, 2020). We fine-tuned the IK configuration to the best of our ability: we use a Cyclic Coordinate Descent

(CCD) solver, with an angle constraint (3 degrees of freedom, max. 45° for flexion-extension / abduction-adduction, and 20° for the twist) on the root finger joints, and a hinge constraint (1 degree of freedom (flexion-extension) from -90° to 10°) on middle joints. We show that our approach achieves a significantly lower error (0.07 cm) than inverse kinematics (1.87 cm unconstrained, 1.08 cm fine-tuned) when there are no occlusions. This suggests that a data-driven approach is better at modeling the angle distributions/constraints than a handcrafted setup, thus producing a more naturally-looking reconstruction. For the occlusion scenario, we report only the statistics associated with our method, as IK solvers cannot handle occlusions (some IK approaches such as Schröder et al. (2015) enable a reduced set of markers, but not a dynamically-changing one).

Table 2.4 – *Evaluation of the error on the final joint positions (error units=centimeters, lower=better). Legend:* ***IK*** *inverse kinematics,* ***FT*** *fine-tuned,* ***NN*** *neural networks.*

	#	Occlusion duration (seconds)				
Method	Occlusions	0.1	0.2	0.5	1.0	2.0
IK	0	1.87 (no occlusions)				
IK FT	0	1.08 (no occlusions)				
NN	0	0.07 (no occlusions)				
	1	0.11	0.14	0.17	0.19	0.29
	2	0.17	0.23	0.29	0.35	0.41
	3	0.26	0.36	0.48	0.57	0.79
	4	0.38	0.55	0.79	0.89	1.32

2.4.3 Performance

Our reference implementation is written in C# and runs on Unity Engine. Running the entire pipeline for a single hand on an Intel Core i5-4460 CPU requires less than 1.2 ms (≈ 833 frames per second). Additionally, the two neural networks (for the occlusion recovery and the finger animation) have a minimal memory footprint (300 kB each). With an Intel Core i9-9900K, the full pipeline takes 2.2 ms on average for both hands integration, a result based on a simulation of 5 minutes (≈ 455 frames per second). In the video, hosted at https://www.youtube.com/watch?v=S8c-F2kvqZ8, our approach is used, without physical simulation or collision detection nor enforced kinematic constraints when objects are touched, to illustrates the fitness of the results from our approach.

2.5 Discussion and future work

Having both hands in VR gives the user a more natural way to interact with elements of the virtual world. It allows us to perform simultaneous actions, like changing gears while driving, but also to achieve more complex tasks, like handling large objects. The provided video illustrates the actions of opening a drawer to grab an object, crossing fingers, and shaking hands.

Concerning related work on this subject, previous methods have mainly addressed passive motion capture and limb reconstruction. Out of the few occlusion-handling solutions targeted at active-marker technologies, we investigated Aristidou et al. (2008), which has already been employed in some studies Molla et al. (2017). However, this method requires at least three markers for each segment, which follows from the assumption that the distance between neighboring markers is approximately constant. Given that our hand model comprises at most two markers per finger (tip and, optionally, base), we suggest that a data-driven approach is more suited to this task because it adapts better to the specific domain that should be addressed (hand and fingers reconstruction with only 1 or 2 markers per finger, in our case). This might also explain why, in our setting, analytic inverse kinematics perform significantly worse: in the absence of intermediate markers, the algorithm does not know the priors that constitute a good-looking posture.

2.5.1 Limitations

Limited training dataset During the training phase of the autoencoders, some complex hand postures might not have been trained due to the likelihood of occlusions, thus excluding these frames from the training dataset. Consequently, the integration of both hands might increase the number of mutual occlusions for which the system was not trained. For instance, in the edge case of too many simultaneous occlusions, the pipeline might give unplausible poses such as illustrated in Figure 2.17.

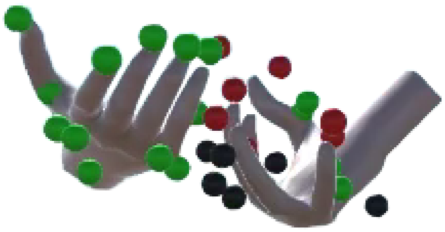


Figure 2.17 – *Too many occlusions*

In that regard, we expect that training a new neural network handling both hands simultaneously could help to predict their correct relative position in contexts where one hand is hidden by the other one. It also has to be noted that the thumb has a more complex structure than the other fingers, and the post-processing that simply applies a rotation on the base joint could be improved for that finger.

Absence of physical hand model Another disadvantage of our approach lies in the feed-forward neural network architecture, which does not model an internal state.

On a collision side, this means that the self-contacts might not be congruent with the actual pose of the user's hands, with nothing preventing interpenetration or unexpected gaps as illustrated in Figure 2.18. Consequently, the results obtained from the animation only rely on the precision of the generated pose by the approach.

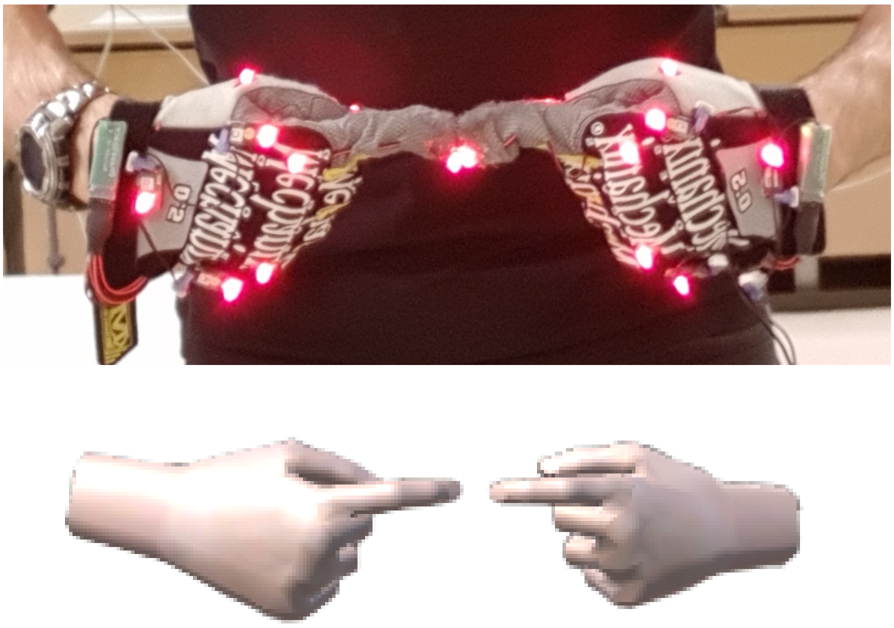


Figure 2.18 – *An illustration of the gap between the two fingers*

On the temporal side, discontinuities, when occlusions occur or vanish, are corrected in post-processing after the prediction from the model. Indeed, our neural model does not embed temporal continuity as it performs deterministic predictions in the sense that equivalent postures (same input with different outputs) are averaged to minimize the reconstruction error. However, recurrent neural networks can produce an output that is conditioned on the previous frames, therefore, potentially handling discontinuities without a post-processing stage, although it is not trivial to enforce temporal consistency on occlusions while keeping the target function differentiable.

Hardware Finally, on a hardware note, we noticed that, depending on the environment (number of cameras, reflective surfaces, etc.), the PhaseSpace tracking system might give wrong marker positions for fast movements (Figure 2.19) rather than yielding occluded ones.

However, compared to the other available tracking methods, this solution remains with greater precision than its passive counterpart, and the labeling of the markers easily allows for tracking multiple hands and, possibly, multiple people. Nevertheless, our method would seamlessly adapt to passive systems if a robust tagging layer were integrated into the pipeline, such as the one proposed by Han et al. (2018); Alexanderson et al. (2017).

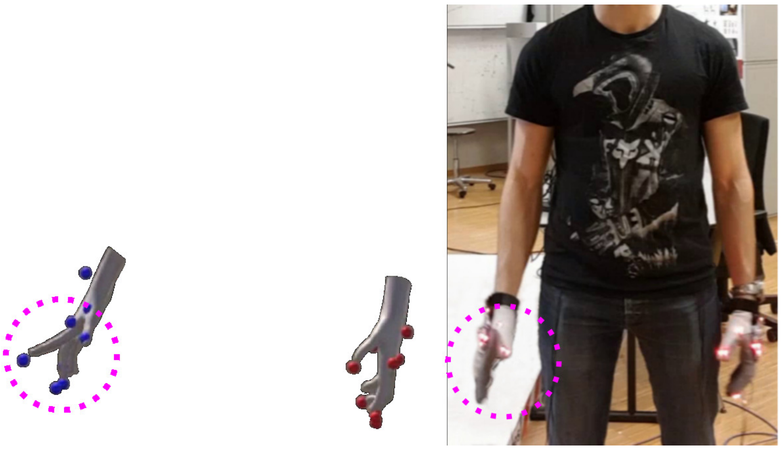


Figure 2.19 – *Fast movement inducing a bad marker position*

2.5.2 Future works

In the future, we would like to experiment with convolutional, *long short-term memory* (LSTM), and *gated recurrent unit* (GRU) architectures. The architecture could also be extended with a second neural network trained with a dataset of two interacting hands rather than having a single-hand neural network exploited independently for each hand and, therefore, might help maintain self-contact consistency known to be critical to support embodiment at the body level as this was covered in the literature Bovet et al. (2018). However, the existing literature presents a dearth regarding the characterization of the finger embodiment, and our finger animation technique can be used to broaden our understanding of the mechanism at stake in the human perception and attribution of finger movements.

2.6 Conclusion

To sum up, we present a hand-tracking pipeline addressing the issue of occlusions from active optical tracking that also provides an animation pipeline of virtual hands through the use of neural networks. This method of mapping markers to joint angles does not require the process of setting up an IK solver.

Our system provides a natural reconstruction of the hands in most real-case scenarios, which we demonstrate by comparing the reconstruction error with a traditional solver based on inverse kinematics. Occlusions are corrected with good accuracy in most cases and with minimal latency, and our data-driven approach does not require defining a set of rules or constraints, as these are learned automatically from data. From an interaction perspective, our finger animation is suitable for object grasping and manipulation, but we observe that the behavior of the thumb, which falls short on pinching gestures, could be improved by using a 3D hand model that resembles the glove more closely.

3 How are errors in finger animation perceived?

3.1 Introduction

We feel responsible for the actions we do and for the mistakes we make. In computer games, as in sports or at work, missing a button press under time pressure is self-attributed as a failure to succeed in the task. But if our avatar in VR (Virtual Reality) would automatically correct for it, pretending the mistake never occurred, would we still feel responsible for that mistake, or simply ignore it, or even not notice it at all?

The self-attribution of authorship for voluntary actions is defined as the SoA (Sense of Agency), which corresponds to the subjective feeling to be responsible for the action of our body Gallagher (2007); Haggard (2008); Salomon (2017). The representation in VR of an avatar replicating a user's movements thanks to motion capture is known to induce a strong SoA for the movement of the virtual body. This is illustrated at the body level through the adaptation from Slater et al. Slater et al. (2008) of the "Rubber Hand Illusion" from Botvinick et al. (Botvinick and Cohen (1998)) with the difference that the limb is not a physical limb, but a virtual one displayed using immersive technologies. Those immersive setups can also induce a sense of self-location in the VE (i.e., I am located where the virtual body is located), which, carefully combined with the SoA can lead to the subjective experience of body ownership SoE (i.e., this virtual body is my body). This feeling of the embodiment is beneficial for the user experience in VR, but any disruption of one of them can potentially lead to a break in embodiment Porssut et al. (2019). However, we are poor at locating our limbs using only the proprioceptive feedback Burns and Brooks (2006) and consequently, we do tolerate motion distortions at the body level, provided that those are below a certain threshold Galvan Debarba et al. (2018); Porssut et al. (2019). Exceeding this threshold would lead to a loss of agency, breaking off the SoE. Therefore, it is crucial not to disrupt the SoA to maintain the user's experience.

It is assumed that to perform this SoA judgment, an underlying neural process would compare a prediction of the sensory consequences of a movement with the actual feedback from our senses Wolpert et al. (1995); De Vignemont and Fourneret (2004); David et al. (2008); Engbert et al.

(2008). If both match, the SoA is high, but if a mismatch occurs, there is a loss of SoA. It was thus surprising to observe that humans could self-attribute actions that were distorted (i.e., When the user's movement differs from the avatar's motion) or performed by others Nielsen (1963) or when the final result was altered by the machine Logan and Crump (2010). Prior results typically showed that, beyond a few hundred milliseconds of delay, people do not self-attribute the response Farrer et al. (2008); Wen (2019); therefore, only real-time feedback is considered in this study.

In the seminal study of Nielsen et al., subjects were asked to draw a line while the experimenter secretly placed a mirror to replace the subject's real hand with someone else's hand doing the same task Nielsen (1963). When both actions were synchronous, subjects experienced the alien limb (i.e., the limb that does not belong to the user) and its movements as their own. When the alien limb drew a curve instead of a line, subjects compensated for the error by making involuntary corrections in their movement while still considering the limb to be their own. With more advanced techniques using computer graphics, Burns et al. managed to introduce motion discrepancy in the middle of a movement and showed that users are much less sensitive to a visual-proprioceptive discrepancy (distortions) than to a visual artifact such as the interpenetration of the hand in an object Burns and Brooks (2006). Interestingly, informing participants of the possibility of a mismatch was shown to influence their tolerance to discrepancies. Burns' research hence showed that 45 degrees offset in the arm rotation could be unnoticed if the participant was not previously warned about the gradual introduction of this deformation, but only of around 18 degrees when the subject was informed Burns and Brooks (2006). Independently of such factors, this tolerance has been shown to be quite useful to progressively remove discrepancies between a real hand position and its virtual counterpart, such as for recovering a gap due to virtual constraints or to rub out tracking and animation imperfections Burns and Brooks (2006).

Concerning finger movements, Krugwasser et al. observed that introducing angular distortions or temporal delays gradually reduces the SoA similarly to other effectors Krugwasser et al. (2019). Importantly, they also report that spatial and temporal distortions affect less the SoA than anatomical distortions (the limb displayed moving is not the actual limb the user moves). They thus suggest that this higher sensitivity arises from the combination of a spatial discrepancy (the moving finger is not located where the displayed moving finger is) with an anatomical discrepancy, accumulating to a larger overall conflict.

To investigate anatomical distortion, Caspar et al. used a robotic hand placed on an over-raised wood plank Caspar et al. (2015). The actual participant's hand was located just below, and both real and mechanical hands were placed above a physical keyboard. Through a mix of conditions where the motion was either congruent or incongruent (by transposing the movement of the index finger to the little finger), the authors evaluated participants' SoA. They observed that having a congruent mechanical hand leads to an SoA similar to the one experienced for the real hand while introducing the anatomical conflict significantly reduced it. They thus concluded that matching the effector to achieve an outcome is a strong factor influencing the agency's judgment, as it links with the sense of embodiment for that effector.

Using a VR display apparatus, Salomon et al. further investigated the link between the swapping of finger motion and the impact on the self-attribution of the performed movement Salomon et al. (2016). They also report the importance of embodiment for the judgment of SoA and, of primary importance for our question, they further showed that participants' accuracy was strongly affected by the movement they had viewed when asked to judge which movement they performed. This conflict elicits the possibility for self-attributing finger-swapped actions in VR.

In fact, the seminal work of Logan et al. previously demonstrated the possibility of a cognitive illusion of authorship, without VR, by asking skilled typists to type words on a computer while the visual feedback was either automatically corrected for typos or with inserted errors Logan and Crump (2010). Their results show that typists typically took credit for correct output on the screen (i.e., interpreting corrected errors as their correct responses) and, more interestingly, that typists, who were unaware of the possible introduction of mistakes by the computer, also blamed themselves for inserted errors, considering the visual output resulted from their action.

Those observations seem to corroborate the idea that the cognitive illusion of authorship could be manipulated in VR such that participants would self-attribute a correction or a mistake introduced in VR. However, it is not known if the real-time feedback of the error (i.e., the participant does not wait for the end of the word to get the typed word displayed on the screen) would prevent such expectations from happening.

It has been shown that a continuous distortion introducing a spatial discrepancy between the real (hidden) and the virtual (visible) arm in a reaching task is rather well-tolerated Galvan Debarba et al. (2018); Porssut et al. (2019); Porssut et al. (2021). More specifically, participants still report being the agent performing the action despite a relatively large distortion, typically when it helps them to reach a goal (around +2dB change in movement's speed in the study from Debarba et al. Galvan Debarba et al. (2018)) as opposed to when it prevents them from doing so. This tolerance for amplified or reduced movement cannot be interpreted solely as a limit in detection threshold, as it is influenced by other factors linked to the achievement of a task and to a more global SoE (Sense of Embodiment). Therefore, the authors revealed that distortions helping users were more accepted than distortions hindering the movement and that those distortions can be thus used to help (or penalize) users reach their goals. It can thus be expected that even stronger distortions, such as changing the motion of a body part for another one, could also be tolerated, although it is not yet proven.

Jeunet et al. evaluated three aspects of the SoA through fingers' animation manipulations as viewed in the first-person perspective in an HMD Jeunet et al. (2018). In this experiment, the authors manipulated the priority principle (i.e., the intention immediately precedes the action) by introducing temporal delays, the consistency principle (i.e., what is expected to be observed is observed) by swapping finger motion, and the exclusivity principle (i.e., one is the only apparent cause of the outcome) by randomly animating the hand. In line with former literature, they confirm a decrease in SoA when any manipulation was introduced, with the lowest agency score when consistency was altered (i.e., finger swaps). Interestingly, they also observe a correlation between the agency score and the level of immersion in VR, outlining the mutual interaction between immersion conditions and the level of SoA. However, these observations were made for isolated movements (not specific to fingers) independently of the execution of a goal-oriented task.

Overall, these studies show that introducing finger swaps reduces the SoA and that visual feedback tends to dominate over motor perception during these conflicts. However, it is unknown if, as for movement distortion in a reaching task, this would still apply to goal-oriented tasks. More specifically, it could be expected that helping or hindering the participant would influence their SoA differently for finger-swapped actions (as is the case for Debarba et al.'s reaching study Galvan Debarba et al. (2018)).

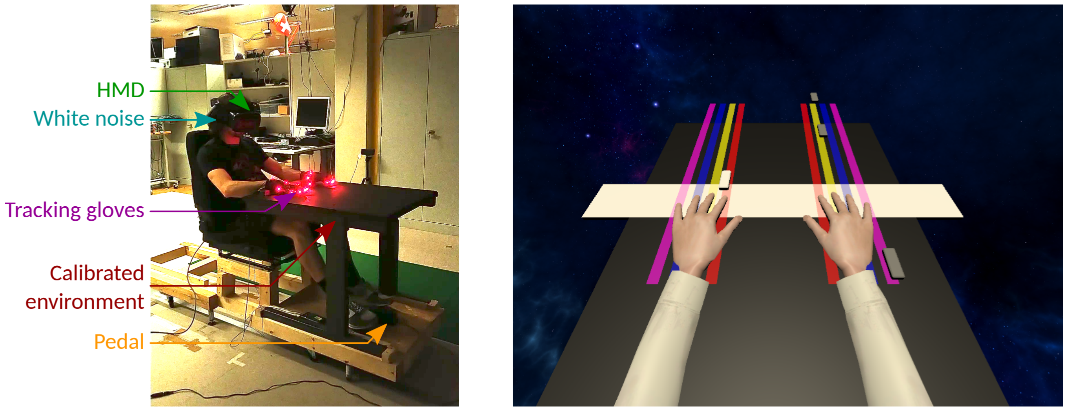
Thus, the present study evaluates the impact of finger swaps during a goal-oriented situation through a challenging VR game in which participants have to validate buttons with fingers. More specifically, this paper analyzes whether participants would detect these anatomical swaps in two contexts: without and with SE (spontaneous errors). In the context without SE, we assess the condition of error introduction (**EI**: the subject does the right action, but the swap prevents the user from validating the button), whereas, in the with SE context, we assess the condition of error correction (**EC**: the subject makes SE, and the system corrects for them).

3.2 Setup

The VR apparatus used for this experiment involves hardware and tangible objects as well as a representation of the user's avatar inside a 3D simulation running with Unity3D 3D (2019) (Figure 3.1a). To study finger swaps and to support the avatar's embodiment, the visible parts of the body are animated thanks to a Mocap (Motion Capture) system and an animation pipeline. As finger swaps must be introduced, the hands' animation pipeline is adapted to allow the permutation of fingers' motions as illustrated in Figure 3.2. The details of the swap implementation are available in Table C.1.

The Mocap system is a Phasespace Impulse X2 PhaseSpace (2019). This tool converts markers (red LEDs) attached to the glove (Figure 3.1a) into 3D points in space. As optical Mocap is sensitive to visual occlusions, in particular for fingers tracking, we used the occlusion recovery process from chapter 2. For animating the avatar, marker positions are fed to analytical IK (Inverse Kinematic) algorithms (Table C.1). Lower body parts are not visible (under the table) and thus not animated. Participants are immersed in VR with an HTC Vive Pro Eye HMD (Head-Mounted Display) and see their avatar body in first person PV (Person Viewpoint).

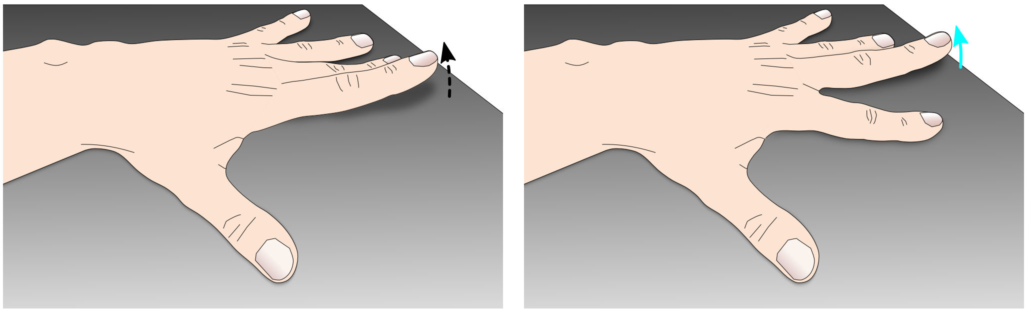
Finally, to allow the user to report an event or validate steps, an *M-Audio SP-2* pedal (connected via an Arduino Uno Arduino (2019) to the computer) is placed under the participants' foot. This system detects pedal press when the pedal reaches its mid-travel, triggering a falling edge detection on the microcontroller.



(a) *Participants are comfortably seated in front of a table that is calibrated and replicated in the virtual environment. They are immersed in VR using an HMD. An active tracking solution with gloves was used to acquire fingers' motion in real-time. According to instructions, participants press a foot pedal to report some specific events.*

(b) *When participants do the task, they see height vertical colored lines on the virtual table: one per finger except for thumbs (four on the left and four on the right). Little white buttons are sliding down along the lines and eventually pass above a finger. The goal is to lift the corresponding finger to validate the buttons in the white area. When validated, the button disappears. In this illustration, the subject should be ready to lift the left index as the button is about to pass over it.*

Figure 3.1 – *Experimental context*



(a) *Subject's real movement from the real source finger*

(b) *What the subject sees, i.e., the displayed destination finger*

Figure 3.2 – *Schematic illustration of a swap in fingers' motion. The original motion of the index (i.e., the real motion from the user, the dashed arrow) is redirected onto the middle finger the user sees (i.e., the displayed motion, the cyan arrow).*

3.3 Task

A gamified finger-movement task was implemented in order to provide participants with a stimulating and challenging task, for which the level of difficulty can be adjusted to maintain an overall success with occasional SE (spontaneous errors). While playing the game, participants are also asked to perform a perception task, consisting in detecting if they noticed the animated finger was not the same as the finger they moved.

Participants' gaming task is to hit buttons that move downwards towards them as they pass over their fingers within the validation area (see Figure 3.1b). When successfully hitting a button, a short validation sound is played (0.235s) and the button disappears. The movement chosen is a lift to ensure that, except for the actuated finger, all the other fingers remain static owing to the table contact. The challenge of the task comes from the difficulty of following the activity on both sides (for the left and right hands) and for keeping the pace as the speed progressively augments during the game.

The speed of the button is automatically and smoothly adjusted by the system to maintain the experience's flow and the difficulty of the task: When the subject performs correctly, the speed continuously increases so that at the highest speed, the subject cannot cope with the game's pace and makes mistakes (missing buttons or mixing fingers). Conversely, when the subject makes mistakes, the speed is reduced drastically to avoid overflowing participants.

Buttons are randomly distributed on each line, and the distance between those buttons remains consistent. The vertical lines laterally follow fingers' positions to ease the task by reducing the amount of attention required and the physical fatigue (movements are not physically guided like on a piano). No physical touch is simulated, and no scores are registered. The game ensures there is always only one button to hit at a time. Therefore, to prevent incidental multiple fingers lifts, the game is interrupted when more than one finger is actuated (followed by a reset of the table with buttons spawning from the beginning of the table).

The important specificity of this game for our experimental manipulation is that the machine will decide at some points to introduce finger swaps (Figure 3.2), and participants are asked to press the foot pedal when they detect such an event. Subjects are informed that they have two seconds to react after they see a finger swap (during which the system cannot introduce more finger swaps) but are not specifically asked to press the pedal as fast as possible. During the experiment's tutorial, participants are trained to recognize such swaps (they must test finger swaps at least three times per hand). Once the pedal is pressed, the game immediately stops, and participants report their confidence level about their perception of the swap on a discrete scale within $[\![0, 10]\!]$. (N.B., To reduce bias while answering the question, the scene is made empty, and the selection is made by maintaining the selection cursor, attached to the gaze direction, in one of the eleven values). Zero means 'I am not sure that the machine introduced a swap', and ten means 'I am sure the machine introduced a swap'. Once the value is validated, the questionnaire disappears, and the game restarts with new buttons at the top of the table.

3.4 Implementation

3.4.1 Speed regulation

To provide an environment maintaining the flow of the experience Csikszentmihalyi and Csikzentmihaly (1990), the game's speed is continuously adjusted through a system inspired by proportional integral derivative controllers (PIDs) aiming to ensure a sufficient amount of SE (targeted value set in the algorithm loop: 10% of the total amount of button press). Preliminary practice and speed assessment sessions are used to establish a reference speed for each user. Speed is contained between $0.35m/s$ and $1.5m/s$, and the acceleration is capped between $+0.05m/s^2$ and $-0.15m/s^2$ to avoid yanks and surprise effects. Those values were assessed with a few pilots to ensure that the game remains engaging, not too demotivating, and to ensure that the needed minimal count required for the analysis could be reached within a 1h long session for the subject.

Automatic introduction of finger swaps

To decide when a swap should be introduced, the system must first detect when a finger is moved (lifted) by the participant. A calibration process inspired by the work from the finger animation calibration (Figure 2.4) was used to store all fingers' vertical position reference when in contact with the table. Then, an offset hysteresis filter continuously compares fingertips' vertical positions to the reference to detect which finger is moved. Before the filter raises this event, no finger swaps can be introduced. This is also used to detect SE and tell the speed regulator when the user made a mistake.

When a SE occurs (i.e., the moving finger is not on the button's line), the system randomly decides to trigger (or not) a correction of the movement. If the correction is triggered, the expected finger movement is swapped with the wrongly moved one. To distribute these error corrections over time, the algorithm enforces that every consecutive chunk of six decisions is balanced (i.e., it triggers randomly three among six cases of corrections and lets the three other movements uncorrected). Of note, pilots showed that six was enough to make sure participants could not predict a pattern. Maintaining a low chunk size ensures that conditions are continuously balanced instead of accumulating unbalance that would need to be fixed toward the end.

When the participant correctly hits buttons without SE, the system introduces five swaps per 100 trials. In such **EI** condition, the swapped finger is randomly chosen.

3.5 Experimental design

The study was undertaken in accordance with the ethical standards as defined in the Declaration of Helsinki and was approved by our local Ethical commission. No minors were involved in this study, and consent was collected on written sheets before the beginning of the experiment. The protocol for this experiment is presented in Figure 3.3 and detailed in the following sections.

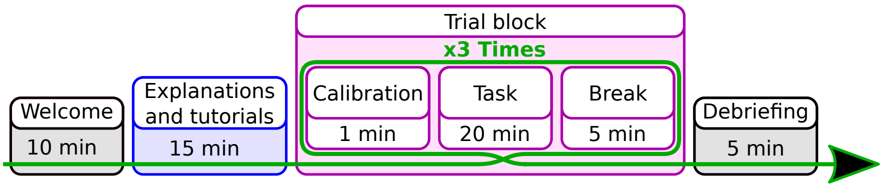


Figure 3.3 – *Timeline illustration of the experimental protocol of the experiment.*

Participants are welcomed before giving their informed consent and filling out a demographic questionnaire (detailed in section C.1) and are equipped with tracking gloves (Figure 3.1a) and HMD. Once the experiment starts, the room's lights are turned off, and all the explanations are given in VR to ensure that all participants receive the same instructions.

Explanations and tutorials

The first step explains to the participants how to calibrate their hands to provide them with virtual hands during the tutorial. Then the game is explained to the participants and they can practice briefly (20 seconds). This is followed by a speed-assessment practice run during which the automatic speed adjustment of the game is monitored: the speed increases as the participant is successful (and reduces upon mistakes), enabling the system to store a personalized initial speed for each participant. The experimenter also observes the participants' ability to do the task.

Participants then undergo a multitask-assessment practice run to ensure they can do both the game and the experimental tasks simultaneously. Here, the goal is to play the game and press the pedal when a validated button turns red (500ms) instead of disappearing. Of note, the confidence question is introduced here to stagger instructions and ease comprehension, therefore, once the pedal is pressed, subjects give their confidence level in the observed presence of a red button (similar to the real task).

To ensure subjects understand the expected phenomenon to be reported, they are explained finger swaps and can try those by lifting their fingers while the swap alternates between enable/disable to highlight the effect (with an indicator displayed in the scene). At this point, subjects are invited to ask questions to ensure all instructions are fully understood.

Finally, participants go through a dry run to ensure that everything works and that the participant can perform the task correctly. At this point, a continuous white noise sound is added (to prevent the participant from hearing sounds from the real environment) and the participant is ready to perform the task for this study.

Of note, warnings are automatically raised and displayed to the user when multiple fingers are lifted simultaneously or when the subject moves his hand off the table. Also, the experimenter can trigger a message to stop the experiment in case of need.

Trial block

During a trial block, participants perform the game task (hit buttons) and the experiment task (press pedal when detecting a swap) until 20 occurrences of each condition are reached (Table 3.1). Each trial block is followed by a break when participants can remove the HMD, gloves, and leave the chair before re-calibrating their hands for the next trial. On average, trial blocks lasted roughly 18min and presented 2788 buttons. Among those buttons, on average 38 are **EI**, and 50 are SE of which 31 are **EC**.

At the end of the experiment, feedback is considered, and participants receive monetary compensation for their time. The average session duration was designed to last 1h30 for roughly 3000 buttons presented.

3.6 Hypotheses, measurements, and analysis

3.6.1 Formal hypotheses

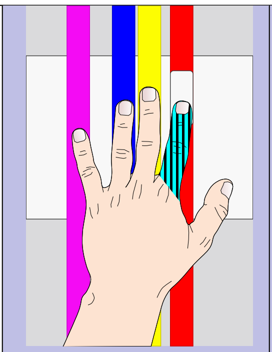
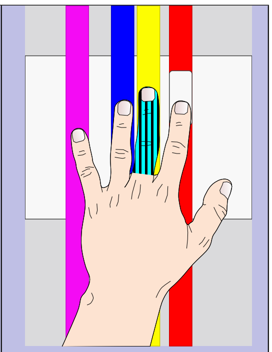
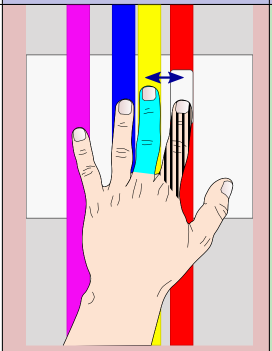
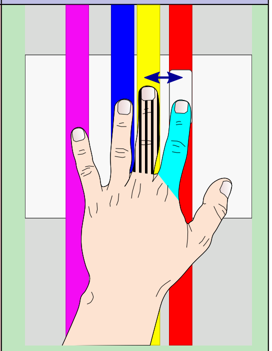
The translation of the research question through the experimental setup can be formalized with the following hypotheses with the different conditions described in Table 3.1.

The experimental conditions for our study are **EI** (error introduction) and **EC** (error correction). The **EI** condition represents the case where a participant moved the correct finger, and the system remapped this movement onto another finger, thus preventing the participant from succeeding. Conversely, the **EC** condition represents the case where the participant moved the wrong finger, and the system remapped this movement to the finger facing the button, thus helping the participant to succeed. Other conditions represent congruent visual feedback and are used to help maintain the game's flow.

The buttons' speed regulation is expected to push participants to make approximately 10% of SE over the total number of cases. The experimental system then introduces **EC** conditions for half of the detected SE cases and introduces **EI** conditions for 5% of the cases.

To elicit whether or not the motor conflict from finger-swaps Salomon et al. (2016) could lead to the self-attribution of finger-swapped actions in immersive VR, and to assess if the direction of the distortion (i.e., hindering or helping) at achieving a goal-oriented task Logan and Crump (2010); Galvan Debarba et al. (2018) affects the former self-attribution, we formulated the following hypotheses:
H1 - Introducing a swap in fingers' motion to prevent users from reaching the goal is more rejected than a swap helping to reach the goal.
As Henmon et al. showed that high confidence levels are correlated with faster reaction times Henmon (1911), our second hypothesis, expecting a higher confidence level at detecting penalizing finger swap compared to the ones helping the user, was extended with a shorter reaction time for the former condition, and therefore formulated as:

Table 3.1 – *Experimental conditions: Hatched areas indicate which finger is (sometimes wrongly) moved by subjects while cyan areas indicate the finger which is actually animated by the system. Illustrations represent an example case when a subject should move the index to validate the button (in the game, buttons can arrive on any vertical line). The arrow in the swap conditions represents the swap count. Here its value is one as swapped fingers are next to each other.*

	No SE	SE
No swap		
Swap	 Error Introduction	 Error Correction

H2a. - Introducing a swap penalizing the user is reported with a higher confidence level than a swap helping the user
H2b. - Introducing a swap penalizing the user is reported with a shorter reaction time than a swap helping the user.

3.6.2 Measurements

As per the instructions, participants press the foot pedal when they observe a finger swap. In such an event, the game immediately stops, and the system stores the pedal press time, allowing to measure (post-analysis) the amount of time elapsed since the previous experimental condition (**EI** or **EC**).

Of note, the condition is ignored for further analysis if the participant did another mistake in the time between the introduction of the experimental condition and the pedal press. This applies for SE as well as for conditions followed with a warning (e.g., multiple fingers moved simultaneously).

In addition, the amount of swap (`swp`), referring to the number of hops between the finger moved and the animated one, is also stored for post-analysis (a swap count of 0 means there is no swap, a swap count of 1 means that the swapped finger is the direct neighbor of the moving finger such as on Figure 3.2, and so on).

Finally, after a pedal press, a questionnaire asks the participant about their confidence in the observation of a finger swap.

3.6.3 Statistical analysis

As our hypotheses only concern **EI** and **EC** conditions the dataset was filtered to remove other conditions (no mistake without error introduction and mistakes without error correction). The experimental design considers that a pedal pressed under no swap condition with no spontaneous error is assumed to be attached to the previous experimental condition unless the time window is closed. The analysis was conducted using R.

We expect that introducing a swap in fingers' motion to prevent users from reaching the goal will be more rejected than a swap helping one to reach the goal. This can be formalized as Equation 3.1.

$$\mathbb{P}(\texttt{pedal\_pressed}|\ \textbf{EC}) < \mathbb{P}(\texttt{pedal\_pressed}|\ \textbf{EI}) \tag{3.1}$$

Therefore, a mixed model providing the pedal pressed outcome (`pp`) as a factor of the fixed effects of the amount of swap (`swp`) and SE (`se`) was used to fit our filtered dataset (3256 points) to assess this hypothesis. The logit function, defined as $\texttt{logit} : x \mapsto \ln\left(\frac{x}{1-x}\right)$, was used as the outcome `pp` is binary (the pedal is either pressed or not within the two second time window after a condition). Plot observations hinted at a square factor for the `swp` predictor and did not highlight interaction factors. Therefore we considered the outcome as a mean (I) plus the impact of `se`, `swp` and $\texttt{swp}^2$ (Equation 3.2).

$$\mathbb{P}(\texttt{pp}) = \texttt{logit}\left(I + \beta_1 \cdot \texttt{se} + \beta_2 \cdot \texttt{swp} + \beta_3 \cdot \texttt{swp}^2\right) \tag{3.2}$$

Random effects from the subject id, the elapsed time since the last condition (to relate the speed of actions), and the number of minutes elapsed since the experiment began (to relate fatigue) were also considered (although not displayed in the formula).

The model's fitness was assessed using the residual analysis using R's `DHARMa` package with a 0.18 p-value for the KS test of deviation, 0.66 for the dispersion test, and 0.77 for the outliers test.

We expect that introducing a swap penalizing the user would be reported with a higher confidence level than a swap helping the user through a higher confidence level and a shorter reaction time. This is equivalent to Equation 3.3 and Equation 3.4.

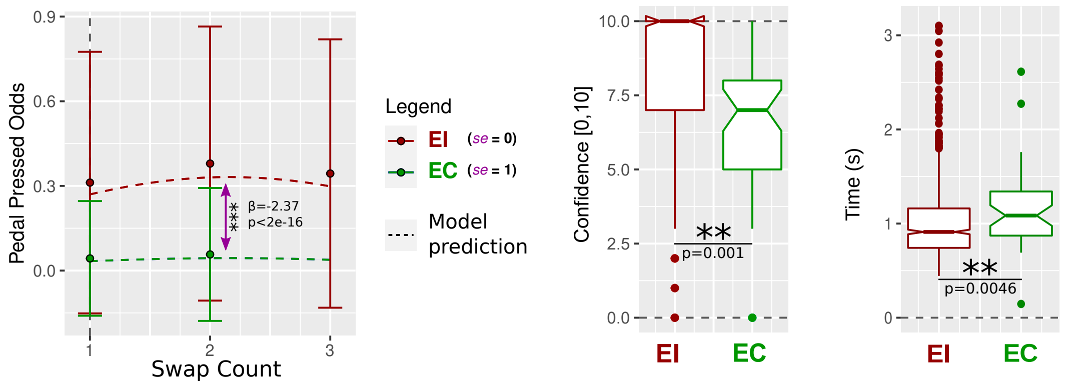
$$\mathbb{E}(\texttt{confidence\_level} \mid \textbf{EC}) < \mathbb{E}(\texttt{confidence\_level} \mid \textbf{EI}) \tag{3.3}$$

$$\mathbb{E}(\texttt{response\_time} \mid \textbf{EI}) < \mathbb{E}(\texttt{response\_time} \mid \textbf{EC}) \tag{3.4}$$

Only conditions with a pedal pressed were retained (783 entries) as the confidence level questionnaire and the reaction time (time elapsed between the button's validation and the pedal press) are only defined after a pedal was pressed.

A two-sided two-sample median permutation test (with 50 000 iterations) was used to compare confidence levels and reaction time medians between both groups (**EI** and **EC**) followed by the one-sided test to retrieve the direction of the difference.

3.7 Results



(a) *H1 - Probability of pressing the pedal after the introduction of a finger swap, for swap count of 1 (e.g., swap index finger with middle finger), 2 (e.g., index finger with ring finger), and 3 (index finger with the little finger, this occurred only for the* ***EI*** *condition, i.e., there is no data-point for* ***EC****). Plots of the mixed model prediction (dashed lines) compared to all subjects' data. The pink arrow highlights the significant difference resulting from the* `se` *fixed effect.*

(b) *H2a. - Effects of* ***EI*** *and* ***EC*** *conditions in terms of confidence levels. The* ***EI*** *condition presents a very significantly higher confidence score than the* ***EC*** *condition.*

(c) *H2b. - Effects of* ***EI*** *and* ***EC*** *conditions in terms of reaction times. The* ***EI*** *condition presents a very significantly shorter reaction time than the* ***EC*** *condition. (Two additional outliers at 9.1s and 3.9s are out of the plot for* ***EI****).*

Figure 3.4 – *Plots of the results from the Analysis section*

3.7.1 Demographics

20 participants aged between 18 and 44 years old (median: 24, std: 5.62), including 10 women, participated in this experiment. One participant stopped the session due to the difficulty of wearing the HMD. Most participants came from *anonymous* area and were all students or people working in academics. One participant was left-handed. The demographic questionnaire indicates that participants mostly experienced VR a few times, were healthy and were comfortable with typing.

Over 52,977 buttons pressed 3,256 entries were retained to assess the first hypothesis (sum of all swap conditions, in red and green in Table 3.2). The assessment of the second hypothesis used the subset of the filtered dataset where the pedal was pressed, a subset composed of 783 entries (sum of all swap conditions where `pp` $\neq 0$).

Table 3.2 – *Detailed count of each case occurrence. The No Swap cases (blue) do not count for the experimental condition evaluation as they represent the large majority of 'normal' events when playing the game. The* `pp` *columns represent the sub pedal pressed count per condition while the %* `pp` *represents its percentage share. In total, on the experimental conditions, 24% of the swaps were noticed with a pedal press (sum of* `pp` *over totals from red and green cells from the last line).*

		No SE (≃ 95%)			SE (≃ 5%)		
	`swp`	Total	`pp`	%`pp`	Total	`pp`	%`pp`
No swap	0	47955	20	0.04 %	1766	238	13.48 %
Swap	1	1187	370	31.17 %	1021	44	4.31 %
	2	567	215	37.91 %	35	2	5.71 %
	3	442	152	34.39 %	4	0	0 %
	Total	2196	737	33.6%	1060	46	1.63%

3.7.2 Finger swap detection

Fitted coefficients for the model (defined in Equation 3.2) are displayed in Table 3.3.

Table 3.3 – *Predictors values for the fitted mixed model. We can observe the very significant impact of the* `se` *predictor on the observed outcome.*

Fixed effect	Equation factor	Estimate	p-value
Intercept	I	-1.72	$2.54 \cdot 10^{-05}$
`se`	β_1	$\mathbf{-2.37}$	$< 2 \cdot 10^{-16}$
`swp`	β_2	0.94	0.0322
I(`swp`^{2})	β_3	-0.28	0.0536

The p-value associated with the `se` predictor coefficient is low ($p < 2 \cdot 10^{-16}$); thus, the odds of the observed effect from this predictor being due to chance are almost null. Since the `se` predictor coefficient is negative (and the `logit` transform function is an increasing function), the odds of pedal pressed are significantly lower when `se` = 1 (**EC**) compared to when `se` = 0 (**EI**) given the model used (Equation 3.2). This translates into the red curve of **EI** being above the green one of **EC** in Figure 3.4a. Finally, the R^2 was measured at 0.21 (interpreted as low Cohen (1988)) for the whole model and at 0.16 for the only effect size from the `se` factor (also interpreted as low). Therefore, introducing a swap in fingers' motion to prevent users from reaching the goal will be more rejected than a swap helping one to reach the goal, hence validating our first hypothesis.

Additionally, we observed that the amount of swap (i.e., `swp`) also significantly impacts the odds of having a pedal pressed. A possible explanation could be that swaps of neighboring fingers are harder to observe than those from distant fingers (e.g., index and pinky).

3.7.3 Confidence level and reaction time analysis

A significant difference was measured between the two samples' median through the two-sided test (a 0.001 p-value). Cohen's D effect size for the raw influence of the self-error on the confidence level was measured at 0.87 (high). The confidence level median for the **EI** is 10 with a mean of 8.39 and a standard deviation of 2.19. In comparison, for the **EC**, the median is at 7, the mean is 6.46 and the standard deviation is 2.66. This difference is oriented with a higher median for the **EI** condition compared to the **EC** condition (a 0.001 p-value). Those results are plotted in Figure 3.4b and validate the first part of our second hypothesis (H2a., Equation 3.3).

The same procedure yields a 0.00456 p-value for the two-sided test, revealing a significant difference between both samples' medians, with a shorter reaction time for the **EI** condition compared to the **EC** (0.0042 p-value, Figure 3.4c) which validates H2b. (Equation 3.4). Cohen's D effect size for the raw influence of the self-error on the reaction time was measured at 0.24 (low). The reaction time median for the **EI** is 0.91s with a mean of 1.02s and a standard deviation of 0.51s. In comparison, for the **EC**, the median is at 1.09, the mean is 1.15s and the standard deviation is 0.41s.

It is to be noted that in the **EI** condition, many outliers were observed. A possible explanation might be the fact that in case of doubts, people might take a bit longer to decide whether a swap was introduced or not, and in such a case, they would more easily recognize an error introduction and press the pedal while in the other case, they might just accept it.

This does not contradict the relation from Henmon (1911) and, together, those results support our second hypothesis.

3.8 Discussion and future work

Based on prior results from the literature (Galvan Debarba et al., 2018; Porssut et al., 2019), we expected that introducing finger swaps in an engaging game would induce a different behavior when the swap helps rather than hinders participants in their task. More specifically, we expected lower odds of perceiving the swaps when helping compared to when hindering the user. Additionally, we expected that the latter case would be detected with higher confidence. This study was designed to answer those two questions by providing subjects with an engaging task performed in immersive VR, with an avatar following users' movements. Confidence levels and reaction times were measured to assess the subject's confidence in the reported swap.

Results validate our two hypotheses: participants are less sensitive with lower confidence levels and lower odds of detecting when swaps help them than when they hinder their movements.

Up to an offset of $\sim$55%, we observed the same detection rate of an alteration of the user's action as in the work from Logan et al. (Logan and Crump, 2010). In their study investigating the self-attribution of corrections/inserted errors, authors reported a detection rate of altered actions of approximately 85% for an inserted error against 55% for a corrected error. In comparison, in our study, the values observed were 33% for error introduction to the 1.63% in the error correction, and a similar drop of $\sim$30% was measured in the perception rate between those two conditions.

Since the discrepancy we introduce in both cases (i.e., a finger swap with the same algorithm on the same game) is the same, a purely sensory-motor comparison with the visual feedback should raise the same warning for any type of swap introduced. Therefore, the comparator model for self-attribution of movement (Wolpert et al., 1995) does not fully explain the observed behavior. Our results rather corroborate the work of Logan et al. Logan and Crump (2010), showing that the authorship illusion is composed of at least two stages (referred to as the inner and outer loop), and consistent with a hierarchical error-detection mechanism.

3.8.1 Relation with agency and embodiment

Although the levels of SoA and SoE were not evaluated during this experiment, the experimental setup with its immersive technology, the self-location of the virtual avatar, and the animation of upper limbs was assumed to provide users with relatively good levels of embodiment and agency, at least comparable to those in similar experiments Salomon et al. (2016); Jeunet et al. (2018). Conversely, a pedal press at the detection of a finger swap thus indicates a disruption of the user's SoA (and probably of SoE).

Results can therefore be interpreted in terms of agency in the following way: finger-swaps in the **EI** condition are more likely to disrupt SoA than in the **EC** condition. Furthermore, considering the low probability of detection of **EC**, our results suggest that error correction with finger swap has a limited impact on the SoA.

In practice, to avoid disrupting the SoA, it is important to prevent users from noticing finger swaps, and **EI** should be avoided. A system can introduce swaps in finger motion in immersive VR in order to help participants achieve a task without them noticing (most of the time) and with a limited impact on SoA and SoE. Such results can be useful for controlling the flow in a training task and maintaining motivation (e.g., learning the piano, typing) or for compensating for finger tracking errors (i.e., in the absence of correct tracking, trigger the expected finger movement).

3.8.2 Limitations and future works

Our experimental manipulation required to place participants in a situation leading to spontaneous errors. Other approaches, with a question following each trial (as in the work from Salomon et al. Salomon et al. (2016) or Balslev et al. Balslev et al. (2007)), could not be used as participants would have constantly been interrupted, breaking the flow of the game Csikszentmihalyi and Csikzentmihaly (1990). Instead, using a method similar to the one from Kokkinara et al.'s study Kokkinara and Slater (2014), we asked subjects to self-report the introduction of finger swaps through a pedal press. Our design can thus only assess perceived swaps and cannot reveal behaviors based on non-observed finger swaps. It is indeed possible that participants deeply engaged in the game might have forgotten to report some swaps or that their attention might have been temporarily disrupted. Using eye-tracking might help reveal some unexpected behaviors and/or disentangle some conflicting cases (e.g., measuring eye saccades when a swap occurs or not). However, current HMDs do not provide fast enough eye-tracking capabilities to measure those saccades (requiring a sampling frequency to be above 500Hz) Stein et al. (2021), and knowing participants' gazes is not necessarily sufficient to relate the actual perception of the change by the user (e.g., movements can very well be perceived in peripheral vision).

Another differentiation with traditional approaches is that the participant's attention is shared between two tasks (the game with validating buttons and reporting finger swaps). Although all participants underwent multitasking training and assessment sessions, it remains unknown how much our results are influenced by their ability to perform the dual-task for the specific experimental manipulation. Further testing and evaluation of participants could be conducted to achieve a more detailed understanding of the interactions between the ability to multitask and the experience of embodiment.

Compared to the study from Burns et al. Burns and Brooks (2006), the subject's task is more complex in our case; hence, all participants had to be trained on the type of distortion to recognize (finger swaps). As a consequence, given that warning the participants about distortions influences the experiment outcome Logan and Crump (2010); Burns and Brooks (2006), it is normal to have higher odds of swap notifications in our context. It could be expected that, without previously informing participants of the possibility of finger swap, the detection rate would be much lower. To study such cases, a system monitoring brain activity with electroencephalography could be used to monitor the brain's spontaneous reactions to error, known as Error Related Potential Falkenstein et al. (1991); Gehring et al. (1993). As previously done for detecting violations of agency in VR Padrao et al. (2016); Pavone et al. (2016), it would probably be possible to directly detect the brain reaction to error correction or error introduction without interrupting the participant, and with the possibility to answer to mechanistic and neurological questions on the agency of error correction.

3.9 Conclusion

Our study shows that virtual distortions of finger movements (swaps) that help users to reach a target with their fingers are more tolerated than distortions hindering their action. This extends the previously observed effect of distortions for full arm reaching tasks in VR Galvan Debarba et al. (2018); Porssut et al. (2019), thus generalizing the observation that some carefully designed discrepancies between real and virtual body movements can be well tolerated as far as they help in achieving a goal in VR.

More specifically, our experimental setup successfully elicited the self-attribution of finger-swaps in immersive VR, with a significant difference between swaps helping or not the subject to accomplish a challenging task. Our results support the hierarchical error-detection mechanism proposed by Logan et al. Logan and Crump (2010), with inner loops taking care of the details of performance (here finger swaps) and outer loops ensuring that intentions are fulfilled, thus leading to the authorship illusion for avatar-corrected actions.

Finally, one take-home message for designers of embodied interaction in VR involving finger movements is that a system can introduce finger swaps without disrupting the SoA as long as those swaps help users in achieving the task at hand.

4 Integration of finger and full body animation with self-contact consistency

4.1 Introduction

We live through our bodies which allows us to interact with the environment. When someone wears an immersive HMD (Head-Mounted Display), the real world around disappears, and the user's physical body is no longer visible. To avoid a conflict occurring between the user's expectation of having a body and the absence of a body in the VE (Virtual Environment), which would negatively affect the user's experience (Porssut et al., 2019; Gao et al., 2020), it is required to provide a user with a virtual body, called an avatar. To allow one to embody such an avatar, the following senses need to be elicited to the user (Kilteni et al., 2012): the sense of self-location ("refers to one's spatial experience of being inside a body"), the sense of body ownership "refers to one's self-attribution of a body" and the SoA (Sense of Agency) (the "global motor control, including the subjective experience of action, control, intention, motor selection and the conscious experience of will" (Blanke and Metzinger, 2009)). It is crucial not to disrupt any of those three components to prevent breaks from occurring, which would lead to a break in embodiment (Kokkinara and Slater, 2014) that would significantly reduce the user experience.

A common way, proven to be effective at eliciting a strong sense of self-location in immersive VR is to provide the user with the full body illusion at the first PV (Person Viewpoint) (Galvan Debarba et al., 2017). This is achieved by using a tracked HMD to allow the user to have a virtual viewpoint placed at the position of the virtual head of the avatar. In the same vein, the SoA and the body ownership can be provided through the animation of a plausible human avatar animated in real-time using MoCap (Motion Capture) systems. However, MoCap systems are not perfect and can be subject to artifacts in the measured movements performed (Tian et al., 2015). Conveniently, it was shown in Burns and Brooks (2006) that humans' proprioception is relatively poor in providing good limb position feedback for static poses. Furthermore, studies on movement distortions in immersive VR showed that the visual feedback was actually more relevant and could be manipulated with amplified or reduced displayed avatar limb movements (Galvan Debarba et al., 2018; Porssut et al., 2019), to a certain extent, without having the user noticing the alteration of the movement. The same phenomenon was observed at the finger level in the previous chapter, and this was also the case for head movements (Jaekl et al., 2002).Consequently, we are pretty tolerant regarding the exactness of the provided animation.

However, it should be noted that these tolerances do not mean that the user will accept any movement that is displayed. Recent research has emphasized the stronger acuity of perceiving virtual touches when a virtual body is provided to the user (Gonzalez-Franco and Berger, 2019), and failing at providing self-contact consistency (i.e., there is a conflict between the perceived skin contact and the absence of the contact in the VE (Virtual Environment) and conversely) was shown to induce breaks in embodiment (Bovet et al., 2018). Despite the importance for the user of providing self-contact congruence, many existing approaches in the literature primarily focus on interactions with objects. As a consequence, in this chapter, we propose integrating the finger animation pipeline discussed in chapter 2 with an adapted version of the real-time body animation pipeline developed by Molla (Molla et al., 2017), specifically designed to address the issue of self-contacts consistency in real-time.

4.2 System overview

Similarly to the original approach from Molla et al. (2017), our system takes advantage of our tolerance to motion distortion. As for the original method, our solution relies on an initial calibration of both the user's body and the virtual character's one, used to determine when contacts are about to occur, and an online animation procedure as illustrated in Figure 4.1.

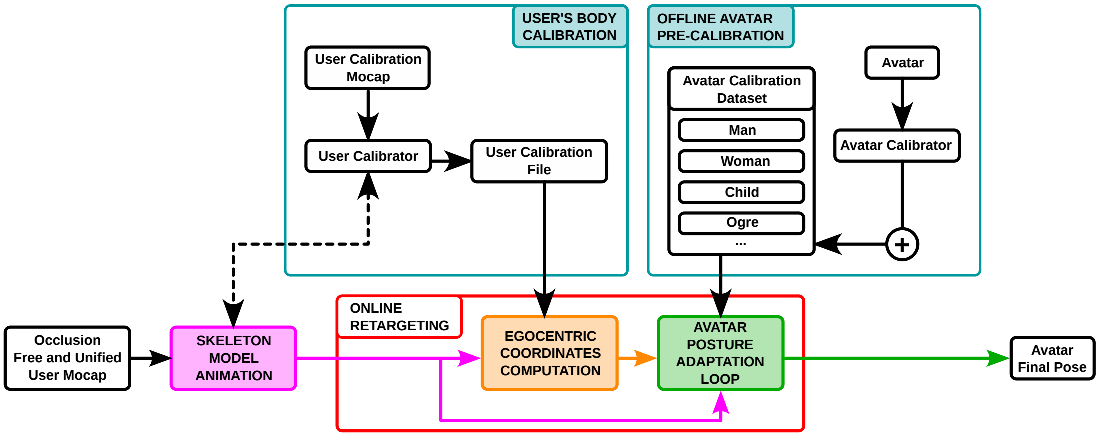


Figure 4.1 – *System overview: the upper stage on the schema represents the avatar calibration; this step can be performed once using the Avatar Calibrator, and the calibration file can be stored for future use. The middle stage corresponds to the user's calibration process. Here the user performs several gym motions and self-contacts to calibrate its virtual skeleton and approximation body. Finally, with the user's virtual skeleton calibrated and the avatar's calibration profile, the third and lower stage computes in real-time the instantaneous pose to be applied to the avatar.*

The avatar calibration needs to be performed only once and can be stored in a database. Conversely, the user calibration must be performed each time the user is equipped with the tracking setup, as the trackers are not required to be placed at exact body locations. During this calibration, the user must perform gym movements to retrieve the internal structure of the joints and touch a few specific points on the body to define a simplified crude trunk mesh representation of his body surface, as initially proposed in Molla et al. (2017). However, this method is insufficient to calibrate the fingers, and the original method was extended with an additional calibration phase for the hands and fingers. This fine-level procedure was made available to extend the gym motion calibration optionally and involves touching the body on several joint locations with a fingertip to enhance the accuracy of locating the joints' locations and limbs' radius.

Once both the target (avatar) and source (user) skeleton/body shapes are calibrated, the live performance phase is composed of three steps with: the Motion capture (to animate the model of the user's structure), the computation of egocentric normalization (to account for a normalized representation of the user pose, invariant from the user morphology), and finally, the animation is applied to the character through a gradual limb posture adaptation.

The motion capture is mostly performed with a direct application of the trackers' position on the different bones of the user's limb. The computation of egocentric normalization involves computing the coordinates of effector positions using an egocentric coordinates system in which a position is measured as the sum of the normalized contribution of vectors toward each surface element from the source structure (user's body shape, subsection 4.6.2). Finally, an adaptation loop progressively attracts each avatar's targets towards its retro-projected egocentric coordinates on the virtual character (subsection 4.6.3) to produce the final avatar's pose. The half-plane used in the original animation from Molla et al. (2017) was replaced with an adapted IK taking as an input both the current limb kinematic chain and the original orthogonal vector defining the limb flexion's axis. In addition, a second animation convergence loop was added to handle the animation of hands and fingers.

The whole pipeline relies on the transformed inputs from the SteamVR and PhaseSpace environment, with the occlusion recovery stage from chapter 2 applied upstream. Therefore, the user's calibration and online retargeting pipeline described here assume that the input is complete and reliable.

4.3 Setup

Animating an avatar requires identifying the current pose of the user. To allow our pipeline to be used with a simple consumer-grade setup, and unlike in the original approach from Molla et al. (2017), the motion capture system for the body tracking is composed of Vive Trackers Figure 4.2 Vive (2022).

However, those devices are too bulky to be placed on fingertips, preventing finger motion tracking. To integrate the finger layer in the animation pipeline, we re-used the technology proposed in

Figure 4.2 – *Vive Tracker 3.0 are consumer-grade devices whose 3D localization can be retrieved through SteamVR. Those trackers are well suited for the gaming experience, for tracking limbs (e.g., feet, knees, etc.) or objects to interact with (e.g., tracking a dummy gun). However, their dimensions prevent them from being placed on each finger. Picture sourced and edited from https://www.vive.com/fr/accessory/tracker3/*

chapter 2 relying on the PhaseSpace (PhaseSpace (2019)) tracking with the occlusion recovery pipeline to acquire missing information from hands and fingertips. LEDs were placed on new gloves Figure 4.3 to track fingertips positions, and, to reduce the jitter due to the glove's flexibility, a wooden support was added to hold the three LEDs defining the hand's rigid body in place.

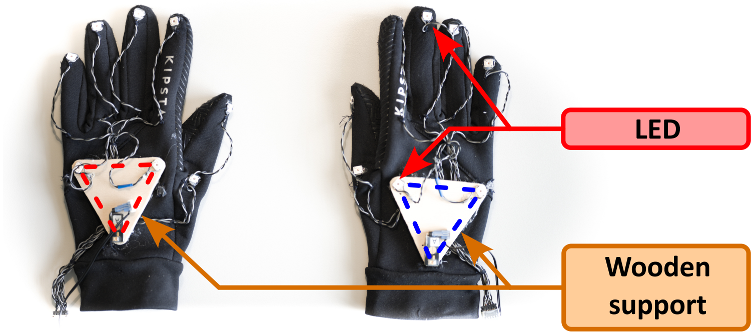


Figure 4.3 – *The gloves' black texture helps to reduce light reflections from the tracking LED to enhance tracking. Each fingertip has a LED to track its position in the 3D space. The wooden support provides a rigid body reference to reduce the LED lateral motion due to the gloves' flexibility.*

The complete set of trackers is illustrated in Figure 4.4 and comprises twelve Vive Trackers and two motion capture gloves. Motion capture gloves can be replaced with Vive Trackers for a configuration without finger-tracking capabilities.

Using several tracking systems requires consistency of both tracking spaces; therefore, the two body level referential must be realigned. Our procedure to perform such a realignment is to place a tracker on three reference points printed on the room's floor: The origin (0,0,0), front (0,0,1),

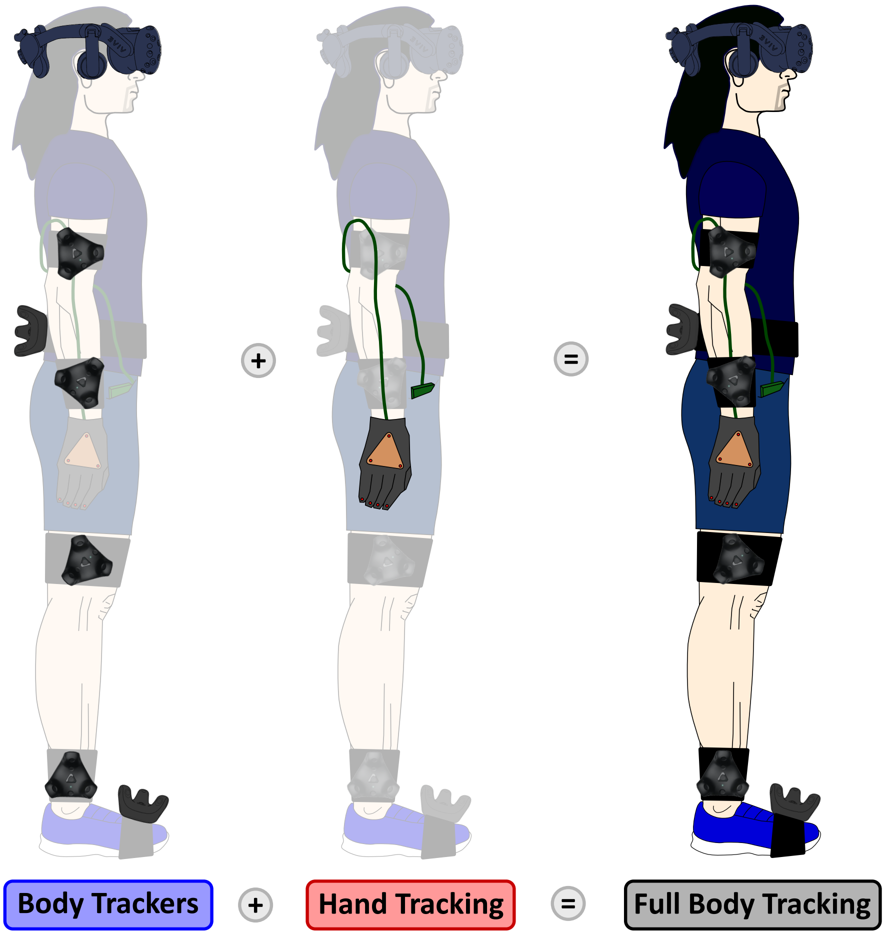


Figure 4.4 – *The setup involves a mix of Vive Trackers 3.0 with homemade tracking gloves, therefore mixing tracking solutions. Vive Trackers are wireless, reducing the user's constraints, while the tracking gloves are still wired to receivers that users keep in their pockets.*

and right (1,0,0) which is then used to compute the input's system positional and rotational offsets. Then, those computed offsets are inverted and directly applied to the input through an abstraction input layer that feeds the animation pipeline.

4.4 Users' body calibration

Similarly to the approach from Molla et al. (2017), our limb calibration uses CoR (Center of Rotation) computation starting from the effectors and tracking back towards the trunk with the difference that a second pass is used to increase the accuracy of joints positions and to measure limb radius.

As our pipeline also involves the animation of fingers, the calibration process starts with the calibration of hands and fingers.

4.4.1 Hands and fingers calibration

Calibrating the user's hand is a task that involves measuring many parameters, and rather than repeating similar poses several times, we chose to measure multiple parameters simultaneously. The first pose consists in placing both hands' palms in contact with each other (Figure 4.5) to calibrate:

- the hands' referential as a regular rigid body tracked using three LEDs to determine its position and rotation in space
- the hands' surface plans: Each position from each LED of one hand is averaged with its opposite position from the other hand, and those averaged points are used to fit a plan that defines each hand's palm surface measured in each hand's referential.
- fingers radius: Knowing the surface plan in each hand's referential and the position of the LED on top of the finger, the finger radius is computed as half the length between the LED position and its projection on the hand's palm surface plan.
- fingers extended position: Local extended fingertip positions are stored in the hands' referential to be used later as a reference to compute the angular rotation to apply on each finger.

The critical information required in the hand structure for its animation is the location of its joints (i.e., the wrist and the fingers' proximal root joints). The wrist position is measured by successively placing the index fingertip from the opposite hand on top and below the wrist (Figure 4.6a) to calibrate its position as the mean of the two measured positions in the hand's referential.

Unlike the method proposed by Aristidou (2018), our approach does not require precise placement of tracking LEDs on the pinky and index finger base joints. However, the counterpart is that those finger base joints must be calibrated.

Our initial tests showed that recording extended finger motion to calibrate the finger base joint yielded unrealistic data. Therefore a manual calibration method was designed to calibrate the finger base's joints precisely.

Once the wrist is calibrated, each finger's base location is measured by placing the opposite index on each finger's base joint. The joint base location is then computed as the measured LED position projected on the hand's palm surface, on which the radius of the finger is added toward the top of the hand. Digits are then initialized as capsules with a radius corresponding to the measured finger radius and placed in the alignment between the joint base position and the extended fingertip position (green bones in Figure 4.6a). As bones' motion within the hand is relatively small, a simplified structure as a rigidly attached bone to the hand preferential is used to attach the fingers' base joint (i.e., proximal's root) to the wrist.

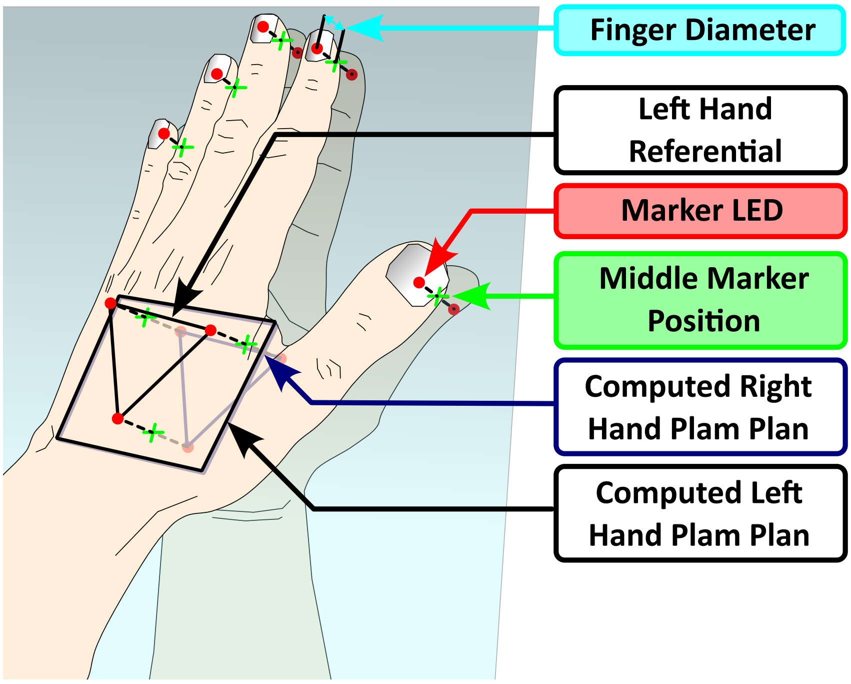


Figure 4.5 – *The middle plan that separates both hands is illustrated in cyan. Its location is computed locally to both hands referential (i.e., computed left-hand palm plan and computed right-hand palm plan) so that each hand model knows where is its contact surface. This pose also determines the fingertips' radius and local positions of extended fingers used in the animation stage to animate fingers' kinematic chains.*

The last information to identify about the hand is the crude approximation of its palm surface. This information is measured as the projection of the other hand's fingertip on the palm surface (Figure 4.6b).

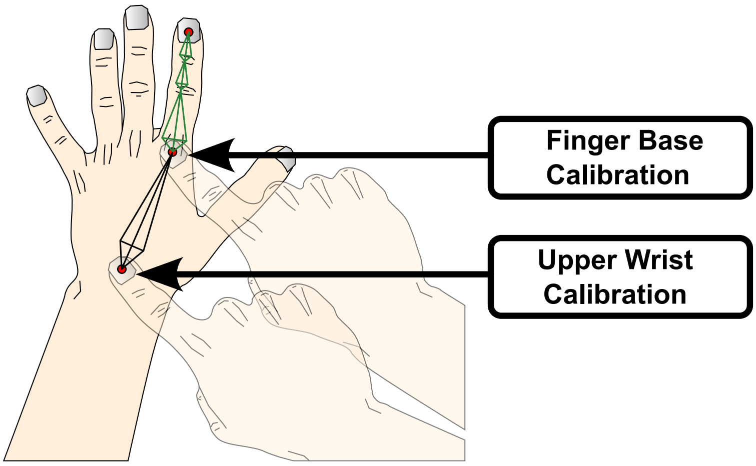
4.4.2 Feet calibration

Similarly to the hands, feet embed a referential: the tracker attached to each foot, a crude mesh representing the contact surface under the sole of the foot, and an anchor joint.

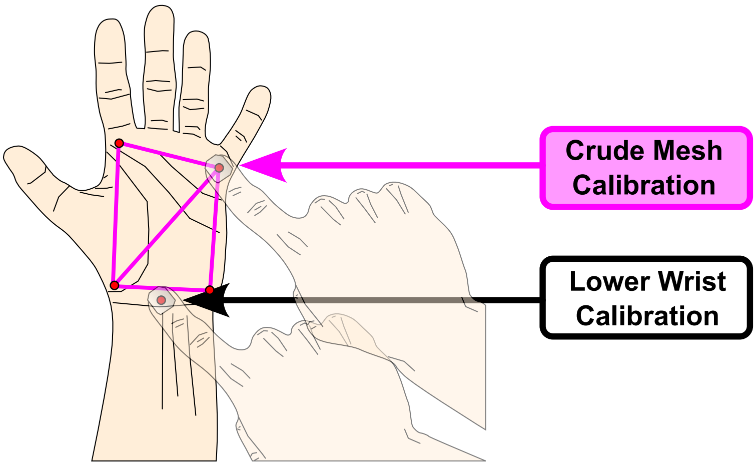
The local positions of the contact surface are measured as the fingertip's projection to the floor and then stored in the foot's referential (c.f., Figure 4.7). The ankle's local position is measured by placing the fingertip on both sides of the ankle and averaging the two positions.

4.4.3 Limb calibration

Once the effectors are calibrated, the next step is to retrieve the user's skeleton structure by progressing proximally toward the trunk.



(a) *Finger base calibration*



(b) *Hands' palm's crude mesh calibration*

Figure 4.6 – *To calibrate a point, one must place the other hand's fingertips (the index by default) on top of the point of interest (e.g., joint) The order in which the user calibrates points is irrelevant, as all calibration points are first stored. The actual computation can be triggered later, hence avoiding flipping the hand several times to calibrate the wrist, surfaces, and finger base joint.*

Therefore the next step is to calibrate the four limbs linking the effectors to the trunk: the arms and the legs.

Limbs are kinematic chains composed of two bones: one close to the trunk, which we call here the anchored bone, and the other one chained to it and attached to the effector joint, called here intermediate bone.

Linear bones are fully constrained once the length, axis direction, and local right directions are determined (two orthogonal vectors are enough to fully constrain the three degrees of rotations from the bone's orientation, and the root point fixes the three remaining degrees of freedom for

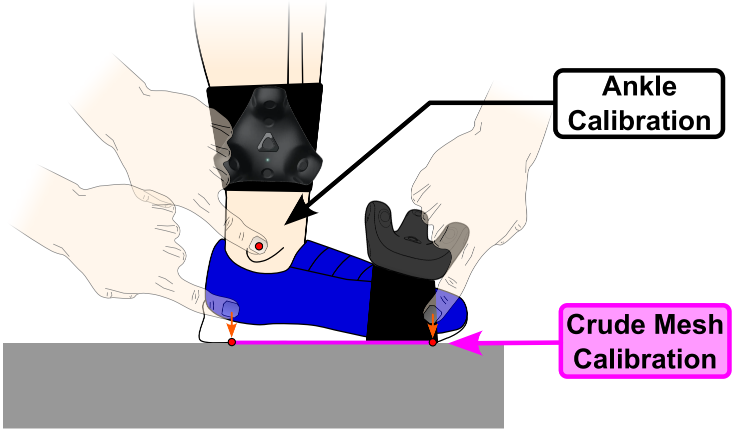


Figure 4.7 – *The foot calibration process expects the user to place their fingertips on the edges of his foot to calibrate the foot's planar surface in contact with the floor, with a projection applied (small oranges arrow) to ensure the measured position is on the floor. The user also places the fingertip on both sides of the malleolus to calibrate the ankle's joint position.*

the bone's placement) as illustrated in Figure 4.8.

Therefore, the intermediate bone is calibrated by determining the intermediate joint position (elbow/knee) with its local right.

Based on the methodology from Molla et al. (2017), the user performs gym motion by flexing arms/legs. At the same time, the relative displacement of the anchored bone tracker is recorded in the intermediate bone's referential (Figure 4.9).

Intermediate joint and bone calibration Knowing the topology of the intermediate joint, the expected shape of the recorded set of point shapes is a circle in a plan. However, knowing the plan is insufficient to determine which normal side should be used as the local right of the limb's kinematic chain.

Therefore, we rely on the knee and the elbow's articular limit, which prevents the joint's angle from exceeding 180 degrees. This means that the average position of the recorded set is necessarily on one side of the half-plan passing by the tracker position and the joint location as illustrated in Figure 4.9.

A first approximation of the joint location used in this computation is performed by fitting a plan from the recorded set of points and then projecting the points on this plan to fit a circle and compute its center.

The plan's fitting is performed by extracting the average positions from the dataset constituting the plan's origin and removing this computed origin from each dataset point. The two main

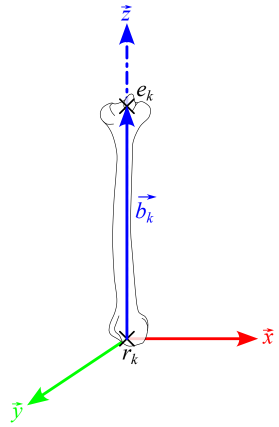
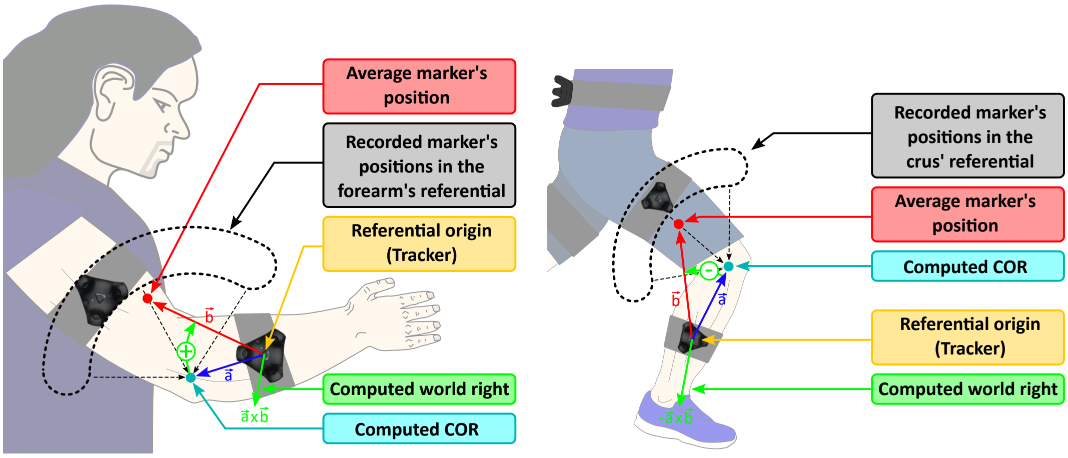


Figure 4.8 – *The linear bone b_k has a structure containing an origin (r_k) and a bone vector $\overrightarrow{b_k}$ that links the proximal joint to the distal joint e_k. Each bone has its referential in which the bone axis is along the local forward direction $\overrightarrow{z}$. The bone consequently also has a local right ($\overrightarrow{x}$) and a local up ($\overrightarrow{y}$). The local right is set as the joint flexion axis (right-sided) for limbs. Constraining the axis direction and its local right (or up) is enough to constrain its world orientation fully. Setting the origin's position fixes the remaining degrees of freedom that fully constrain the bone's placement.*


Average marker's position
Recorded marker's positions in the forearm's referential
Referential origin (Tracker)
$\vec{b}$
$\vec{a}$
$\vec{a}$x$\vec{b}$
Computed world right
Computed COR
Recorded marker's positions in the crus' referential
Average marker's position
Computed COR
$\vec{b}$
$\vec{a}$
Referential origin (Tracker)
Computed world right
-$\vec{a}$x$\vec{b}$

(a) *The tracker's average position is above the half-plan passing by the tracker and the computed COR.*

(b) *The tracker's average position is below the half-plan passing by the tracker and the computed COR.*

Figure 4.9 – *The position of the parent bone's tracker is recorded on a time window of 150 frames (this allows a sufficient average for measurements of the CoR when the user moves the limb). It generates a cloud of points illustrated in black, covering the history of the tracker positions, which is used to retrieve the unique normal corresponding to the joint direction's local right. N.B. Unity uses a left-handed referential. Thus, the output of the cross-product is the opposite of what is expected with a right-handed referential.*

directions of the plan are extracted from the two eigenvectors with the largest eigenvalues of the product of the transposed matrix of the dataset with itself.

Once the plan is determined, a circular regression retrieves the circle center and radius. A second pass is applied to remove the contribution from points whose distance is outside two times the standard deviation in terms of distance toward the center.

However, this method can only retrieve the axis on which the joint is located but not the joint location itself, as this relies on the radius of the limbs that cannot be inferred from this motion as only one tracker is placed on the user's bone, unlike the approach from Molla et al. (2017) where multiple LEDs are placed around the arm and helps to retrieve the location of the joint.

Therefore, to calibrate more precisely the intermediate joint location and the limb radius simultaneously, the user places the fingertip from the other hand on each side of the elbow to calibrate its position as the average position between both points. The intermediate limb radius is computed as the average between the measured radius at the effector's joint (c.f., subsection 4.4.1, subsection 4.4.2) and the current measure of the limb's radius at the joint.

Anchor joint and bone calibration The shoulders and hips are joints that provide more degrees of freedom than the elbows and knees. Thus, rather than having a tracker distribution be a circle, the distribution can now be extended to a sphere that fully constrains the location of the CoR.

Users, therefore, move their arms, paying attention not to lift the arm above the horizontal line and not to mobilize the clavicle. The algorithm records the root's tracker position in the brachium/thigh trackers' referential to locate the joint position.

The CoR is computed as the sphere's center of the recorded dataset using spherical regression. The radius is the average distance between each recorded point and the computed center. The second pass excludes points whose distance to the center is larger than two times the standard deviation, and the same process is reapplied with the filtered input.

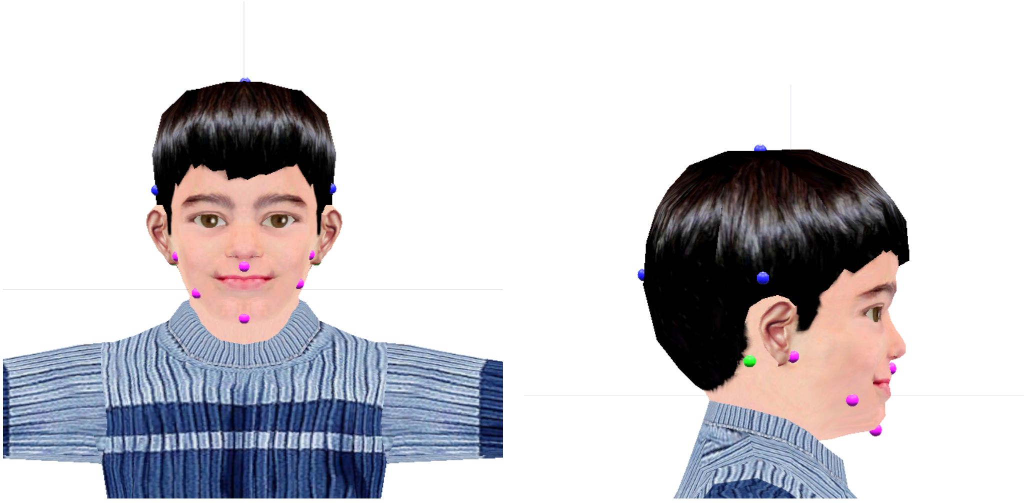
This is followed by measuring two points diametrically opposed at the anchor joint to average the anchored limb's radius.

4.4.4 Head calibration

The jaw's surface is calibrated with six calibration points located at the left and right ear, on the upper lip, at the chin, and on the middle of the left and right side of the jaw, as illustrated in magenta. The calibration is performed by placing fingertips on the illustrated locations. This crude mesh is rigidly attached to the HMD's tracker referential; therefore, it does not consider when users open their mouths.

The crude mesh topology differs slightly from the one from Molla et al. (2017) as the user does not wear an HMD in their approach; hence, their crude mesh can also cover the rest of the face, which is impossible here due to the presence of the HMD.

Also, here, we approximated the back of the head as a sphere rather than using the crude mesh; the sphere was calibrated by placing the fingertips on the skull's top, right, left, and back (blue dots on Figure 4.10). Those calibration points are stored locally in the head's referential: the HMD. The same fitting procedure is used to fit a sphere passing through those calibration points to approximate the back of the head surface. Two additional measurements, illustrated in green, are performed behind the jaw to measure the skull base where the spine is attached to the head.



(a) *Front view* **(b)** *Right view*

Figure 4.10 – *Except for the spine, which might contain a different number of joints compared to the user's skeleton model, the calibrated avatar contains the same structure as the user's skeleton. This means that each surface element has an equivalent in the source user model.*

4.4.5 Trunk calibration

With all limb anchors and the head calibrated, we can perform the trunk calibration. When standing straight up, the sacrum bone width is measured as the distance between both hip joints. According to Langner et al. (2020), the sacrum height is, on average, 11.4 cm for men (standard deviation of 1.1 cm) and 10.9 cm, with a standard deviation of 1.0 cm, for women. Therefore, given the relatively small range of scale of this bone compared to the user's morphology, the root of the spine is statically set to be 10cm above the defined origin of the sacrum (the sacrum bone's model is constructed in a way that both hips are symmetrically placed from the origin) and 5cm backward, and its initial rotation along the hips axis is set so that the spine is vertically aligned as the user stands straight. Then, its position is stored in the back's tracker referential. Additionally, the height of the user's sacrum when standing up is also stored.

Knowing the location of the spine root on the sacrum bone and the skull's base joint, we compute the extended spine distance (used later to compute its flexion) and each vertebra length. Our model uses a spine composed of 12 vertebras and the clavicular bones are calibrated to link the 5th vertebra to each shoulder.

Finally, the user calibrates the torso shape by placing his hand palm on different key points of the torso marked as yellow spheres on Figure 4.12a to measure its body shape. Those crude mesh calibration points are stored in the closest's vertebra referential so that when the user moves, the crude mesh can be deformed accordingly.

4.5 Offline avatar pre-calibration

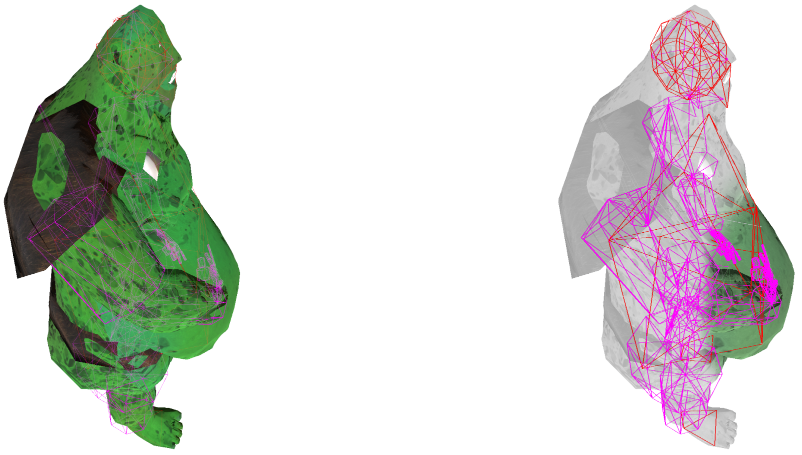
The retargeting pipeline relies on measuring distances toward each surface element and re-applying those scaled distances onto the targetted avatar. Therefore, the avatar structure must comply with the one from the user's skeleton model. Here, the calibration of skeleton bones is direct through the skeleton's rig of the avatar; only the local right directions for knees, elbows, and fingers must be specified to know along which axis joints flex.

The evaluation of the body shape uses the same principle as the user's body surface calibration, except those surface measurements are performed using ray cast hit points on the collider mesh of the avatar, crude mesh calibration points are stored as the position on the avatar's mesh, and the number of vertebrae can differ from the user's skeleton model.

It was observed that the simple mesh representation of a character's belly in Molla et al. (2017), which consists of only seven points on the front and three on the back, is not suitable for accurately representing rounded surfaces, such as an ogre's large belly. This is because there may be interpenetration caused by the gap between the spherical surface and the crude mesh surface that is its chord, as illustrated in Figure 4.11.

To mitigate this issue, we have included four additional points in the center of the crude mesh's belly to reduce the distance between the chord and the surface itself (Figure 4.12a). By default, these points are interpolated from the four corners that are used to define the user's belly, unless the user has a large belly that necessitates more refined calibration. These points remain calibrated manually once for the targeted avatars. Figure 4.12 illustrates a calibrated avatar with the new topology of the crude mesh and the whole set of surface elements.

This process generates a configuration file that can be stored for each avatar; hence this process needs to be applied only once per avatar.



(a) *Illustration of the pose with interpenetrations*

(b) *View of the inner structure used to animate the avatar; the right hand is correctly placed on the surface of the crude mesh but the latter being within the belly, the hand is also within the belly*

Figure 4.11 – *Example demonstrating the problem that can arise when the polygon count in the crude mesh is too low. In this case, the right hand is positioned correctly on the surface of the crude mesh, but since the crude mesh represents is a chord of the belly's rounded shape, the hand interpenetrates with the belly.*

4.6 Online retargeting

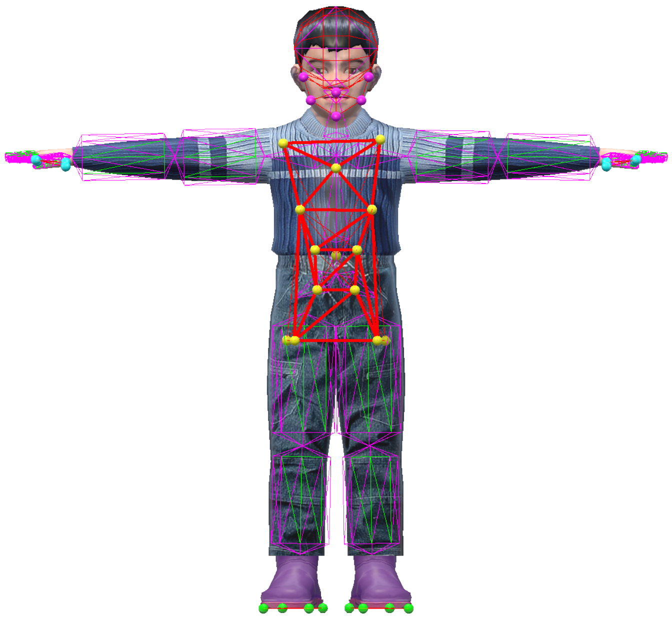
Both the user's dimensions and their current posture are necessary for computing the distances named egocentric coordinates. Therefore this section describes the animation pipeline used to animate the user's skeleton model, followed by the computation of the user's egocentric coordinates that are finally iteratively applied to the avatar's skeleton to animate it.

4.6.1 Skeleton model animation

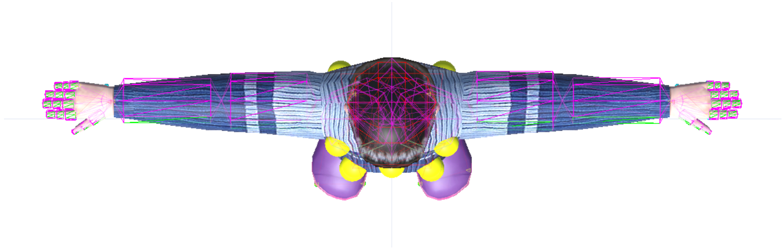
As for the calibration process, our animation pipeline starts from the effectors and moves toward the user's trunk, with the first stage consisting of animating hands and fingers.

Finger motion capture

The inputs from the transformed mocap data combined with the occlusion recovery pipeline from chapter 2 are used to provide a set of ordered points for the animation of the hand's model structure. This is used in both the skeleton reconstruction (to acquire the reference finger poses of the user) and the reconstruction stage of the avatar's hands in the retargeting stage.



(a) *Front view with the new crude mesh topology*



(b) *Top view*

Figure 4.12 – *Capsule bones are calibrated to represent the user's limb shape, and finger digits bones, while the crude mesh represents the shape of the torso, hand and feet, palm surfaces, jaw, and a sphere approximates the back of the skull's surface.*

With the information retrieved from the hand's calibration stage, and assuming that the flexion angle is the same between the intermediate-distal joint and the proximal-intermediate joint Aristidou (2018), we can compute the flexion angle of each finger based on the distance between the fingertip and the finger proximal's joint location that can later be applied on the finger's kinematic chain as illustrated in Figure 4.13. The computation firstly computes and caches the coefficients from Equation 4.1 and then calculates the flexion angle following the steps from Equation 4.1.

$$a = 4 \cdot l_1 \cdot l_3, \qquad b = -2 \cdot (l_1 + l_3), \cdot l_2 \qquad c_1 = l_1^2 + l_2^2 + l_3^2 - 2 \cdot l_1 \cdot l_3 \tag{4.1}$$

$$
\begin{gathered}
c = c_1 - d^2, \qquad \Delta = b^2 - 4 \cdot a \cdot c, \qquad x_1 = \frac{-b - \sqrt{\Delta}}{2 \cdot a} \\
\hat{x}_1 = \begin{cases} -0.9999 & \text{if} \quad x_1 \leq -0.9999 \\ x_1 & \text{if} \quad -0.9999 < x_1 < 0.9999 \\ 0.9999 & \text{if} \quad 0.9999 \leq x_1 \end{cases} \\
\alpha = \pi - \arccos\left(\hat{x}_1\right)
\end{gathered} \tag{4.2}
$$

Before its application, the finger is realigned with the artificial bone linking the wrist to the base joint, the angle α is constrained not to exceed 90°, and then finally applied to the finger. For the thumb, the bone linking the wrist to the base joint is rotated along its axis by 45°.

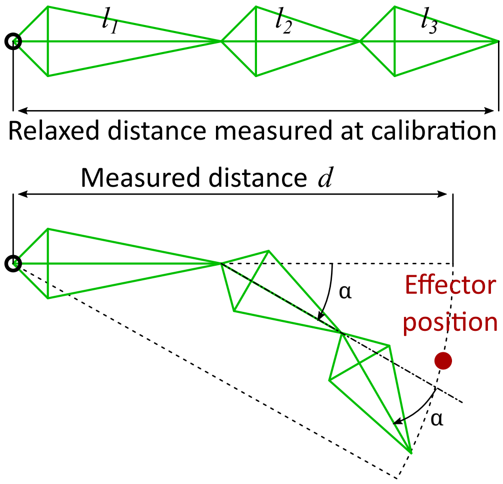


Figure 4.13 – *Flexing finger based on the effector-finger base joint distance. Flexing the finger is insufficient to enforce the effector's position to match the finger's kinematic chain extremity.*

A second pass is then applied to enforce the alignment of fingertips with the expected effectors' positions. The realignment is performed by measuring the pitch and yaw from the expected effector position in the metacarpal bone's referential centered on the proximal's root (Figure 4.14) As flexion induces pure pitch in the finger's tip location in the bone attached to the wrist referential's (e.g., metacarpal for the index), we already know that the current yaw of the flexed finger is zero; therefore, only the pitch of the animated finger is computed before computing the realignment rotation.

The differences in yaw and pitch are then applied to the proximal's root joint and forwarded to the rest of the chain (Figure 4.14). To prevent impossible positions, the measured targeted yaw and pitch from the direction of the expected effector's position is capped using values from Aristidou (2018) for the base joints: The yaw is constrained within $[-15°; 15°]$ and the pitch within $[-85°; +10°]$ for the index, middle, ring, and pinky fingers, whereas the yaw is constrained within $[-30°; 40°]$ and the pitch within $[-15°; +15°]$ for the thumb.

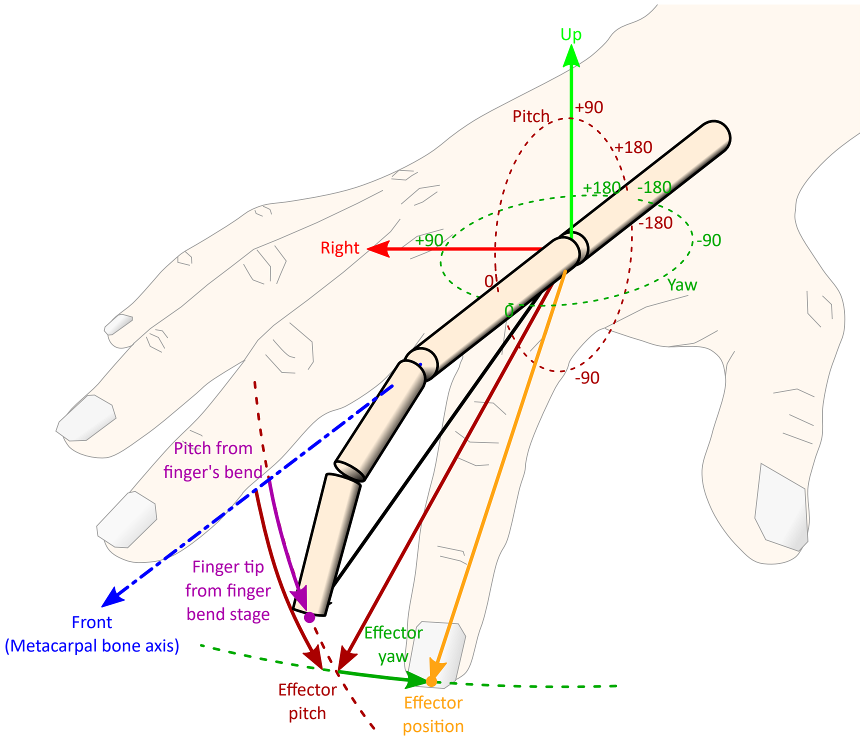


Figure 4.14 – *In the realignment process, the algorithm measures the difference in yaw and pitch between the position of the user's fingertip and the only flexed finger model. Then, the rotation required to align the reference axis with the user's finger is computed as the rotation that rotates the reference axis by the differences in yaw and pitch. This rotation is then applied to all the joints of the finger, and the position of the distal end of each digit segment is computed to update the origin of the next proximal digit side.*

Limb animation

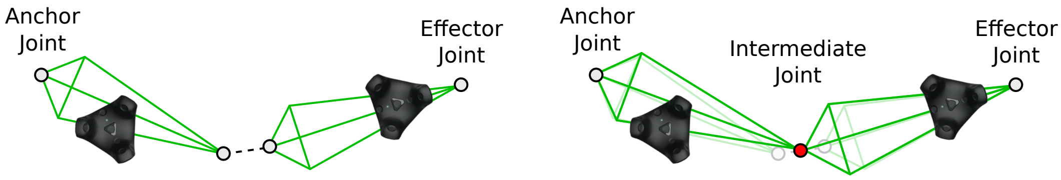
During the calibration process, joint locations were recorded in each tracker's referential; therefore, the computation of joint positions is done by expressing calibrated joint positions in world coordinates. Local directions of the bones' axis and local right are also stored during the calibration process allowing for direct placement in the space of each individual limb bone.

However, trackers' locations are not perfectly rigidly attached to the user's bone, and some offsets may occur, leading to structural gaps. Therefore, anchored and intermediate limb bones (brachium/thigh and forearm/crus) are scaled to ensure a junction of the kinematic chains.

This process is performed by computing the intermediate joint location in both the intermediate and anchored bones' trackers and to average the computed position of the joint. As we gave priority to the effector over the intermediate joints location, the effector position is solely determined using the effector's referential and is not averaged with the intermediate's bone extremity.

The anchored bone is then scaled and oriented to align it with its previously computed anchor position and the newly average intermediate joint position. The intermediate bone is scaled and aligned to make the junction between the intermediate joint position and the effector's joint position.

The realignment is performed while maintaining the local right direction to prevent the twist of bones. This process is illustrated in Figure 4.15.



(a) *Initial limb's structure*

(b) *Adapted limb's structure with continuity enforcement*

Figure 4.15 – *Illustration of the process of linking bones on a limb*

Trunk animation

The last component to animate before animating the trunk is the head. As a simple rigid body attached to the head tracker (i.e., the HMD), the head animation is a simple placement of a rigid body in space. This placement determines the skull base corresponding to the spine's targetted effector position IK.

At this point, all four limb anchors' positions are determined, and the root trackers' positions and orientation are known as the targeted position of the skull base and its local right.

The animation of the trunk is performed in two passes:

- The first places the sacrum as a rigid body attached to the root's tracker. To accommodate the back tracker's potential lateral displacement compared to the sacrum bone, the sacrum's bone lateral rotation is averaged with the lateral direction computed from the hips using the thigh trackers. The same is also applied to the root position of the sacrum. Once the sacrum is placed, the approach from Unzueta et al. (2008) is used to animate the spine flexion as illustrated in Figure 4.16. The spine is then realigned with the targeted effector position, inducing an unrealistically large joint rotation between the sacrum and the first vertebra, according to biomechanics Unzueta et al. (2008).
- Therefore, a second pass is applied by rotating the sacrum along its hip flexion axis to align its up direction with the first vertebra's direction. This changes the root position of the spine, hence the distance between the anchor (sacrum) and the effector (skull base); therefore, a second pass is applied to compute the new flexion and the new realignment of the spine producing the final position from Figure 4.16.

Finally, vertebrae are uniformly twisted along their axis to account for the hips-shoulders and shoulders-head twists.

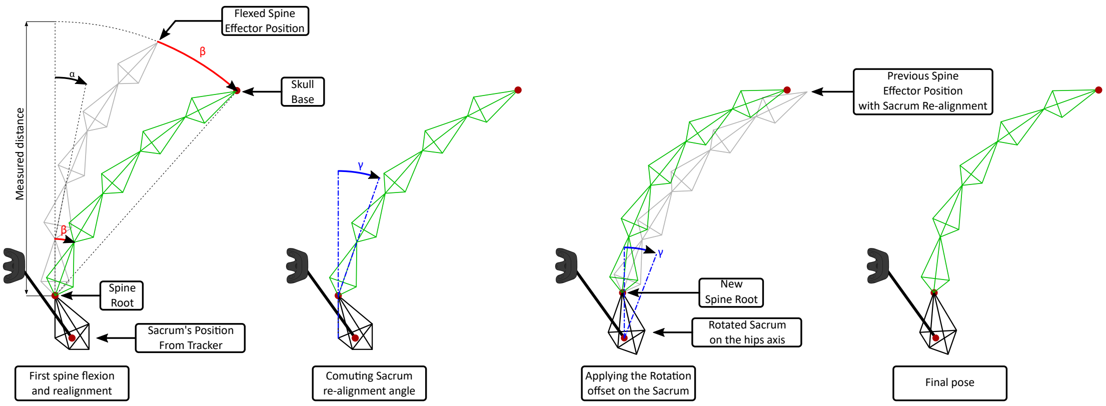

Flexed Spine Effector Position
β
α
Skull Base
Measured distance
β
Spine Root
Sacrum's Position From Tracker
First spine flexion and realignment
γ
Comuting Sacrum re-alignment angle
Previous Spine Effector Position with Sacrum Re-alignment
γ
New Spine Root
Rotated Sacrum on the hips axis
Applying the Rotation offset on the Sacrum
Final pose

Figure 4.16 – *The spine animation process comprises a first resolution of the spine using the IK method from Unzueta et al. (2008), followed by a realignment and a second pass.*

Once the spine state is determined, the clavicle bones are computed to link the 5th vertebra (for our spine model) to the evaluated shoulder anchor positions.

For the later stage of kinematic path normalization, shortcut bones are added, in addition to the bones used to animate the user's skeleton (Figure 4.17), to link: The sacrum root to the left and right hips links the sacrum to the clavicle root and links the clavicle root to the head, similarly to what was proposed by the normalized skeleton representation from Kulpa et al. (2005). Those bones have no constraints on the twist, as their contribution is only used to compute kinematic path normalization, which only considers the bone's axis vector. Those bones can be easily spotted in black down to Figure 4.22.

4.6.2 User egocentric coordinates computation

Here, the value of interest is the distance between each target and each body surface element. Therefore, our approach re-uses the egocentric coordinates from Molla et al. (2017) with the difference that relative rotations between surfaces and effectors are not computed and are replaced by having three targets, forming a rigid body Figure 4.26, used to determine the effector's orientation as described in §Effector position and orientations.

Notations The body's coarse surface structure, including fingers, supporting the computation of egocentric coordinates, consists of 65 surface elements: 26 Triangles (2 per hand and foot, 4 for the face, and 14 for the trunk), 38 Capsules Bones (3 per finger and two per limb), and one sphere for the head. We note the set of surface elements: $(s_i)_{i \in \mathbb{S}}$

The structure also comprises 28 target points (p_j, Figure 4.18) attached to the skeleton's structure:

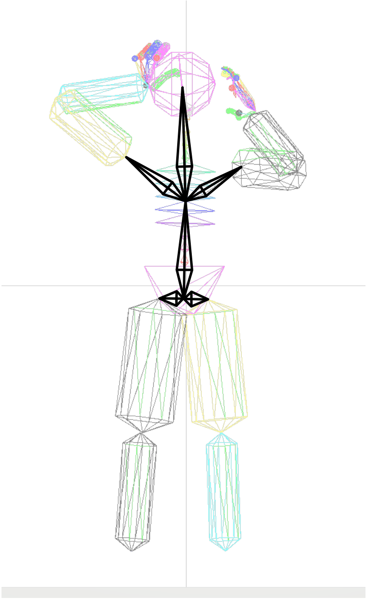


Figure 4.17 – *Illustration, in black, of the shortcut bones used to skip intermediate bones for the normalization computation process.*

three per limb effector (hands and feet), one per intermediate limb joint (elbow, knee), and one per fingertip. Those targets are uniquely indexed in a set we note $\mathbb{T}$. $(p_j)_{j\in\mathbb{T}}$ is the notation of the set of targets.

Unless specified differently, i refers to the surface element index and j to the target point's index.

Coordinates decomposition A self-contact occurs when the distance between two surface elements becomes null. Therefore, the relative distance between a surface element and a target point attached to another surface is one of the most critical components that must be stored.

Rather than using Cartesian coordinates to represent the positions of each point p_j, we represent p_j as a sum of the contribution from each surface element s_i named Egocentric Coordinate. In such a system, the distances and directions (i.e., vectors) between a point p_j and its projection x_i on the surface s_i are noted $\overrightarrow{v_{i,j}}$ and each p_j position is computed as in Equation 4.3, which can be illustrated in Figure 4.18.

$$\forall i \in \mathbb{S}, \forall j \in \mathbb{T}, \quad p_j = x_{i,j} + \overrightarrow{v_{i,j}} \tag{4.3}$$

To account for avatars with different sizes (e.g., bones can be longer), x_i are represented in a normalized form, all noted $\hat{x}_i$ regardless of the surface element, though, each surface elements yield a different representation for the normalized form:

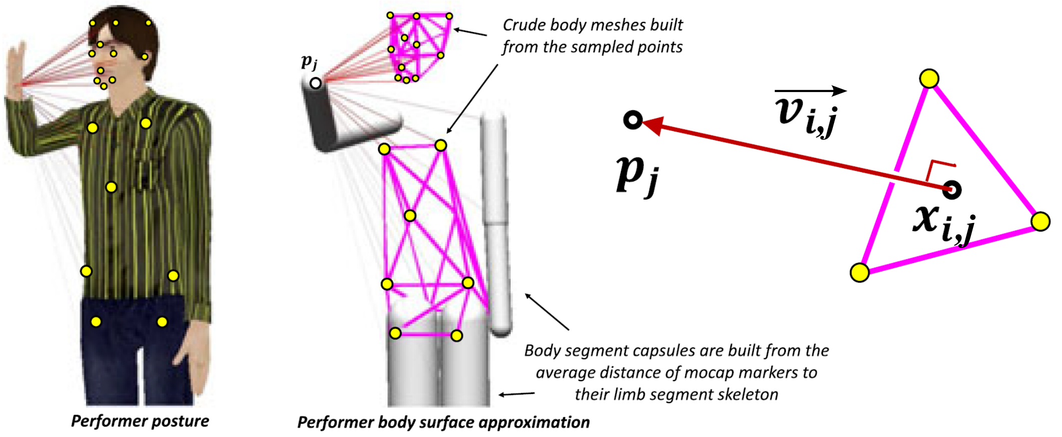

Crude body meshes built
from the sampled points
p_j
$\overrightarrow{v_{i,j}}$
p_j
$x_{i,j}$
Body segment capsules are built from the
average distance of mocap markers to
their limb segment skeleton
Performer posture
Performer body surface approximation

Figure 4.18 – *Illustration, adapted from Molla et al. (2017), representing the decomposition of a point's position into a surface contact point and a vector.*

- x_i on triangles are stored as barycentric coordinates
- x_i on cylinders are stored as cylindrical coordinates
- x_i on the sphere is stored as a spherical coordinate

Furthermore, the contribution of bones on the contributing vector $\overrightarrow{v_{i,j}}$ are also considered with the length of the bones that contribute to the kinematic chain, as detailed later in subsubsection 4.6.2. This overall contribution of the kinematic chain on $\overrightarrow{v_{i,j}}$ is noted $\tau_{i,j}$.

Finally, to prioritize self-contacts over global positioning targets in space, each contributing vector $\overrightarrow{v_{i,j}}$ is weighted according to its relevance to a self-contact. For instance, if the contribution indicates that the target is close to surface elements, its weight would be high, while conversely, when the contributing vector does not either help reconstruct the pose or when its information is less relevant than another contributing vector, its weight would be lighter. We note $\lambda_{i,j}$ such a weight for the importance of the vector $\overrightarrow{v_{i,j}}$ in the reprojection process detailed in §Target point reprojection (The actual implementation embeds different sets of weights, but this relates more of the low-level implementation than the high-level logic).

Therefore, in the end, for each target point p_j, the contributions comprise the following elements that are illustrated in Figure 4.19:

- The set of normalized surface projection points $(\hat{x}_{i,j})_{i \in \mathbb{S}}$
- The set of normalized vector contributions from the surface projection $(\overrightarrow{v_{i,j}})_{i \in \mathbb{S}}$.
- The set of normalization factor $(\tau_{i,j})_{i \in \mathbb{S}}$ that represents the effective displacement induced by each bone of the kinematic chain linking p_j to $x_{i,j}$

- The set of the importance of the contribution of $\overrightarrow{v_{i,j}}$ denoted $(\lambda_{i,j})_{i\in\mathbb{S}}$

With such a system of coordinates, the opposite operation to compute the retro-projected target point p'_j is done through the analog equation Equation 4.4 with $x'_{i,j}$ the transposition of $\hat{x}_{i,j}$ on the avatar's surface element s'_i, and $\tau'_{i,j}$ the normalization factor computed on the avatars' kinematic chain.

$$p'_j = \sum_{i\in\mathbb{S}} \left(x'_{i,j} + \overrightarrow{v_{i,j}} \cdot \frac{\tau'_{i,j}}{\tau_{i,j}} \right) \cdot \lambda_{i,j} \tag{4.4}$$

Computing target's projections and contributing vectors

For each frame, we must compute all p_j egocentric coordinates. The computation thus involves first projecting p_j on each element surface as illustrated in Figure 4.19 to firstly determine a set of $(x_{i,j})_{i\in\mathbb{S}}$ and $(\overrightarrow{v_{i,j}})_{i\in\mathbb{S}}$.

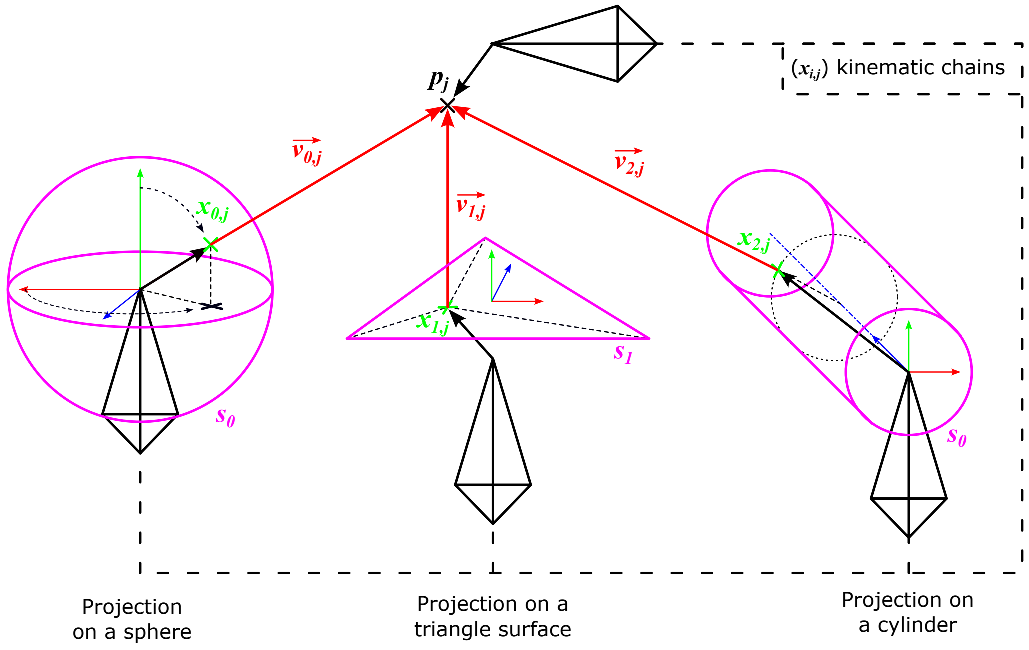

p_j
$(x_{i,j})$ kinematic chains
$\overrightarrow{v_{0,j}}$
$\overrightarrow{v_{1,j}}$
$\overrightarrow{v_{2,j}}$
$x_{0,j}$
$x_{1,j}$
$x_{2,j}$
s_0
s_1
s_0
Projection on a sphere
Projection on a triangle surface
Projection on a cylinder

Figure 4.19 – *In the egocentric coordinate system, each point position p_j is decomposed into a sum of contributions from each element surface s_i. Those surface elements can either be a sphere (on the left), a mesh triangle (middle), or a cylinder (on the right). The contribution for each surface element to the jth target point is denoted $\overrightarrow{v_{i,j}}$ (in red) and represents the vector between p_j and the closest projection point of p_j on s_i that is noted $x_{i,j}$ (in green). Finally, a normalization factor $\tau_{i,j}$ is computed as the sum of the dot product of each bone's length and $\overrightarrow{v_{i,j}}$.*

When s_i is a sphere, the projection is directly performed by measuring p_j in spherical coordinates in the referential of s_i and setting its radius to the sphere's radius to measure $x_{i,j}$, while the same spherical coordinates already represent $\hat{x}_{i,j}$ the normalized coordinate of $x_{i,j}$ on the sphere (the radius coordinate is not relevant as it is replaced with the avatar's sphere radius later on the projection stage described in the next section).

When s_i is a cylinder, we measure p_j's cylindrical coordinates in s_i's referential and set its radius to the cylinder's radius. The axial height (component along the blue axis on Figure 4.19) is capped to maintain the point on the cylinder surface. $x_{i,j}$ is computed as the retro projection of this capped cylindrical coordinate in the cylinder's surface. Its normalized representation $\hat{x}_{i,j}$ is the cylindrical coordinate whose height is divided by the length of the cylinder so that it is always between 0 and 1. As for the spherical coordinate, the radius is irrelevant, as it is later overridden by the avatar's corresponding radius.

Finally, when s_i is a crude mesh's triangle, the point p_j is projected on the surface. If the projection falls outside the surface, it is computed as the closest point on the edge of the triangle. The normalized coordinates of $\hat{x}_{i,j}$ are then computed as the barycentric coordinate of $x_{i,j}$ in the triangle s_i.

In addition to the user's surface, the distance toward the floor is also considered. Here, only the contribution from the vertical component h_j is retrieved. Hence the projection consists in taking only the height of p_j.

With the set of $(x_{i,j})_{i\in\mathbb{S}}$ computed, $(\overrightarrow{v_{i,j}})_{i\in\mathbb{S}}$ is easily computed as the difference $\overrightarrow{v_{i,j}} = p_j - x_{i,j}$. Figure 4.20 illustrates this process of computing the contribution vectors from each surface element of the user.

Computing contribution weights

For each contribution vector $\overrightarrow{v_{i,j}}$, we defined a contribution weight ($\lambda_{i,j}$) such that the retro-projection of the point is performed as Equation 4.4. To compute this relative contribution of each vector, we first compute a set of raw weights $\Lambda_{i,j}$ that only relies on the surface element and the target point to be computed. Then, a normalization process described below yields the $\lambda_{i,j}$.

As detailed in subsection 4.6.3, the animation pipeline is performed through two separate convergence loops, at first the limb level, followed by the finger level (as the finger position requires the hand's position to be computed first). To prevent artifacts in the attraction of the limbs due to the finger's contribution, two sets of weights are computed; one set with the contribution of the fingers' surface elements set to zero (i.e., as if the finger were not integrated), and one set with the full contribution of all surface elements to address the fingers' self-contacts.

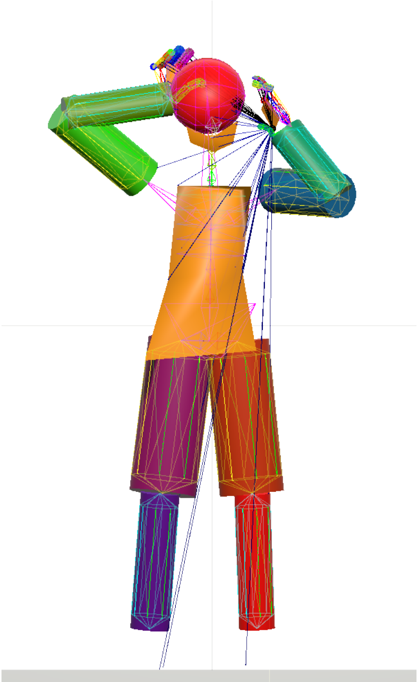


Figure 4.20 – *Illustration of the set of vector contributions* $(\overrightarrow{v_{i,j}})_{i\in\mathbb{S}}$ *for a single target point* p_j. *N.B. The thumb wireframe layer overlaps the extremities of the contribution vectors.*

Removing trivial contributions Prior to computing the weight of the contributions, we already know that the hand, for instance, is always located close to the forearm's extremity. Therefore the importance of the associated contribution vector should be small, not to erase the contribution from other vectors in the weights normalization process. Consequently, the weights from the same limb on which a target point p_j is attached are always set to zero, and their associated computations are skipped in the rest of the pipeline.

Raw contributions weights computation The raw $\Lambda_{i,j}$ weights are computed as:

- $\frac{1}{\|\overrightarrow{v_{i,j}}\|^2}$ for spherical coordinates
- $\frac{1}{\|\overrightarrow{v_{i,j}}\|^2} \cdot \left|\sin\left(\angle(\overrightarrow{v_{i,j}}, \overrightarrow{b_i})\right)\right|$ for cylindrical coordinates with $\overrightarrow{b_i}$ the axis of s_i cylindrical referential.
- $\frac{1}{\|\overrightarrow{v_{i,j}}\|^2} \cdot \cos\left(\angle(\overrightarrow{v_{i,j}}, \overrightarrow{n_i})\right)$ for crude mesh elements with $\overrightarrow{n_i}$ the normal of s_i's triangle
- $\frac{1}{h_j^2}$ for the floor's height contribution.

Weights contributions normalization Once the raw weights are computed, the process described in this paragraph is executed twice: once to generate the sets of weights used for the limb convergence loop and once for the fingers one, to ensure the stability of the limb convergence first and then the animation of the fingers. The difference between the two executions is that, for limbs' weight computation, the weights associated with fingers' surface elements are set to zero and thus skipped.

In both cases, the raw weights $\Lambda_{i,j}$ are normalized into normalized weights ($\lambda_{i,j}$, with $\sum_{i \in \mathbb{S}} \lambda_{i,j} = 1$) such that the reprojection yields the target point p'_j.

However, the floor contribution presents a singularity: It only contributes to the vertical height of p'_j but not the lateral location on the ground plan. Hence, when a target point is close to the ground, the weight contribution of the floor $\lambda_{\text{floor\_id},j}$ would tend to 1, erasing all the other contributions, which are the only ones contributing to the planar lateral position. Ultimately, it would result in a retro-projected point to the origin of the space rather than just only on the floor, not to mention the precision issues when dealing with vectors with tiny amplitude to determine a direction.

Consequently, two sets of weights are actually computed (for both executions): $(\lambda_{i,j})_{i \in \mathbb{S}}$ and $\left(\lambda_{g_{i,j}}\right)_{i \in \mathbb{S}}$. The last one ($\left(\lambda_{g_{i,j}}\right)_{i \in \mathbb{S}}$) is computed by normalizing the set of all raw contributions (i.e., including the one from the ground) while the former ($(\lambda_{i,j})_{i \in \mathbb{S}}$) skips the contribution of the ground in its normalization process.

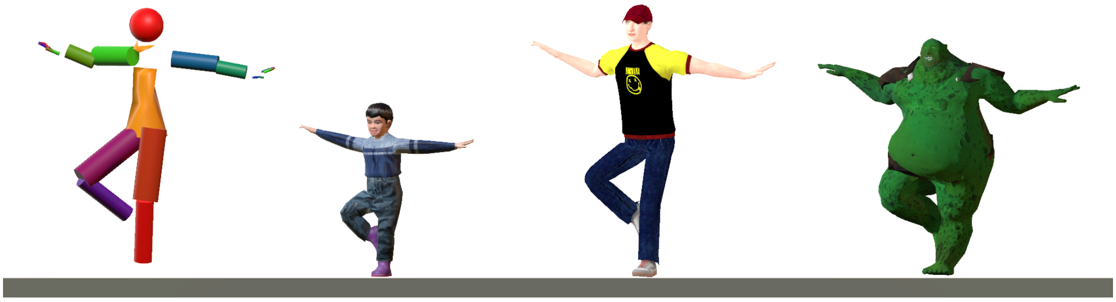
The retro-projection formula of p'_j Equation 4.4 is therefore adapted into Equation 4.5 to address this singularity (with $\overrightarrow{e_1}$, $\overrightarrow{e_2}$ and $\overrightarrow{e_3}$ the x (right), y (up), and z (front) world axis respectively).

$$\begin{cases} p'_j \cdot \overrightarrow{e_1} = \left(\sum_{i \in \mathbb{S}} \left(x'_{i,j} + \overrightarrow{v_{i,j}} \cdot \frac{\tau'_{i,j}}{\tau_{i,j}} \right) \cdot \lambda_{i,j} \right) \cdot \overrightarrow{e_1} \\ p'_j \cdot \overrightarrow{e_2} = \left(\sum_{i \in \mathbb{S} \setminus \{\text{floor\_id}\}} \left(x'_{i,j} + \overrightarrow{v_{i,j}} \cdot \frac{\tau'_{i,j}}{\tau_{i,j}} \right) \cdot \lambda_{g_{i,j}} + h_j \cdot \lambda_{g_{\text{floor\_id},j}} \right) \cdot \overrightarrow{e_1} \\ p'_j \cdot \overrightarrow{e_3} = \left(\sum_{i \in \mathbb{S}} \left(x'_{i,j} + \overrightarrow{v_{i,j}} \cdot \frac{\tau'_{i,j}}{\tau_{i,j}} \right) \cdot \lambda_{i,j} \right) \cdot \overrightarrow{e_3} \end{cases} \tag{4.5}$$

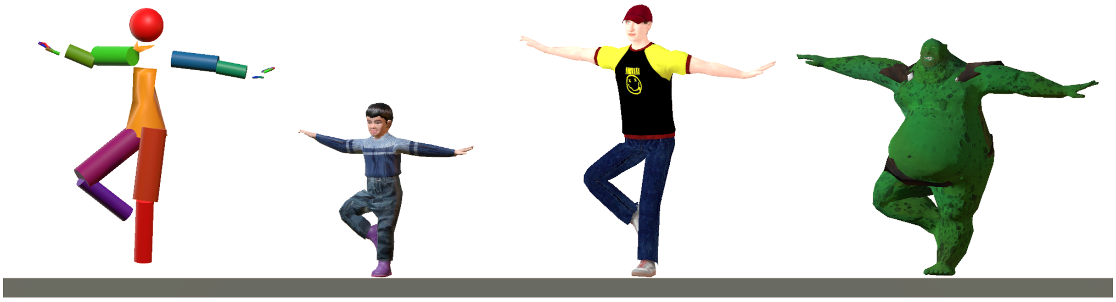
Kinematic path normalization

In their approach Molla et al. (2017), the authors stressed the importance of normalizing the contribution vectors based on the kinematic chain to avoid introducing deviations in limb positions. This can be illustrated in the mismatch of arm pose from Figure 4.21a.

Therefore, each vector contribution $\overrightarrow{v_{i,j}}$ must be scaled before the avatar adaptation loop procedure. Here, the idea is to measure each source bone's contribution alongside the kinematic chain to



(a) *Retargeting animation without kinematic chain normalization*



(b) *Retargeting animation with kinematic chain normalization*

Figure 4.21 – *Illustration of the same pose on different avatars with and without the normalization enabled. The structure on the left is the modelized user's skeleton. On top, the non-normalized animation produces flexions in the characters with long arms (the two avatars on the right), while for the child, the arm is completely extended, although this was not the case on the source skeleton. On the bottom, the normalization of effector positions straightens the arms of the two characters on the right and reduces the extension of the child's arms.*

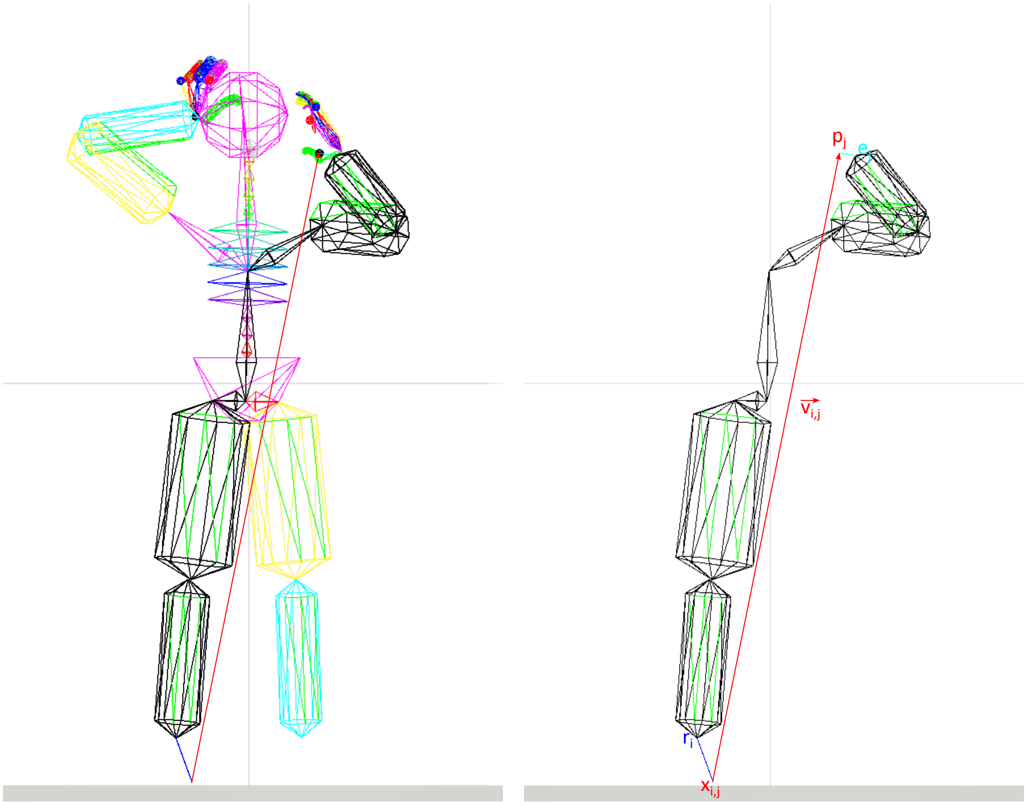
then scale up (or down) the contribution of the avatar's bone.

The scaling factor for the source kinematic chain linking the target point p_j to its projection $x_{i,j}$ on the surface element s_i is noted $\tau_{i,j}$. Those chains are pre-computed and stored so that the tree search is not performed in every frame.

Let $\left(\overrightarrow{b_k}\right)_{k \in [\![0,n]\!]}$ be the list of n bones axis vectors that comprise this kinematic chain, with r_i the root of the surface element s_i and e_j the root of the bone to which the target point is attached. We have Equation 4.6

$$\tau_{i,j} = \sum_{k=0}^{n} \overrightarrow{\hat{v}_{i,j}} \cdot \overrightarrow{b_k} + \overrightarrow{\hat{v}_{i,j}} \cdot \overrightarrow{(x_{i,j} - r_i)} + \overrightarrow{\hat{v}_{i,j}} \cdot \overrightarrow{(p_j - e_j)} \tag{4.6}$$

This is illustrated in Figure 4.22 where the contribution vector $\overrightarrow{v_{i,j}}$ is displayed in red, the kinematic chain in black, and the extremity segments in blue and cyan. The kinematic path computation uses the trunk's simplified kinematic chain to accelerate the process.



(a) *Kinematic chain with the whole skeleton structure (without the crude mesh)*

(b) *Kinematic chain only with the surface reference points and extremity segments detailed*

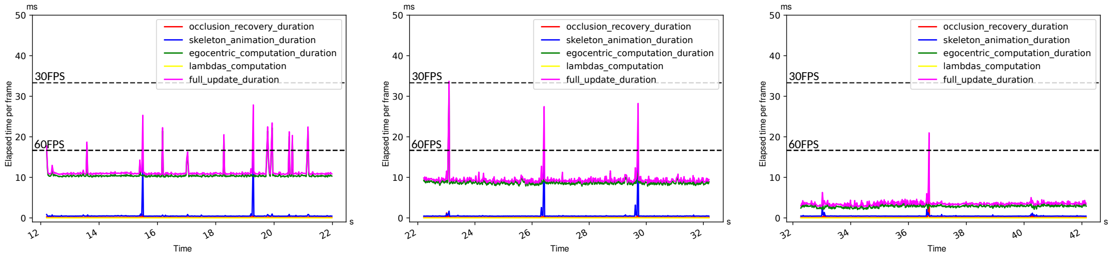
Figure 4.22 – *Illustration of the kinematic chain used to compute the normalization factor* τ. *The spine model was simplified to reduce the computational cost.*

When the associated lambda for a kinematic chain is null, the computation of the kinematic chain is skipped, as the associated vector will not contribute to the reprojection stages. An example of a normalization process enhancing the final result is illustrated in Figure 4.21.

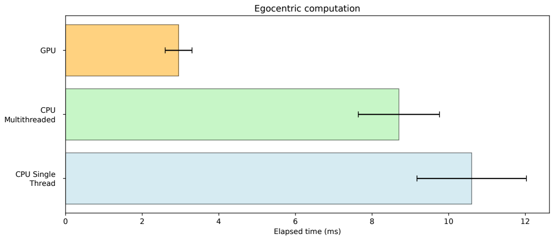
Performance optimization

The heavy computations described in this section must be iterated in each frame and for each target point p_j (Figure 4.20). Therefore, its computational cost is not to be neglected and sums up into the entire animation pipeline computation time. With the real-time constraints required to allow the embodiment of an avatar, our observations showed us that this process struggled to be computed sequentially in C# with Unity Figure 4.23a.

Therefore, before any egocentric coordinate computation is performed, all the pertinent information from the structure is computed and placed in cache (e.g., vectors directions, points, bones rotations, or world up, right, and forward directions).



(a) *CPU Single threaded* **(b)** *CPU Multi threaded* **(c)** *GPU*



(d) *Update time*

Figure 4.23 – *Performance measurements between parallelized (CPU and GPU) and sequential pipeline processing for egocentric computation. We can observe that the largest time consumption is drawn by the computation of $x_{i,j}$, $\hat{x}_{i,j}$, $v_{i,j}$ and $\Lambda_{i,j}$. Consequently, we implemented several pipelines using CPU and GPU parallelization to reduce the latency. Values measured on a desktop equipped with an Intel Core i7-4790 CPU (4 Cores at 3.6GHz, boosting at 4GHz) with 8GB of DDR3 RAM and an NVIDIA GeForce GTX 970 GPU with 4GB of DDR5 SDRAM*

Here, the computation of the projection and normalized surface elements represents the heaviest part of the egocentric computation. To address this issue, most of the memory was set to be allocated once and updated rather than conveniently instantiating and freeing objects in each frame. Mutexes were added to prevent concurrent access to the same memory blocs when running in CPU multithreaded mode (Figure 4.23c).

However, those improvements did not produce a sufficient significant difference to be used in the animation and a final implementation was made using GPU acceleration through Compute Shaders written in HLSL for DirectX12. Those parallelized pipelines are compared to the sequential pipeline in Figure 4.23 with measured performance improvement by roughly 200% for the GPU-based version compared to the CPU-based one.

4.6.3 Avatar posture adaptation loop

The computation of the avatar's pose from the set of egocentric coordinates is composed of several stages illustrated in Figure 4.24. The first one places and directly applies the raw angles captured from the source user's skeleton model onto the avatar's structure (further noted "direct kinematics" in the evaluation section 5).

Table 4.1 – *Performance comparison between parallelized and sequential pipeline processing for egocentric computation*

Measure		Mean Elapsed Time (ms)	Standard Deviation (ms)
Occlusion Recovery Pipeline for both Hands		0.498	0.640
Skeleton Animation		1.089	0.752
Egocentric Projections	Single-Threaded	10.6	1.428
	Multi-Threaded	8.7	1.067
	GPU	2.95	0.353

Then, an iteration loop (with a fixed number of iterations, here set to three) is used to animate the body at the limb level to place and orient the limb effectors (i.e., wrists and ankles). Finally, a second iteration loop, also fixed in terms of iterations with a value of two, handles the finger animations to produce the final avatar's pose.

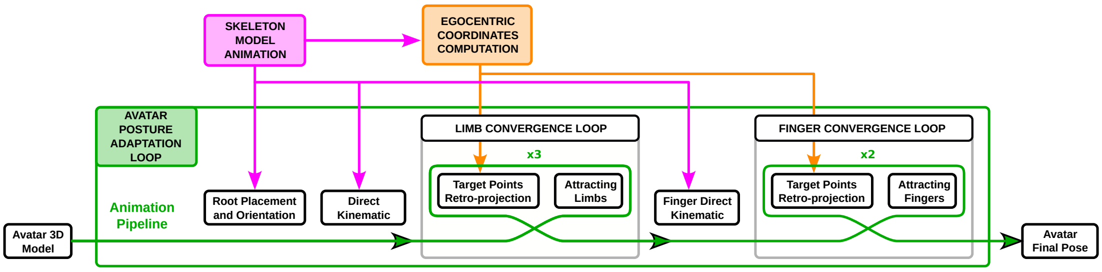


Figure 4.24 – *The posture adaptation pipeline is applied for each frame. It resets the placement of the avatar and pre-orient limbs using direct kinematic forward. Limbs are then progressively attracted toward their retro-projected target points (p'_j). Fingers are then initialized using direct kinematics and are also progressively attracted toward their retro-projected target points to produce the final avatar pose.*

Avatar pose initialization

In this process, the root of the avatar's skeleton is placed at the scaled height of the user's root. With h_{c_u} the height of the user's sacrum calibrated when standing up, h_{a_c} the height of the avatar's sacrum when standing up, the location of the avatar's sacrum is initialized at the height h_a as defined based on the current height of the user's skeleton h_u in Equation 4.7.

$$h_a = \frac{h_{a_c}}{h_{u_c}} \cdot h_c \tag{4.7}$$

Once the height is adjusted, the avatar's sacrum is aligned with the user's sacrum, followed by the animation of the spine to account for the twist of the source skeleton. Such a design choice might lead to the footskate phenomenon; however, this is out of the scope of this research, and addressing this issue is discussed below in the limitation section.

Once the trunk is placed and animated, the limbs are animated by orienting the anchor bones first to match the orientation of the source skeleton's structure. This process then continues toward the limbs' extremities with the orientation of intermediates bones and, finally, effectors' orientation. The head is also oriented to match the original head orientation.

This constitutes the baseline for the progressive attraction of effectors toward their expected reprojection location.

Limb animation convergence loop

The progressive attraction of effectors' positions towards their final position is performed by iteratively updating the surface elements of the avatar and computing the retro-projection of the position of the target points from the current pose (§Target point reprojection), computing the new effectors' position and their orientation (§Effector position and orientations) before finally progressively attracting the current effector position toward the new effector position and animate the limbs to reach the effector position (§Limb inverse kinematic).

Target point reprojection Based on the current avatar pose, the calibrated surface elements are updated to match the avatar's skeleton structure. This corresponds to the source location and orientation for the capsule bone and the sphere from the current avatar's skeleton. For the crude mesh, this corresponds to updating its vertices from the location of the avatar's mesh corresponding vertices (i.e., the avatar's crude mesh is attached to the avatar's skin and not to an internal model of the spine, although the avatar's skin is rigged to the spine).

Once the surface elements are placed, the formula from Equation 4.5 is applied to determine the set of reprojected target points $\left(p'_j\right)_{j\in\mathbb{T}}$ onto the avatar as illustrated for a single p'_j in Figure 4.25.

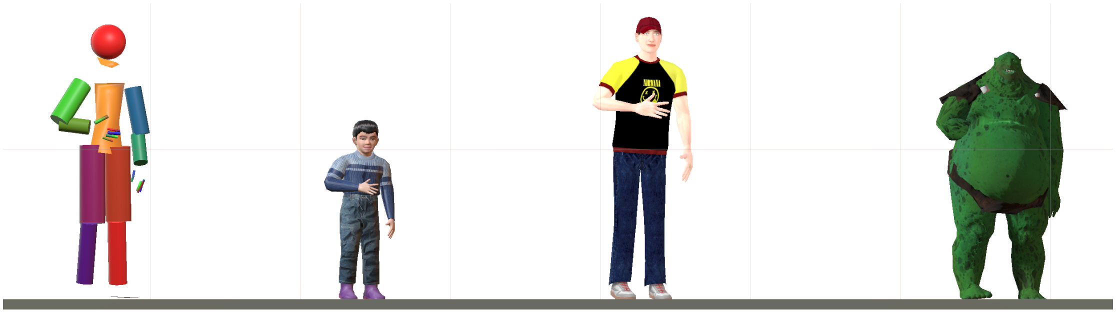
However, to compute p'_j, Equation 4.5 expects all $\tau'_{i,j}$ to be known. In the Equation 4.6 the last term to compute $\tau_{i,j}$ is $\overrightarrow{\hat{v}_{i,j}} \cdot \overrightarrow{(p_j - e_j)}$. However, p'_j is unknown at the first pass during the retro-projection stage; hence, the computation cannot be performed.

Therefore the actual computation of all $\tau_{i,j}$ is performed in parallel with the computations of a set of $\left(\tau_{el_{i,j}}\right)_{(i,j)\in\mathbb{S}\times\mathbb{T}}$ using the formula from Equation 4.6 but without the final term, i.e., the effectors' kinematic chain normalization, computed from Equation 4.8.

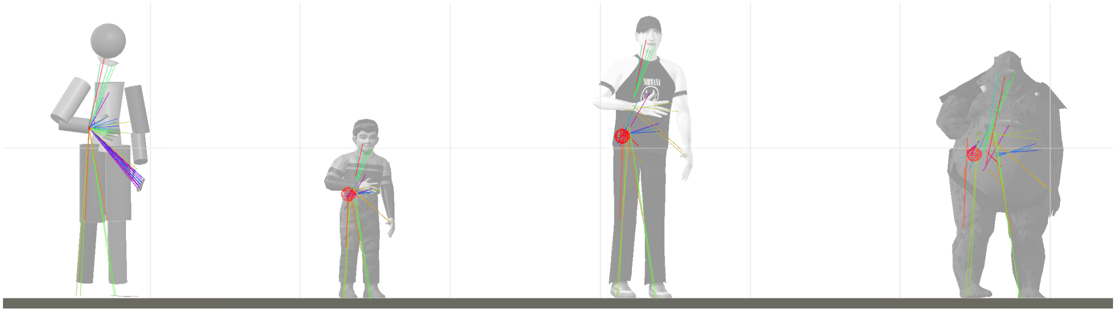
$$\tau_{el_{i,j}} = \sum_{k=0}^{n} \overrightarrow{\hat{v}_{i,j}} \cdot \overrightarrow{b_k} + \overrightarrow{\hat{v}_{i,j}} \cdot \overrightarrow{(x_{i,j} - r_i)} \tag{4.8}$$

To be noted, individual dot products are also stored not to recompute those intermediates several times.

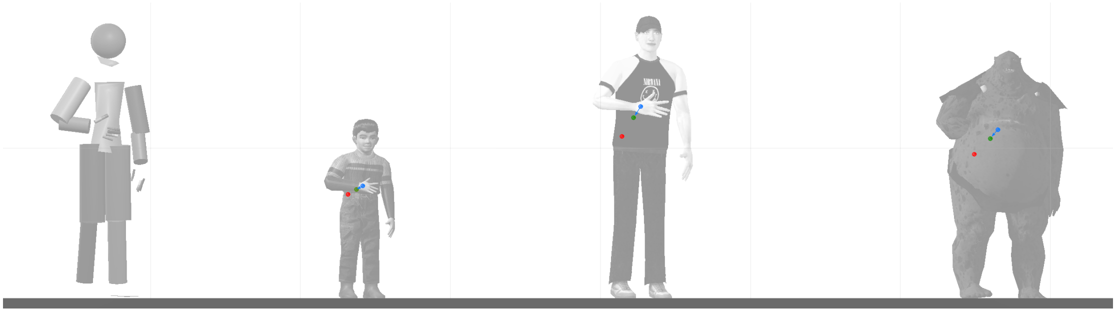
This alternate computation is only used on the first pass. After that, the p'_j from the previous frame is known and fed to the original equation with the effector's term Equation 4.6.



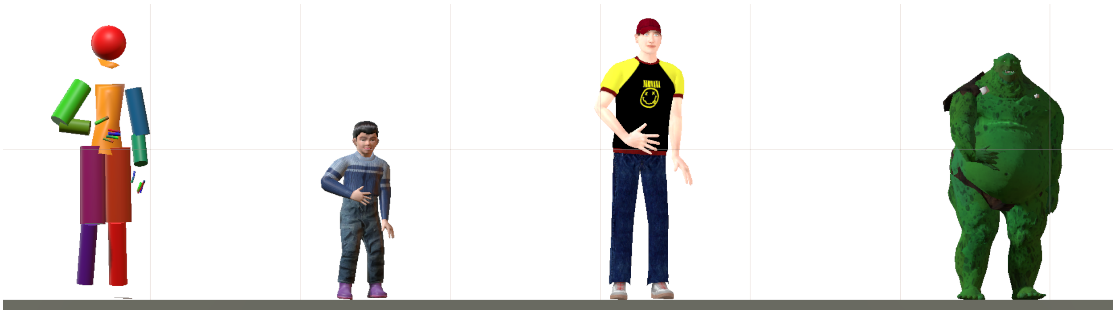
(a) *Raw angle application from the source skeleton onto the avatar*



(b) *Contribution vectors applied onto the avatar surface elements with the computed target point* p'_j *illustrated with the red sphere.*



(c) *Attraction process illustrated for the first of the three limb passes. In red, the retro-projected target point* p'_j*, in blue, the source position* a_j *before the attraction, and in blue, the same target point* a_j *after the attraction.*



(d) *Attraction process after the three passes and two passes for limb and limbs retargeting convergence loops.*

Figure 4.25 – *Illustration of the computation of the contribution vectors and their normalized re-application on the destination avatar to compute the avatar's position.*

Once the complete set of target points $\left(p'_j\right)_{j\in\mathbb{T}}$ is computed, each current target point $(a_j)_{j\in\mathbb{T}}$ is progressively attracted towards its reprojection, in proportion to the pass number over the total number of passes (set to three), following Equation 4.9 with w the ratio factor $\frac{\text{current iteration}}{\text{total iterations count}}$.

$$a_j \leftarrow (1-w)\cdot a_j + w\cdot p'_j \tag{4.9}$$

Effector position and orientations Once the complete set of target points for the avatar $(a_j)_{j\in\mathbb{T}}$ is computed, the effectors' positions (i.e., wrist and ankles) are then extracted from the rigid body composed of the three reprojected target points from the hand's palms or the feet illustrated in Figure 4.26.

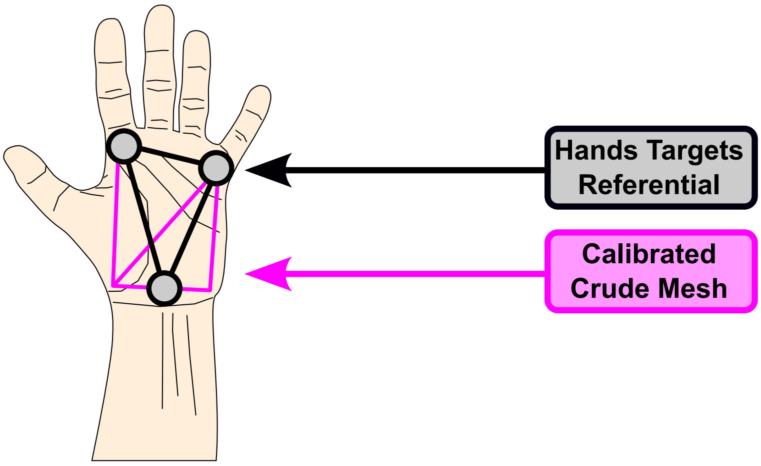


Figure 4.26 – *Three target points are assigned to each effector used as a rigid body referential that allows the expression of the wrist location in this referential and to measure a rotation.*

Unlike Molla et al. (2017), when the effector position is far from any surface, the effector's orientation is directly sourced from the raw user's skeleton model to help at respecting the semantic meaning of the pose. When the effector gets closer to a surface element, using the raw orientation may induce interpenetrations if a close surface is present nearby with, for instance, a different orientation than the source surface.

For this reason, the avatar effector orientation is computed as the average between the user effector orientation and the one defined by the three retro-projected target points. The ratio selection uses the maximum weight of the lambdas computed in section 4.6.2, and the rotation average is performed using the method from Markley et al. (2007), with the selection ratio varying from 0 (the effector orientation is the user's skeleton effector orientation) to 1 (the orientation is entirely defined by the three retro-projected attracted target points). The value chosen for this selection ratio is $\max\left((\lambda_{i,j})_{(i,j)\in\mathbb{S}\times\{a,b,c\}}\right)$, with $(a,b,c)\in\mathbb{T}^3$ the indexes of the three target points attached to the effector, as $\lambda_{i,j}$ represents the importance of the link, and subsequently, relates also the importance of the contribution vector to the orientation.

Limb inverse kinematic To finish the limb's animation, an analytical IK is used with the newly computed location of the retargeted effector and intermediate joint position. This is applied to produce the animation of the four avatar's limbs.

The IK works as follow: Knowing each bone's length (l_1 and l_2), this information is combined with the measurement of the distance effector base joint (d) to retrieve the intermediate flexion angle α (Figure 4.27a) using the formulas Equation 4.10 and Equation 4.11.

$$\cos_\alpha = l_1^2 + l_2^2 - \frac{d^2}{2 \cdot l_1 \cdot l_2} \tag{4.10}$$

$$\alpha = \begin{cases} \pi & \text{if} \quad \cos_\alpha \geq 1 \\ 0 & \text{if} \quad \cos_\alpha \leq -1 \\ \pi - \arccos(\cos_\alpha) & \text{otherwise} \end{cases} \tag{4.11}$$

Once the flexion angle is computed, it is applied to the intermediate joint as a rotation along its right (conversely left for the legs) axis to produce the flexion. The second stage then aligns the produced effector position with the expected one.

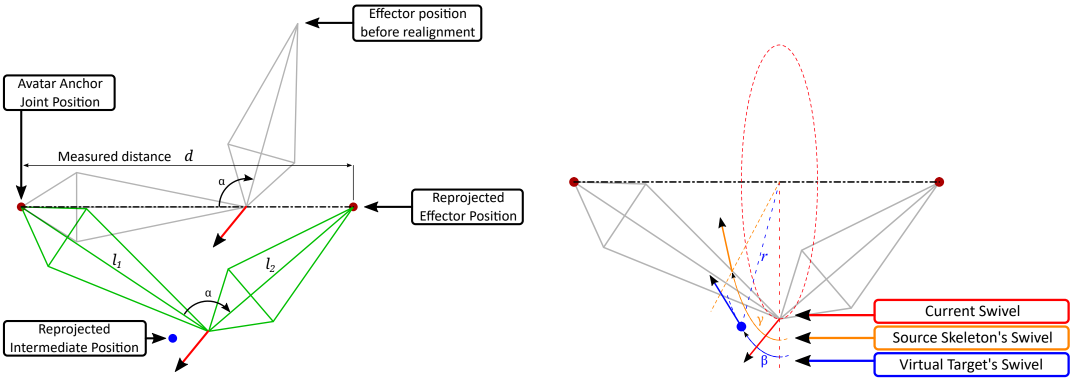
At that stage, the swivel angle along the root-effector axis remains to be determined (Figure 4.27b); logically, one expects to infer it from the location of the reprojected intermediate joint target point. However, this approach becomes unstable the closer the retro-projected intermediate joint is to the root-effector axis, as three aligned points cannot constrain a plan.

Therefore, the swivel angle is adjusted using an interpolation between the source skeleton model limb swivel angle (used for unstable cases) and the angle to align the limb intermediate joint in the half-plane determined by the reprojected intermediate joint position (used when its distance to the swivel axis is sufficient to avoid instabilities). The alignment swivel angle δ is computed using Equation 4.12 and Equation 4.13 with r the radial distance of the reprojected intermediate joint's target point onto the root-effector axis (Figure 4.27b).

$$t = \begin{cases} 0 & \text{if} \quad r \leq 0.05 \\ (r - 0.05) \cdot 10 & \text{if} \quad 0.05 < r < 0.15 \\ 1 & \text{if} \quad 0.15 \leq r \end{cases} \tag{4.12}$$

$$\delta = t \cdot \beta + (1 - t) \cdot \gamma \tag{4.13}$$

Finally, the angle δ is applied through a rotation along the effector axis to end the process.


Effector position before realignment
Avatar Anchor Joint Position
Measured distance d
α
Reprojected Effector Position
l_1
l_2
α
Reprojected Intermediate Position
r
γ
β
Current Swivel
Source Skeleton's Swivel
Virtual Target's Swivel

(a) *Limb flexion and realignment computation* **(b)** *Swivel computation*

Figure 4.27 – *The process computes limb flexion followed by limb realignment. Once the flexion is determined and the realignment applied, the twist is adjusted. For a large eccentricity of the intermediate joint reprojected position from the anchor effector axis, the swivel angle is determined by the radial direction of the retro-projected joint (not necessarily on the same disc as the current bone's right). Conversely, when the retro projection of the intermediate joint is close to the effector anchor axis, the swivel retained is the one from the source skeleton to avoid instabilities in the animation. In between, a linear interpolation is performed.*

Finger animation convergence loop

The finger animation pipeline is mainly similar to the one for the user's skeleton animation subsection 4.6.1; to the difference that this pipeline also uses inputs from the user's skeleton's hand model. Each finger possesses its animation pipeline composed of an inverse kinematic with the flexion determined by the effector base joint distance and a realignment based on the yaw-pitch of the target point's position (p_j).

As for the effector orientation, the avatar finger animation is either sourced from the raw user's skeleton hand animation or computed based on the avatar's fingers' IK fed with the retro-projected fingertips input.

The value chosen for this selection ratio is $\max\left((\lambda_{i,j})_{j\in\mathbb{S}}\right)$ for j the fingertip's target's index. Rule chosen for the same reason as for the effector orientation, which is that high $\lambda_{i,j}$ highlights an essential contribution to the pose to reconstruct.

This finalizes the pose reconstruction of the avatars. Figure 4.28 presents some of the results obtained from a different set of poses involving self-contacts and shows the results alongside those obtained using forward kinematics. In this figure, the source model from the user is illustrated through its surface elements.

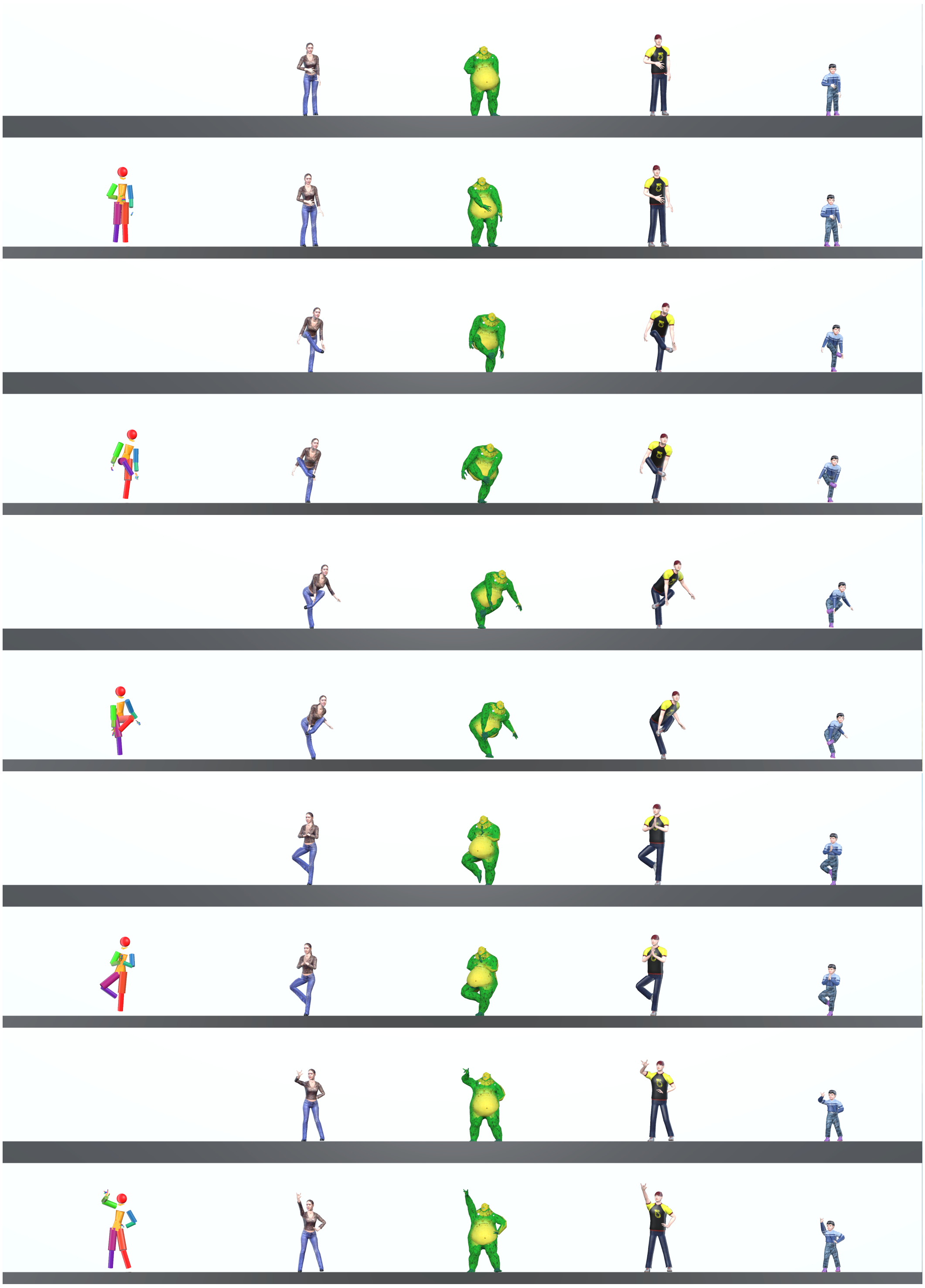


Figure 4.28 – *This figure shows a set of poses in pairs of two rows, with the upper row representing animations using forward kinematics and the lower row (the one with the source user skeleton model on the left) representing the retargeted animations.*

4.7 Discussion and future works

Instabilities One of the primary issues observed in the animation pipeline is the instability of the resulting animation. The instability in the animation sequence is due to the fact that each frame is generated independently of the previous one and only relies on the transformed input from the MoCap systems. This is compounded by the fact that target points are the result of an iterative process (the retro-projection of target points followed by the application of IKs), which can amplify even a small change in the input at each sub-iteration. In the end, this produces a result lacking continuity with the previous frame, hence resulting in the unstable character's motion and noticeable jitters in the animation sequence.

To address this continuity issue, the inputs from the MoCap could be filtered using approaches to avoid a jitter propagation across the whole pipeline. Low-pass filtering could also be applied to each iteration of retro-projected target points to enforce the animation's stability. However, using a low-pass filter would inherently introduce a slight delay in the animation but still might improve the user experience.

Limiting the displacement of each retro-projected target point in regard to the source target's motion would be another possible option to damp potential instabilities. This would keep static points on the source avatar static on the retargeted character and limit jitter to a small amount of inter frames variations. This technique could also be valuable for addressing the second and common issue observed in the avatar animation, which is the phenomenon of "Footskate".

Footskate This issue refers to the problem of feet sliding or floating unnaturally across the ground during movements rather than making proper contact with the ground (Glardon et al., 2006), hence negatively impacting the realism and believability of character movements. In our current approach, only the vertical height of the feet is constrained; therefore, nothing prevents the characters' feet from sliding on the floor surface.

The solution proposed in Lyard and Magnenat-Thalmann (2007); Mourot et al. (2022a) to increase the overall naturalness of the pose is to permit an offset to exist between the user and the avatar location, a difference that could go unnoticed if the gain in displacement remains reasonable (Steinicke et al., 2010).

Crude mesh resolution Another limitation of the current approach is the small density of the crude mesh used to animate the avatar. Even after increasing the density of the crude mesh on the lower belly, we observed that the surface area of the belly still exhibited some slight inter-penetrations. Addressing the remaining small gaps between approximated surfaces and the actual surface could involve another topology that could switch surface elements based on specific needs, such as using triangular-based elements for flat surfaces and spheres for rounded

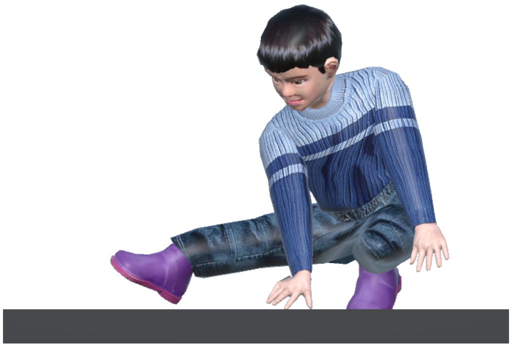


Figure 4.29 – *Example demonstrating a missed ground contact due to the target point for the right ankle being out of reach.*

surfaces. A mapping between these surfaces would also need to be established. For example, a user with a flat belly could have a triangular surface approximation, while the avatar could have a spherical belly approximation when having a large one. In this case, the triangular coordinates from the source character would need to be mapped to spherical coordinates before being applied to the avatar. To optimize this process, a partial pre-computation of the mapping could be done to assign triangles to spherical subsections and vice versa.

Another possible solution to address this issue is to implement an iterative sub-division of the defined triangles to better approximate the surface of interest from avatars or users. For instance, the current representation of the upper chest of avatars only uses a single point which might not be sufficient to represent prominent upper chests. This sub-division could be done iteratively until a desired level of accuracy is achieved while also considering computational efficiency.

Kinematic path normalization and movement correlations It was observed that the legs' target points could be drawn to the sides during large arm movements when the user's legs remained stationary or that the target points for legs could become unreachable during extended leg movements, causing the loss of self-contact consistency with the floor Figure 4.29.

To address this limitation, a more conservative approach comparing the limb extension of the source and targeted characters could be adopted. It would balance the priority between maintaining self-contact consistency in near-contact situations and preserving the overall semantic correspondence otherwise.

Setup and calibration Equipping a user and performing the calibration requires around 20 minutes despite the semi-automated process. However, this process could be greatly improved by calibrating the user using computer vision to infer joint locations and body surface elements during a single (or a few) static pose(s). Furthermore, four of the trackers used in the tracking setup (those placed on forearms and lower legs) provide redundant information that could be removed to lighten the setup for the users.

4.8 Conclusion

In summary, we proposed an animation method, extending the work from Molla et al. (2017), taking advantage of human tolerance to motion discrepancies to address the issue of interpenetrations in the animation of an avatar now with both body and finger levels. With pre-calibrated avatars, our approach only requires retrieving the user's skeleton structure and body shape through a calibration process. Once the user's model is calibrated, the posture is stored in a normalized form of joint angles and normalized relative vectors between target points and the body surface. This normalized form is then used to iteratively attract the avatar's posture toward the applied normalized posture on the avatar's structure at the limb level to provide a posture that respects self-contact consistency at the body level, followed by a second iteration loop dealing with the convergence of fingers pose. Finally, this method's observed drawbacks were discussed, and we proposed solutions to address those issues in future work.

5 Subjective evaluation of the full body animation

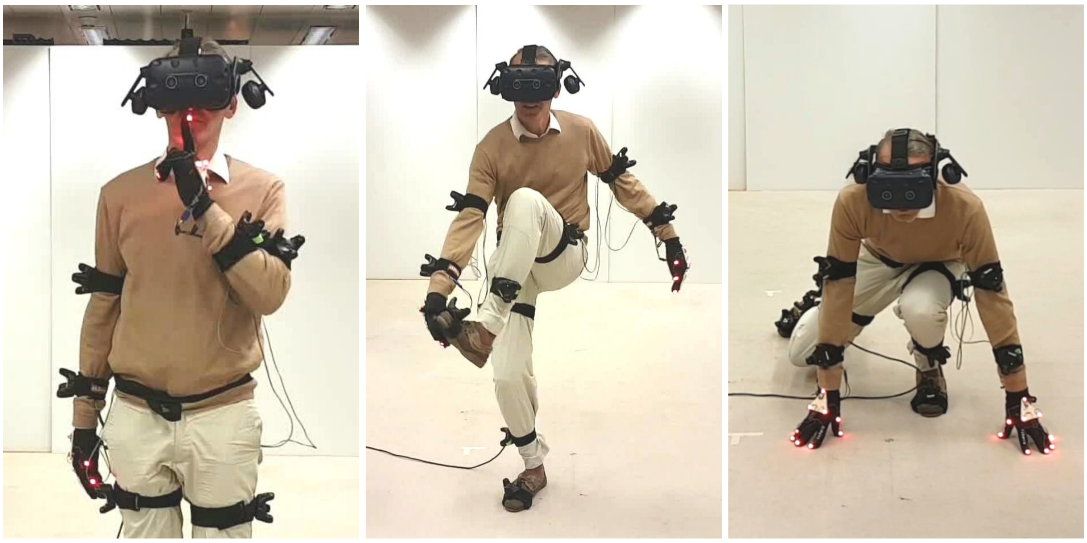
Our movements convey semantics and meanings that are easily perceived by others. Failing to remap the user's movement onto the avatar might lead to failure at the interaction and convey the wrong semantics to the other. In particular, it was shown that the self contact consistency was an important factor in producing a convincing avatar animation (Bovet et al., 2018; Basset et al., 2022). In immersive VR, the small field of view combined with the opacity of the virtual body limits our ability to observe all of the movement of the virtual avatar, hence allowing mismatches to occur while remaining invisible to the user wearing the HMD.

Therefore, it is essential to know whether our approach provides a plausible animation that conveys the correct intended semantics. Consequently, we compared it to direct kinematics (i.e., where the avatars' joint angles are directly sourced from the source model) through a subjective evaluation using the third PV (Person Viewpoint). This evaluation consisted of showing three videos to naive observers: one showing a recorded source pose through a camera and two recorded videos using each approach with a similar viewpoint, and then asking the participants to evaluate both animations. This setup was similar to the one proposed by Molla et al. (2017) and maximized the overall view of the virtual body, hence the ability for participants to spot inconsistencies in the animations.

5.1 Video dataset

To construct the database used in the comparison, we asked two persons to be equipped with the tracking system, to perform their body calibration, and then to be recorded while performing movements to reach predefined poses. Those poses were chosen to carry semantics (e.g., placing the finger in front of the mouth, placing the hand near the ear, placing the hand in front of the mouth to express surprise Figure 5.1a), poses with self-contacts (e.g., both hands touching each other, hand foot contact Figure 5.1b), or poses with contacts with the floor (e.g., crouching with one hand on the floor Figure 5.1c).

The two persons recorded were a man and a woman, on the thinner side of what can be considered



(a) *Shussing pose* **(b)** *Hand-foot contact* **(c)** *Floor contact*

Figure 5.1 – *Example of semantic poses conveying different types of interactions*

regularly shaped, and the list of the recorded poses is illustrated in Table 5.1.

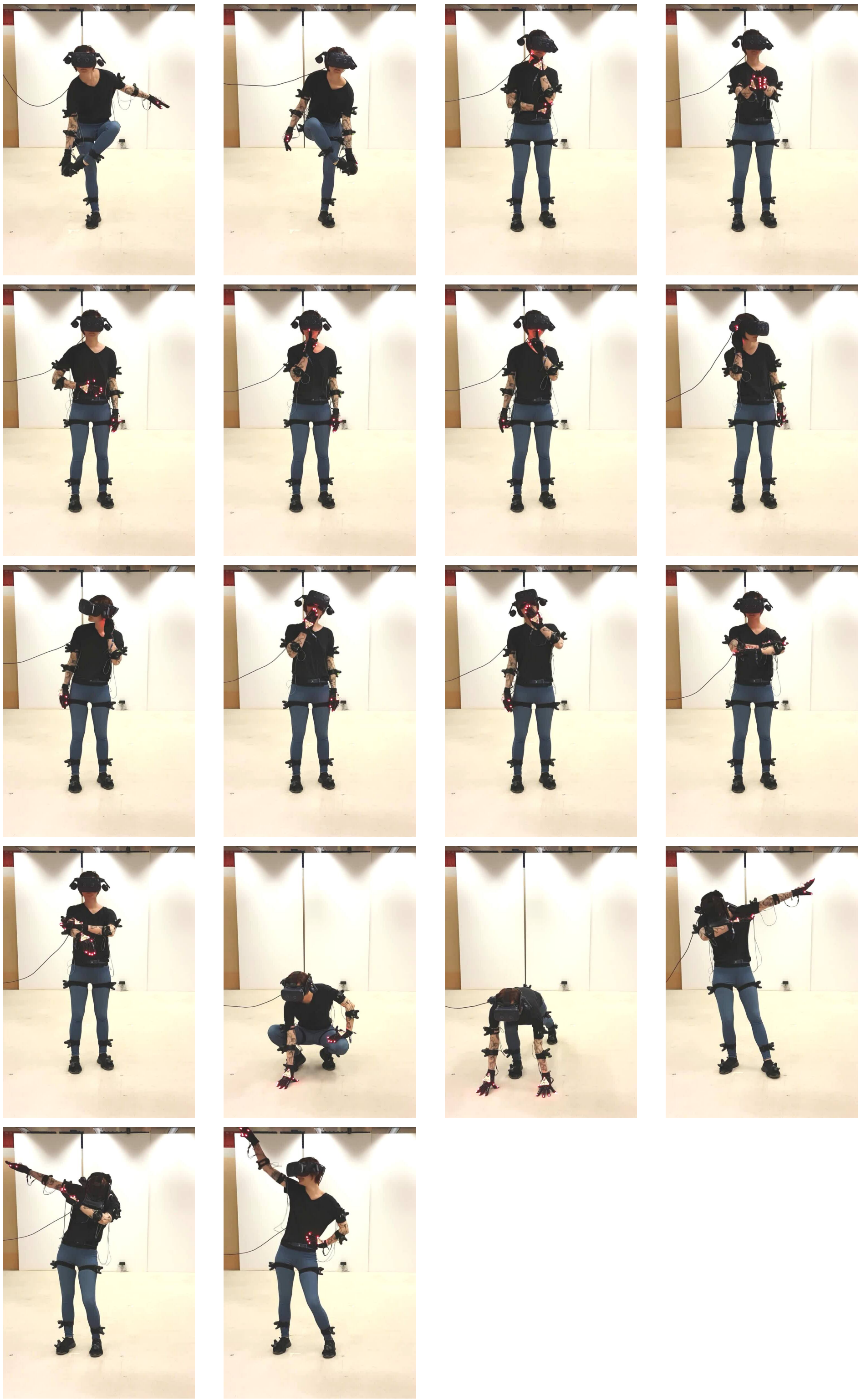
With the recorded samples, we applied the recorded MoCap using both methods on four avatars: a tall male, a woman, a small child, and an ogre with a large belly (Figure 2.2). All of the animations were recorded simultaneously at a lower capped framerate to avoid timestamps mismatches in the processing of extracting video segments later displayed in a randomized order to the participants. Then, those segments were sliced into separate video streams for each avatar with each animation method and then cut at the manually annotated timestamps placed at the beginning and end of each movement. Each of the movements is then presented four times to the participant, not necessarily in consecutive order, each time with a different avatar, hence producing a total of 152 individual motion clips. In addition to the recorded animations, the real movements of the user were recorded using a camera and sliced together with the generated animations to produce the corresponding reference video.

The animation was rendered using an orthographic projection. Due to the physical dimension of the room, using a telelens to approximate an orthographic view was not possible. Therefore, we framed the view of the camera so that the viewing direction is the same and that, when placed in the center, the participant would cover most of the captured zone, with a bit of padding to accommodate for displacements.

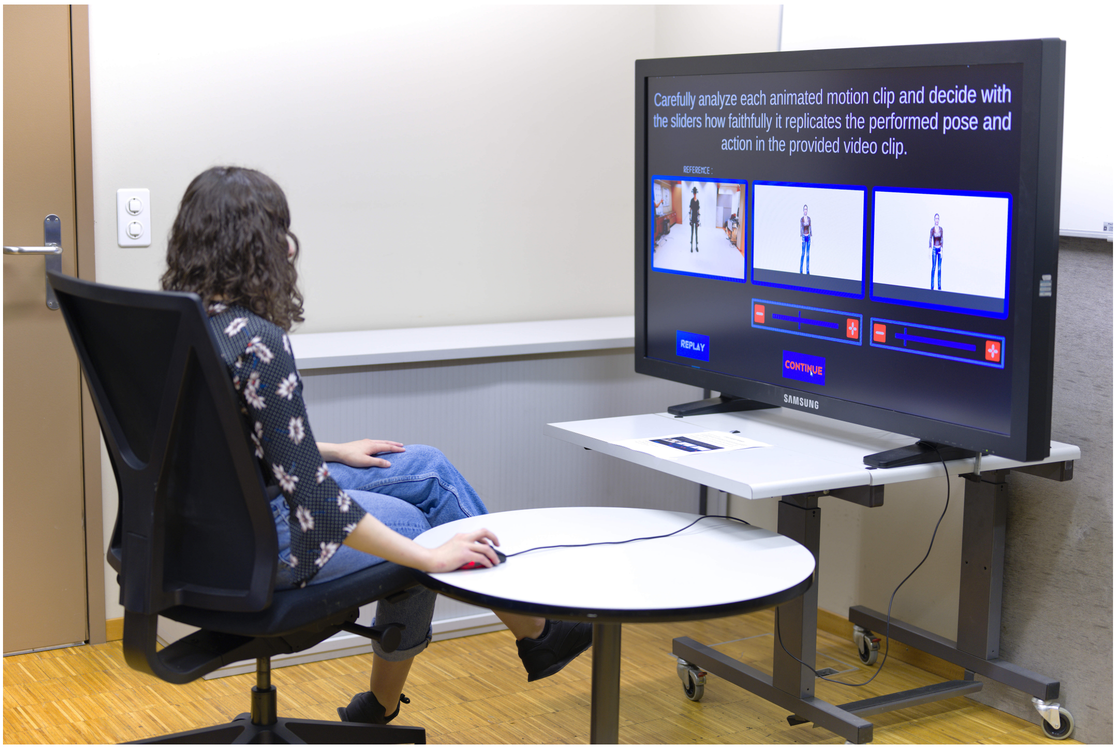
5.2 Subjective evaluation procedure

Before participating in this study, participants were informed about the task of the study and were asked to give their written informed consent and filled out an anonymous demographic

Table 5.1 – *Illustration of the targeted poses used for the evaluation*



questionnaire. Participants were then placed in front of a large monitor (to enlarge the videos), the room was then set in the dark to remove potential reflections or distracting elements to provide the best viewing conditions, and an interface was presented to the participants, explaining that two animation motion clips are presented side-by-side and that the participants' role would consist of "Carefully analyzing each animated motion clip and decide with the sliders how faithfully it replicates the performed pose and action in the provided video clip" (Figure 5.2).



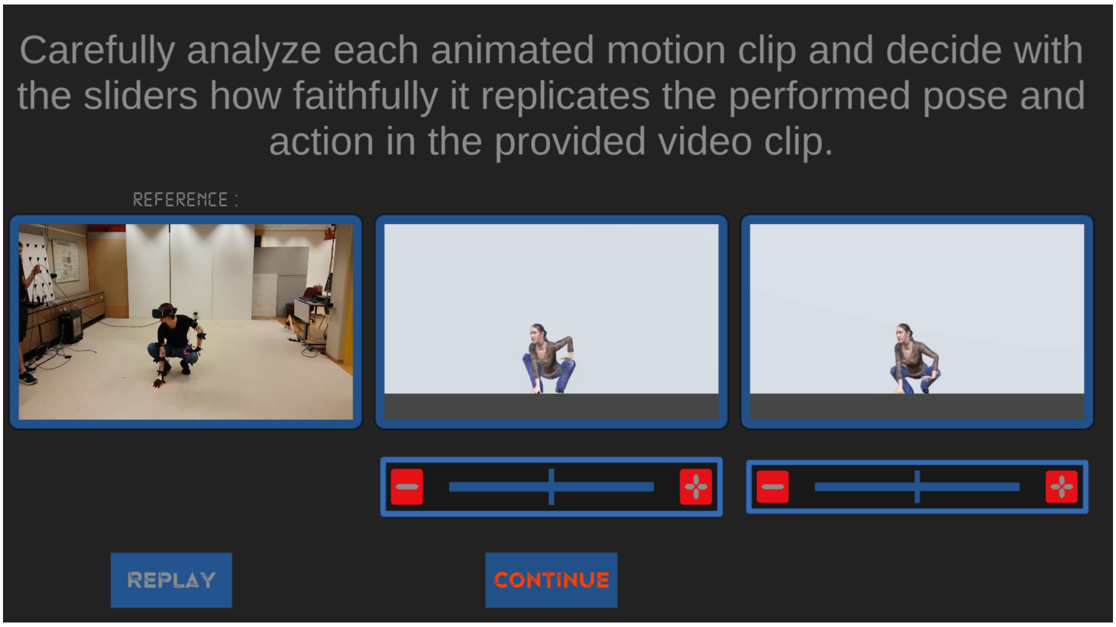
Carefully analyze each animated motion clip and decide with the sliders how faithfully it replicates the performed pose and action in the provided video clip.

Figure 5.2 – *The participants were seated comfortably in front of a large screen that displayed the videos. They were given the option to take breaks, and to avoid any potential issues with light reflections, the room lights were turned off (unlike what is depicted in the picture).*

The retargeted clip and the non-retargeted one were randomly swapped, and the continuous sliders were used to collect the evaluation scores. Participants could also replay the clips as many times as required to ensure they could consider every detail from the clips.

The interface used a regular desktop mouse placed on a low table for more convenience. Participants were finally told that they could ask questions during the experiment, such as querying the experimenter for the number of remaining clips or if they had any questions related to the experiment.

Once the experiment started, the clips were presented in a pseudo-randomized order (each pair of motions and avatar was only presented once) and the scores were recorded (with their time stamps) to constitute the analysis dataset for comparing the two approaches.



Carefully analyze each animated motion clip and decide with the sliders how faithfully it replicates the performed pose and action in the provided video clip.

Figure 5.3 – *Screenshot of the displayed interface to the user on the full right: the original action performed by a person, on the two middle and right: the animated avatar using either the approach with the retargeting pipeline enabled or with only raw angles provided by the user's skeleton input. The order between the two animation methods is randomized, hence unknown to the participant. Two continuous sliders are displayed below the videos to allow participants for individual video evaluations. Finally, the participant can replay the videos as much as they want and validate their choice using buttons from the interface.*

At the end of the experiment, we asked participants to report which criteria they used, by order of priority, to evaluate the proposed motion clips and received their overall feedback from the experiment. Ultimately, participants collected their financial compensation for their time of CHF20/h and were offered a small snack.

5.3 Analysis

The analysis was conducted using R. To verify the presence of an effect linked to the animation method factor, we performed a pair-wised comparison between both samples drawn from the retargeted approach and the direct kinematic one. The analysis pipeline was performed on both the full dataset as a whole and on targeted avatar subsets. The test used for the comparison was the parametric pair-wised two-sample, two-sided Student's t-Test, which evaluates the null hypothesis of the equality of the two means of the samples. Therefore, before applying the test, we verified the required hypothesis for this test to be performed. We first verified that each sample was normally distributed using the Shapiro tests, hence allowing parametric methods to be used, and assessed the homogeneity of the variances through the parametric Hartley's Maximum F-Ratio test. Finally, a test on the direction of the difference was performed in post-hoc using the one-sided version of the t-test, and the effect sizes were measured using Cohen's D.

5.4 Results

Demographics In total, 20 participants (15 women), aged between 20 and 30 years old (average: 21.7, std: 2.43), participated in this study. Those participants were mostly not used to playing video games (75% reported never or rarely playing action video games). Participants often practiced sports such as dance and were in the large majority (17 out of 20) right-handed. On average, the experiment took one hour to be completed.

Evaluation scores The results from the Shapiro tests, presented in Table 5.2, indicate that all samples were normally distributed, allowing parametric tests to be performed.

Table 5.2 – *p-Values for Shapiro tests on evaluations scores; we can observe that all data were normally distributed*

Condition	All	Ogre	Child	Men	Woman
Direct Kinematic	$2.90 \cdot 10^{-25}$	$3.82 \cdot 10^{-12}$	$2.05 \cdot 10^{-12}$	$8.13 \cdot 10^{-11}$	$4.90 \cdot 10^{-12}$
Retargeting	$1.20 \cdot 10^{-28}$	$3.56 \cdot 10^{-11}$	$1.03 \cdot 10^{-15}$	$4.97 \cdot 10^{-13}$	$1.15 \cdot 10^{-14}$

The homogeneity test did not reveal any significant difference in the variance distribution across the different paired samples between the retargeted approach and the direct kinematic animation. Therefore t-test comparisons were performed, and the effect sizes were measured using Cohen's D formula. Due to the observed significant differences, the post-hoc analysis was conducted. All of the test values and effect size measurements are reported in Table 5.3 and illustrated in Figure 5.4.

Here, we can observe that, among all situations, the score given by the participants were on the upper side of the range and, in all situations except the ground interaction, the scores for

Table 5.3 – *Test and measure results for the scores of the evaluations between the Direct Kinematic and the Retargeting evaluation scores*

Test / Measure	All	Ogre	Child	Man	Woman	Semantics	Self-Contact	Ground
Hartley's maximum F-ratio	0.92	1.00	1.00	1.00	1.00	1.00	0.99	1.00
t-Test (two-sided)	$2.98 \cdot 10^{-25}$	$1.30 \cdot 10^{-12}$	$3.73 \cdot 10^{-11}$	$1.01 \cdot 10^{-2}$	$1.75 \cdot 10^{-6}$	$< 2.2 \cdot 10^{-16}$	$< 2.2 \cdot 10^{-16}$	$2.28 \cdot 10^{-11}$
t-Test (greater)	$1.49 \cdot 10^{-25}$	$6.50 \cdot 10^{-12}$	$1.87 \cdot 10^{-11}$	$5.07 \cdot 10^{-3}$	$8.75 \cdot 10^{-7}$	$< 2.2 \cdot 10^{-16}$	$< 2.2 \cdot 10^{-16}$	1.00
Cohen's D	0.27	0.37	0.34	0.13	0.25	0.38	0.31	0.63

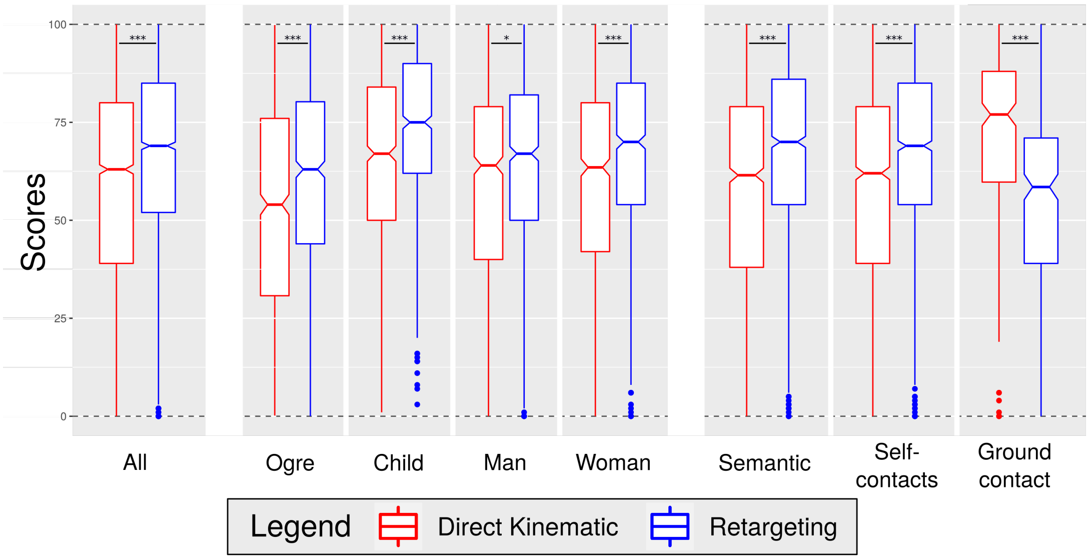


Figure 5.4 – *Normalized boxplots of the participants' evaluation scores.*

the approach with the retargeting enabled were higher for all the postures and across all of the destination avatars.

In addition to the measured data, the collected participants' spontaneous feedback highlighted the role of instabilities in some of the animations, and only nine participants, out of the twenty who participated in this study, placed the interpenetration as their first concern when evaluating the animation, while the rest were more focused on the fluidity of the movement, and on the overall posture semantic associated with the pose.

5.5 Discussion

The subjective evaluation showed that, except for the ground interaction, our retargeting approach was significantly preferred over the direct kinematic animation pipeline. Consequently, and despite the small effect size, given the limitations of our approach in terms of smoothness, we can expect the observed preference towards our retargeted approach to be driven by the ability of the system to maintain the self-contact consistency known to be a critical point for the animation of a 3D character Bovet et al. (2018); Basset et al. (2022).

Additionally, we can notice a link between the measured effect size (Table 5.3) and the discrepancy between the performer and the avatar's body: when both the avatar and the user share more or less the same morphology and proportions, the contribution of the retargeting might become less relevant (small effect size for the man and the woman avatars), and those conditions can therefore act as a kind of control condition. However, even in this case, the retargeted approach was preferred over the direct kinematic one, hence suggesting that simply relying on direct kinematic animation may not be adequate, despite the similarity between the avatar and the user. Conversely, when the avatar's body differed more (for the child or the ogre), the observed effect size was higher.

Finally, this study addresses the evaluation of the animations using a third PV to provide participants with the maximum amount of details on the animation. However, when provided with an avatar from the first PV, one might have more trouble seeing all of their limbs, hence reducing the risk of being bothered by some error in the animation if those are not visible, in particular, if the HMD (Head-Mounted Display) provides only a small field of view. Hence, this might further stress the importance of hand interactions and dim the concern about the overall body posture representation reinforcing the observed trend, especially considering the expected stronger multisensory integration of tactile sensation with visual feedback.

5.6 Conclusion

To summarize, we conducted a subjective assessment to compare our retargeting method with direct kinematics in generating believable avatar animations based on the movements of the source character. Participants were asked to evaluate motion clips produced using both approaches, and we performed a statistical analysis of the gathered scores. The results revealed that our approach significantly enhanced the perceived overall quality of the animations compared to direct forward kinematics. The participant feedback emphasized the role of smoothness in the animation, which was not sufficiently addressed in our approach, hence providing an opportunity for improvement. Ultimately, this user evaluation sets the stage for conducting an immersive first PV user evaluation in the future.

6 Synthesis and conclusion

VR (Virtual Reality) is a real-time simulation that creates the subjective illusion of a virtual world for the user. Failing at integrating a plausible virtual body responding to the users' moves comes at the cost of diminishing the user experience Slater et al. (2022). Tracking technologies are, of this day, imperfect and cannot accurately transcribe users' motions, especially for fingers. As a consequence, one of the standard ways to track users' movements is through the use of controllers, which, given their large size, can embed more sensors and be less subject to occlusions. However, most of the controllers do not track finger movements and, consequently, can only propose metaphoric finger animations. For those controllers equipped with proximity sensors (e.g., Valve (2019)), being held in hands inherently diminishes the allowed range of finger movements. Furthermore, controllers cannot be used in all situations. For instance, when a training requires tangible interactions with objects to correctly train the trainee (e.g., CPR (Cardiopulmonary Resuscitation) in VR Delahaye et al. (2021)), having a controller at hand can induce collisions or prevent the movement to be performed. Therefore, in this thesis, we first addressed this limitation by providing users with a hand and finger tracking solution that is real-time compliant.

6.1 Finger animation and user perception

Optical active tracking systems provide millimeter precision of markers' positions in the 3D space and have low latency Tian et al. (2015), making those solutions particularly suited to track fine levels of motions, such as fingers' one. However, markers might suffer from occlusions, making them impractical in a situation where they can easily be hidden, like on fingertips. This issue finds a solution in our approach where we rely on the predictability of finger poses to train a neural network (autoencoder) to recover the markers' positions due to occlusions. This training was performed using a ground truth system composed of the combination of IMU (Inertial Measurement Unit) placed on the tracking glove equipped with the optical tracking LEDs (IMU (Inertial Measurement Unit)s provided positions during occlusions and the optical system corrected the drift when there were no occlusions). In addition, some random occlusions were added to the training to reinforce the system for handling occlusions. In parallel, a second autoencoder was used to provide an IK (Inverse Kinematic) solution to animate hand fingers.

Our observations show that the neural networks autoencoder could handle reasonable changes in hand morphology for handling occlusions. The approach yielded convincing recoveries in the occlusions up to one or two markers occluded simultaneously for up to a second of continuous occlusion. However, we observed that the animation could still produce artifact poses with fingers pointing toward implausible positions. Consequently, we later replaced the second stage of the finger animation pipeline, which plays the role of IKs for finger, with an analytical solution described in subsection 4.6.1. This analytical solution considers the biomechanical constraints of the hands to prevent unnatural hand postures from being generated. Concerning the occlusion recovery pipeline, its autoencoder's overall topology could be adapted to extend the input with binary inputs representing the occlusion state of the markers rather than replacing occluded markers' positions with zeros. This would likely help the neural network better learn the semantics of the meaning of occlusions and therefore address those in a better way. Long short-term memory and gated recurrent unit topologies could also be investigated in parallel to take advantage of the history of the marker position to predict its next state when occluded; however, a more extensive sample set might be required to cover motion directions in addition to the simple poses.

On the other side, we also know that humans are relatively poor at locating limbs using only proprioception (Burns and Brooks, 2006). Furthermore, at the finger level, the work from Logan et al. Logan and Crump (2010) showed that, in conditions of word typing with a keyboard, one could experience the illusory authorship for the correction of typos made by the machine, but also, in a smaller measure for the introduced ones. This was observed with no visual alteration of the participants' fingers as the task was performed without VR on a regular keyboard. This higher tolerance in distortion helping user was also observed when combined with visually introduced movement distortions at the limb level in immersive VR Galvan Debarba et al. (2018); Porssut et al. (2021). Also, with the introduction of a live modulation of the visual feedback, this time at the finger level and using a non-immersive VR setup, Salomon et al. Salomon et al. (2016) showed that visual judgments were not affected by the introduction of swaps, but conversely, that participants were getting confused at saying which finger they actually moved when a swap was introduced.

Therefore, with the new possibilities offered by our finger animation pipeline, we investigated with an immersive VR setup whether users could tolerate an extremely distinct distortion: finger swaps. Assuming that if users can accommodate finger swaps, they could also accommodate two finger motion alterations at once, hence also a simple motion alteration. To also observe the effect of going in the direction or against the goal of the participant, our experimental protocol involved a game task to be performed with fingers (validating virtual buttons sliding over virtual lanes by lifting the fingers), a manipulation (the machine pseudo-randomly introduced finger swap to help or penalize the participant), and an experimental task (participants had to press a pedal each time they noticed a finger swap was introduced).

The observed results showed that participants could easily take credit for both types of finger swaps, especially when those help participants with the game task, where participants reported significantly less accuracy at noticing the introduced finger swaps. Unsurprisingly, the lower odds of perceiving the swaps compared to the reduced, but still high, accuracy at identifying the finger moved in Salomon et al. (2016) could be explained by the fact that users were no longer only focusing on a simple task but also actually playing a game, making the detection task even harder. Through the acceptance of the visually impaired feedback of the wrong finger movements, the results from Logan and Crump (2010) were replicated as the authorship of the action differed in the same direction when the swap helped or penalized the participant at achieving the task. Similarly, the lower swap detection scores compared to the scores from Logan and Crump (2010) could be due to the added incongruent visual feedback matching the expected output, making the task harder to notice finger swaps.

Overall, this study allowed us to extend the characterization of the SoE at the finger level. We observed that the tolerance to motion distortions gains, especially when it helps, for arm/leg reaching tasks (Galvan Debarba et al., 2018; Porssut et al., 2021) worked in a similar way for finger swaps, where helping the participant is more accepted than penalizing him. However, unlike the experiment performed in those studies for leg/arm-reaching movements, here, the nature of the distortion was a finger swap. Although we cannot compare the swapping of a limb movement with its opposite to the swapping of two fingers on the same hand, it is interesting to observe how different these distortions impact the embodiment. For the former, Boban et al. (2023b) observed that limb swaps in the animation would drastically diminish the scores of embodiment, while in our study, those finger swaps were bearly even noticed.

Finally, in this study, participants were primed to recognize the manipulation of finger movements. However, as this was shown in Burns and Brooks (2006), priming participants at noticing motion discrepancy makes them better at detecting those. Therefore, we can make the assumption that the observed tolerance to finger swaps would be even higher if participants were not instructed and trained to observe and notice those, hence providing guidelines to avoid disrupting the SoE when animating avatar fingers.

6.2 Full body animation down to the finger level

The learned mechanisms of the cognitive functioning of the SoE at the finger levels for finger swaps, combined with the recent knowledge from the literature on arm/leg reaching movements (Galvan Debarba et al., 2018; Porssut et al., 2021) constitute a set of guidelines to be followed if one wants to introduce distortions without having the used rejecting the animated body. Consequently, on the one hand, we can introduce distortions in the displayed movements to adjust the movements of a virtual body to better suit the user morphology, but on the other hand, we know that users are quite sensitive to self-contact consistency, both for first (Bovet et al., 2018) and third (Basset et al., 2022) PV (Person Viewpoint), making those points crucial to produce a convincing animation. Therefore, providing an avatar to the user in the VE (Virtual Environment), regardless of their morphologies, must prioritize the self-contact constraints first, especially for effectors such as hands (Basset et al., 2022), and could consequently be more tolerant on the rest of the limb's position to accommodate this essential constraint. Such an approach was designed at the body level to address those constraints through the work from Molla et al. Molla et al. (2017). Here, the authors relied on an active optical MoCap system (to acquire the user's movements), a user's body calibration procedure (to construct a numerical model of the user's morphology), and an animation pipeline to transfer the original motion from the user onto an avatar with different shapes and sizes in real-time. However, this approach did not cover the integration of the finger for the avatar.

This issue is addressed in the integration of our finger animation pipeline described in chapter 2 that extends the original approach from Molla et al. (2017). Technical switch from the original MoCap system to a consumer-grade solution implied some redesigns of the pipeline to adapt to the technology used and to take advantage of redundant information to enhance the reliability of the tracking. In this process, the topology of the calibrated user's morphology was also adapted to fit more precisely avatars with large bellies as it was observed that the initial topology couldn't handle such large cases. Finally, and more importantly, larger changes were at stake when addressing the integration of the finger-level animation on the avatar's body: The original animation pipeline did not account for finger in the model calibration, nor in the animation pipeline. Consequently, a fine-level calibration was implemented to calibrate the hand's surfaces, fingers' radius, lengths, or root joint locations. On the animation side, the original animation process, performed through a single pose convergence loop for the limbs' movements, was extended with a second loop addressing the fingers' animations using the finger animation pipeline described earlier.

The proposed technique was then assessed in a subjective evaluation comparing its output to the output obtained using just the direct forward kinematic method (i.e., where the avatars' joint angles are directly sourced from the source model). The evaluation was performed with participants evaluating pre-recorded videos of animation clips generated using both approaches, with the recorded camera video displayed as a reference. The viewpoint was set at the third PV to maximize the overall view of the virtual body, hence maximizing the ability for participants to spot inconsistencies in the animations. The results showed that the retargeted approach yielded significantly better scores than the approach using only direct kinematics, hence confirming previous results from the literature. However, the measured effect size was not as large as initially expected, indicating that improvements could be adapted to our method. Nevertheless, the results highlighted the necessity of adapting the motion, even if the avatar and the user look similar.

Among the improvements that can be applied to the method, it was first observed that the retargeted approach could induce some jitter in the animation, diminishing the fluidity of the motion. Indeed, the current pipeline generates each frame only using the sole (filtered and transformed) MoCap positions from the current frame without considering the previous frames. One efficient way to ensure smooth temporal consistency is to use low-pass filters in the animation convergence loops. Low-pass filters inherently induce latency, but this might be negligible compared to the benefits of a smooth animation. A second method to reduce the jitter could limit the avatar's limbs and finger movements based on the user's one (with a scaling to accommodate limb length differences). Consequently, if the user is standing still, the avatar would be constrained to remain also still regardless of the output of the retargeting approach. This technique could also be highly valuable for addressing another common issue in avatar animation: the phenomenon of "Footskate".

This is observed when characters' feet slide on the ground during movements where they were supposed to remain grounded (Glardon et al., 2006). In our approach, only the vertical height of the targeted feet position is constrained, subjecting our approach to this issue. Several approaches Lyard and Magnenat-Thalmann (2007); Mourot et al. (2022a) investigated this issue and proposed solutions to mitigate this effect. However, enforcing a foot's position on the ground implies that the pelvis is no longer the root of the character animation; hence the user and the avatar might move in the 3D space at different speeds. To our advantage, it was shown that we could tolerate gain in our displacement, especially when it helps to go faster, provided that those remain reasonable (Steinicke et al., 2010) hence allowing those methods to be investigated in future works.

On the self-interaction size, despite having increased the density of the crude mesh on the lower belly, we could still observe some areas of the belly exhibiting slight inter-penetrations. This issue comes from the approximation of rounded surfaces using small triangles; maintaining a small maximum distance between the discretized surface and the original one might require a lot of triangles. As the pipeline computes many projections, coordinates, or kinematic chain normalization factors, on both the user's surface and the avatar's one for each surface element, increasing the number of elements linearly augments the execution time of a heavy workload on a CPU. Using a GPU, the hardware-implemented parallelized processing structure allows, up to the maximum set of parallel threads, to maintain the temporal complexity of the computations constant. However, the cost of transmitting the data between the CPU and the GPU linearly increases as the number of elements transmitted increases. Consequently, attention should be put on not using too many surface elements and, therefore, identifying correctly the areas of interest where a higher density of surface elements is required.

Finally, the long calibration process combined with a heavy MoCap setup (eleven trackers are required for the calibration, seven for the runtime, plus two pairs of tracking gloves) makes this approach a bit heavy to set up. To this day, this MoCap (Motion Capture) setup cannot yet be replaced with computer vision due to its limitation at tracking fast movements (Li et al., 2022). However, with a static calibration pose, an adaptation module could be implemented to use computer vision to replace the current tedious calibration process. In the same vein, by anticipation of advancements in computer vision technology, the pipeline was implemented to accept a different set of MoCap input so that, when computer vision would be sufficiently fast and reliable, it could be used instead to make the setup simpler.

6.3 What about first-person viewpoint subjective experience?

Through the evaluation presented in chapter 5, we addressed the subjective experience from the third PV in a passive context where the participant had no motor control over the animations of the 3D characters. When switching to the context of embodying an avatar in the first PV, the multisensory integration of the tactile sensation with the visual feedback is much richer; thus, maintaining self-contact congruence would become even more important (Bovet et al., 2018; Gonzalez-Franco and Berger, 2019). However, additional parameters must be taken into account in the first PV context.

For instance, in real life as in immersive VR, having a (virtual) body is a source of occlusions as we cannot see through our own body and some body parts might potentially hide others. Consequently, if an animation method was expected to fail in those situations, the errors would be unnoticed by the user, hence not penalizing the user experience. Therefore, in a similar way that foveated rendering only renders at high resolution the elements in the eye focus and the rest at a lower resolution, our approach could benefit from a similar technique to increase performances with a high-resolution mesh to address interactions elements in the field of view, and a low resolution one for elements the elements that are non directly visible at the first PV. This could be particularly useful as self-interactions occur relatively close to the eyes of the user in first PV (the maximum distance cannot exceed the body's size), hence making it easier for the user to spot even slight discrepancies that would go unnoticed at the third PV.

Another point that could be considered when switching from the first PV to the third PV is the head's movements. Our eyes are anchored in the head thus, manipulating the head's movement implies manipulating the viewpoint. Therefore, discrepancies between the two views (the one the user would have without the HMD and the one from the avatar) might be introduced in a context where an avatar would have a longer neck than the user's one. If discrepancies can go unnoticed for users seated in a simple environment in terms of translations or rotations (Jaekl et al., 2002), in the long term, those inconsistencies can possibly induce motion sickness to occur, making the system impractical. Typically, we already know that introducing rotations in the yaw axis induces more motion sickness than translational motions (Tian et al., 2023), the reason for which, in the implemented animation pipeline, we enforced the head orientation to be precisely the same as the user's head. However, another discrepancy in the viewpoint can occur: the feet's motion is an important aspect of the plausibility of an avatar (Debarba et al., 2020), and fixing the footskate issue comes on par with introducing displacement discrepancy. However, if both distortions can be individually accepted (Jaekl et al., 2002; Steinicke et al., 2010), it is not guaranteed that the combination of both would not result in a situation where the user would lose the sense of balance, feel sick or simply reject the PSI (Plausibility Illusion). Therefore, further studies need to be performed to relate to the acceptance of such a combination of discrepancies. In the meantime, it is essential to set the priority on colocating the virtual head's position with the actual user's head one within the body referential.

Assuming the practicability of the approach, the possibilities of the retargeting approach are numerous. Firstly, this approach could be used to determine how far a user can embody an avatar whose morphology is larger, thinner, shorter, or taller, but also test different body/legs or body/arms ratios, genders, and skin tones, to create a map of the effects and ranges of acceptance. Those maps could be put in regard to the original user's morphology to know if the transfer can be performed in both ways or if the user could more easily accept thinner avatars than larger ones, for instance, and see if this could be linked with societal acceptance factors.

In another direction, it was observed that user immersed in VR could synchronize their movements based on the ones from the avatar (Kannape and Blanke, 2013; Boban et al., 2023a), raising the question of the consequence of embodying an avatar with a different morphology on our own movements; would we perform actions differently with a different virtual body than we would normally do?

In the study Appendix D, we investigated a locomotion technique where the user could scale up and down to ease navigation in the virtual environment without teleportation. But what if we could actually continuously morph our avatar, through an adapted version of the retargeting approach, to allow the user to progressively scale in size, or even between completely distinct morphologies? Would this allow us to further extend the limits of embodiment?

6.4 Conclusion

In conclusion, we explored the perspective of integrating the user's body and fingers into the VE (Virtual Environment) on the user's subjective side. We investigated an approach relying on an active camera-based MoCap (Motion Capture) system, combined with trained auto-encoders, to address the important animation of virtual hands and fingers in real-time.

To confirm the usability of our approach, we investigated, through a user study, whether one could tolerate those errors in the finger animation through the evaluation of an even more distinct type of distortion than motion amplification in the context of succeeding interactions: finger swaps. Our experimental protocol involved a game task to be performed with fingers, a manipulation (finger swaps), and an experimental task. The analysis of the data showed that participants could easily take credit for finger swaps, in particular when those swaps help the participants with the game task, under which condition, participants bearly noticed the finger swaps. This extended the knowledge on the characterization of the SoE (Sense of Embodiment) at the finger level and allowed us to provide guidelines to avoid the disruption of the SoE when dealing with fingers animation.

This knowledge was then combined with the one from the literature on arm/leg reaching movements to extend a full-body animation pipeline addressing the critical issue of self-contact consistency for both the limb and the finger levels. In the integration process, the topology of the calibrated user's morphology was adapted to address more pronounced morphology variations, and more importantly, the original animation process, performed through a single pose convergence loop for the limbs' movements, was extended with a second loop addressing the fingers' animations.

The proposed technique was assessed in a subjective evaluation, comparing it to direct forward kinematics. Twenty participants performed the evaluation of pre-recorded videos of animation clips generated using both approaches, with the reference user's motion. The analysis reported that the retargeted approach outperformed the direct kinematics forward one. Despite the smaller effect size observed than initially expected, the evaluation highlighted the necessity of adapting the motion, even if the avatar and the user look similar.

Finally, we discussed the potential implications of this approach when used in first PV. For example, how it could be used to help characterize the limits of embodiment with huge motion discrepancies, how one could self-attribute and change their behavior depending on the target character, and why not, what could happen if their avatar would progressively morph into another one?

A On the Importance of Providing a Tangible Haptic Response for Training Cardiopulmonary Resuscitation in Virtual Reality

Full reference:
Delahaye, M. G., Zbinden, B., Herbelin, B., & Boulic, R. (2021, December). On the Importance of Providing a Tangible Haptic Response for Training Cardiopulmonary Resuscitation in Virtual Reality. In ICAT-EGVE 2020-International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments (pp. 17-25). The Eurographics Association. https://diglib.eg.org/handle/10.2312/egve20201255

B Training in VR with tangible haptic elements: when controllers become a limitation (e.g., CPR)

B.1 Introduction

It is now well accepted that human intelligence relies on Embodiment as defined in Pfeifer and Bongard (2006) as "the idea that the body is required for intelligence". Likewise, computed-mediated interaction has evolved from the traditional desktop metaphor to integrate embodied interaction as a powerful means to achieve new classes of tasks leveraging on our full-body synergies and skills Dourish (2001). This is one of the core contributions of VR to take advantage of users' full-body movements while displaying a plausible scenario within a virtual world; the goal is to make them behave as if they were experiencing the real situation Manganas et al. (2005). Such an approach is particularly useful to train individuals to react correctly to stressful situations, e.g., an emergency requiring to perform first aid in case of sudden cardiac arrest Lemaire (2018). In that specific context, the full training includes mastering two types of knowledge: the procedural knowledge of the correct sequence of actions to perform, e.g., first calling the emergency service if the victim is not responding, and the coordinated movement knowledge (skill), e.g., the cardiac massage. However, the question remains as to whether the sole visual immersion is sufficient for the skill training or whether the haptic component provided by a tangible mannequin is necessary.

The feasibility of such skill training has been shown to be possible in Semeraro et al. (2009). In the present paper, we focus on cardiac massage skill training in immersive VR by examining the impact of the two following factors: Haptic feedback (with/out) with the mannequin device from Brayden BraydenManikin (2019) (Figure B.1) and Real-time Performance feedback (with/out) in the HMD (Figure B.2 right). The performance criteria mainly consist of the amplitude and frequency of the cardiac massage during a standardized two minutes duration. This duration is recognized by rescuers as the best duration to reduce turnovers breaks while maintaining a good quality of movement.

Beyond assessing the impact of training with a tangible mannequin, we wish to ensure that, if really necessary for ensuring a correct skill transfer, such a piece of hardware remains the simplest possible. In that frame of mind, we chose to track the location of the top hand (Figure B.2 left) with an HTC-Vive tracker so that the same low-cost measurement system can be used independently from the mannequin device. Many studies used such an approach using trackers to acquire performance data, but surprisingly, few studies took care to assess the fidelity of the measured data. For instance, Buttussi et al. (2020) used an instrumented mannequin tracked in VR but did not analyze the probe data assuming the match between the tracker data and the probe one as proposed by Semeraro et al. (2019). Thus, to address this lack of validation, we calibrated the internal probe of the mannequin and compared its results with the one from the HTC tracker.

The mannequin location itself is tracked with a second tracker (Figure B.1) to ensure consistency with the victim's virtual body location.

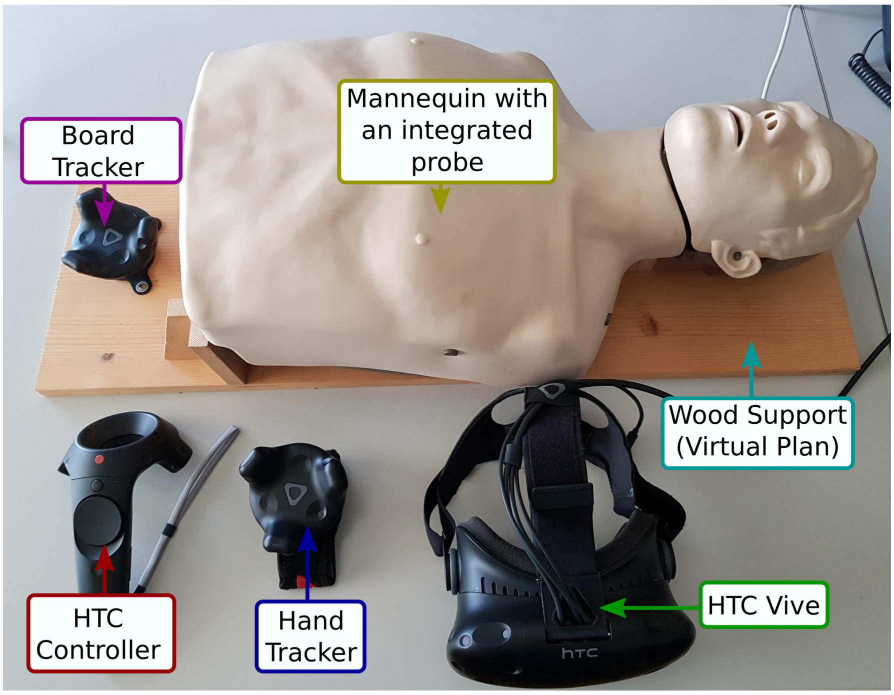

Board Tracker
Mannequin with an integrated probe
Wood Support (Virtual Plan)
HTC Controller
Hand Tracker
HTC Vive

Figure B.1 – *The CPR mannequin BraydenManikin (2019) is tracked through the HTC-Vive tracker mounted on the wooden support to align the virtual victim's body with the mannequin. The same type of tracker is attached to the dominant hand. Finally, an HTC controller is only used to launch the application.*

The purpose of the chosen setup is twofold. Firstly we want users to benefit from a sufficient level of *presence* Schwind et al. (2019) through the HMD visual immersion by cutting them from the potential distractions of their real surroundings. As advocated in Lemaire (2018), the presence dimension is critical for training to reduce fears and taboos related to the action of resuscitation. Secondly, we want to ensure that users also feel a high level of *agency*, i.e., the Embodiment component characterizing the feeling of *being in control* Kilteni et al. (2012) of the user avatar hand movement. This is achieved by tracking and displaying the top hand during the massage performance (Figure B.2 left). With only two trackers, our approach contributes to reducing the

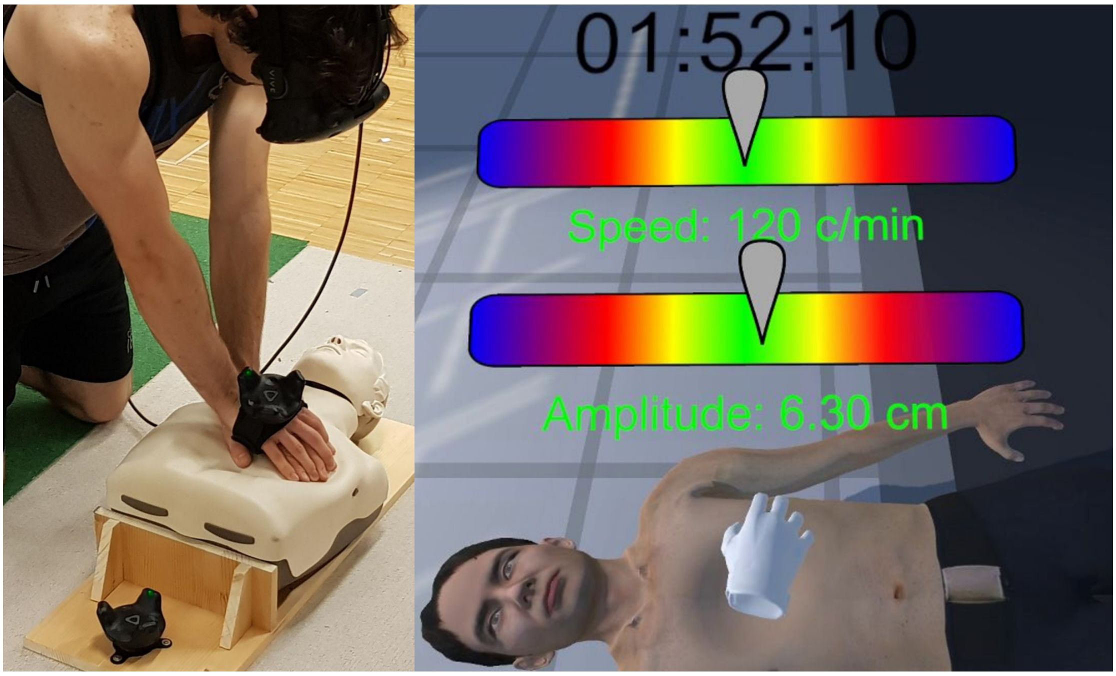


Figure B.2 – *Cardiopulmonary Resuscitation training in VR: setup with the tracked mannequin device and tracked hand (left) and 1PP view with performance feedback provided in the HMD (right)*

cost and the complexity of the complete setup as only a non-instrumented, hence more affordable, mannequin with a regular VR kit is then sufficient.

Inspired by Cummings and Bailenson (2016), we expect this minimal immersive setup to be sufficient to elicit presence. Likewise, we expect it to elicit a sufficiently high level of Embodiment through the agency component.

Our additional hypotheses associated with the evaluation experiment are the following: First that the use of the tangible mannequin benefits the quality of the performance, second that the combination of the mannequin use with performance feedback in VR leads to a performance increase and third that the performance display reduces the level of presence compared to the context without the performance feedback.

The remainder of the paper is organized as follows: After the related work section, section B.3 presents the system overview with a special emphasis on the validation of the tracked hand measurements for evaluating cardiac massage performance. It is followed by the pilot evaluation experiment description and results in section B.4 prior to the concluding discussion.

B.2 Related work

In 2004 the pioneering work from Manganas et al. (2005) demonstrated the interest of VR for first aid training in a virtual environment populated with virtual agents. The immersive display consisted of a vertical stereo retro-projected 3m x 3m screen. The user could navigate, interact with virtual agents and react to events representative of a stressful situation under the global supervision of an external operator. An evaluation based on two scenarios demonstrated the ability of the system to instill a sense of presence. However, at the time, the technology was not mature enough to involve the user to the point of performing first aid actions on the virtual victim. Instead, they would interact with other virtual agents present in the scene to instruct one of them to perform the CPR. As a consequence, this system was more suited for training the first aid procedural knowledge rather than the CPR skill itself.

The medical education field has offered a wide range of simulation solutions Maran and Glavin (2003) with some degree of success in offering applications displaying haptic feedback in VR such as with laparoscopic simulators or for training breast exams by employing a mannequin together with a virtual agent in VR Raij et al. (2009). Yet CPR training has been limited to the non-immersive manipulation of instrumented mannequin BraydenManikin (2019) or immersive VR without mannequin Lemaire (2018). In 2014 Kwon et al. (2014) proposed to use a mannequin with Augmented Reality to deliver information about the scenario and the user performance. However, the solution remains mostly 2D hence reducing the sense of presence. A similar approach was also proposed in Javaheri et al. (2018).

More immersed simulations were studied in Khanal et al. (2014) where authors compared a regular face-to-face team training against VR enhanced approaches and observed a similar learning experience. However, this study focuses more on the team training rather than on the CPR massage in itself. Almousa et al. (2019) mixed VR with a real mannequin but focused more on benefits from gamification and training availability rather than technical validation of the approach or on the training quality. In Yang et al. (2020), authors also used a similar approach but replaced the mannequin with a tangible "force sensitive model" and only used a piezoelectric element to capture compression rate; thus, they do not have access to the real depth compression range.

The contributions of the approach we propose are the following: ensure a sufficient level of presence while allowing correct skill training through the interaction with a mannequin dedicated to CPR training, prevent break in presence by applying a deformation to the virtual victim torso consistent with the user hand movement, ensure a high level of agency over the interaction with the virtual victim body through a minimal embodiment while using only a consumer ready VR setup plus a generic mannequin providing haptic feedbacks.

To ensure the fidelity of the system, we took advantage of the integrated probe inside of the mannequin to use it as a ground truth. Compared to Bergeron (2019), where an external visual motion analyzer is used as a reference, using the inner side of the mannequin gives more straightforward information with less risk of errors due to occlusions and is more natural in the sense that the heart, during CPR is compressed by the inner side of the body. In the same paper, authors provide a higher granularity for hand and finger movement using a Leap motion. However, they do not use this information to compute performance. Instead, their setup uses an additional accelerometer to achieve this goal.

B.3 System overview

In immersive VR, the users are visually cut from the real world, i.e., they do not even see their own bodies. For cost and efficiency reasons, we adopted a minimal embodiment strategy of tracking a single hand to allow users to adopt the standardized CPR hand postures that include both hands. For this reason, the tracker was attached to the back of the top hand (Figure B.2 left). This choice guarantees good stability and visibility of the tracker without impeding the user's comfort.

Likewise, the alignment of the virtual victim body with the CPR mannequin from Brayden BraydenManikin (2019) was enforced through the same type of tracker. Initial tests revealed that the cardiac massage performance would transmit oscillations to the tracker when attached directly to the mannequin. Hence it was decided to attach both the mannequin and the tracker to a wooden plate to prevent this issue (Figure B.1).

The main idea of these initial design decisions is that a simple non-instrumented mannequin is sufficient to infer the user massage performance. The remaining requirement for the mannequin is to offer a similar resistance/deformation and shape as a real human for CPR training. The information of location, amplitude, and frequency of the massage can then be deduced from the tracker data as detailed now in subsection B.3.1.

B.3.1 Data acquisition

Our use of the HTC Vive trackers is compatible with their latency of 22ms (sampling frequency of roughly 45Hz) as measured by Niehorster et al. (2017). This paper also reported their relatively good accuracy and precision without occlusions for static positions, at least for our quite small and well-located interaction area, preventing tracking loss Niehorster et al. (2017) and allowing us to obtain hands height in the referential of the mannequin (Figure B.3).

Frequency

Once the heart stops beating, the blood circulation ceases immediately. If we consider that the heartbeat of a healthy person is near 60bpm, the expected heartbeat during a CPR massage is around 120bpm (i.e., 2 Hz) to compensate for the fact that it is externally induced. So it is mandatory to provide the user an accurate and stable frequency feedback for proper training.

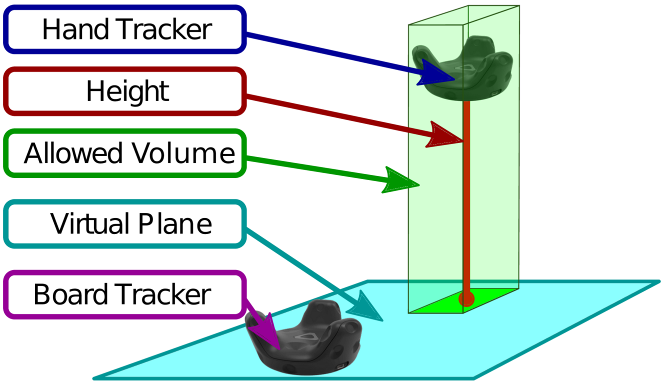
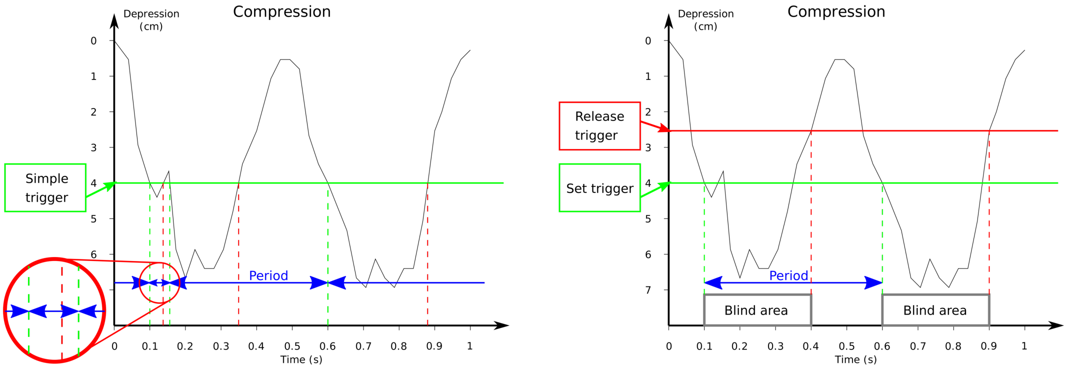


Figure B.3 – *Hand height and allowed interaction area are deduced from the tracker's data*

Despite the good aforementioned tracker characteristics, we nevertheless observed some tracker measurement variations when the cardiac massage was changing direction (actual measurements are provided in subsection B.3.2). We explain these artifacts by the movement dynamics that may induce some wobbles to the tracker through the hand tracker fixation (Figure B.2 left). The immediate consequence is that using a single threshold on the hand height signal is not appropriate for robustly counting the massage periods. This is illustrated on the conceptual drawing of Figure B.4a where each successive pair of green-red vertical lines delimit a trigger to compute a beat. To fix this issue, we applied a hysteresis filter with two fixed thresholds (Figure B.4b): one used to set the trigger and the other one for the release. Thus, to start counting a period, the height signal has to fall below a low triggering value first. Conversely, the period end is reached when the next trigger is reached.



(a) *Hand height data artifacts induce false positive beat detection when using a single threshold*

(b) *A two-threshold hysteresis approach makes the massage frequency measurement robust to reasonable artifacts induced by the movement dynamics*

Figure B.4 – *Conceptual illustration of the frequency count without (a) and with (b) a hysteresis filter*

Amplitude

This information requires a short user-specific calibration stage to define the highest hand location while in contact with the virtual victim's torso, as each user may have a different hand thickness. The resulting hand height defines the zero of the depression signal plotted in black in Figure B.4. The amplitude is computed by subtracting the current hand height from the calibrated zero height and retaining only positive values.

B.3.2 Hand tracking validation

In order to ensure that the hand height data acquired using the HTC Vive trackers match the actual torso compression depth, we conducted a validation study to compare our tracker-based measurement with the output of the mannequin internal depth sensor. For this, we connected the output of the integrated depth sensor to an external microcontroller. In our case, we used a simple atMega328 MicroChip (2018) connected on a development board (an Arduino Uno) to use the integrated power section, resonator, and USB $\iff$ Serial adapter Arduino (2019). The integrated depth sensor is an analogical Time-Of-Flight sensor: it is a Laser device composed of an emitter and a receiver. It measures the time required for the light to achieve the way out, way back between the sensor and the inner part of the mannequin torso. As the datasheet of this sensor does not give the manufacturer tolerance, we had to perform a test to calibrate our ground truth (Figure B.5a).

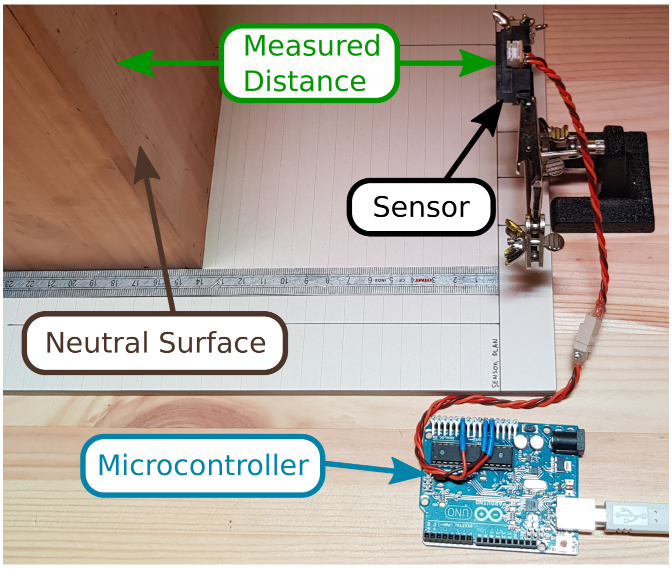
In order to link the analog value and the distance measured by the sensor, we measured several times the equivalence Distance $\iff$ Analog value on the whole range of distance allowed by the mannequin, computed point means, and fit a second-order polynomial using scipy.optimize.curve_fit. We obtained

$$h = \left(1.708 \cdot 10^{-3}\right) \cdot v^2 - 1.373 \cdot v + 346.8$$

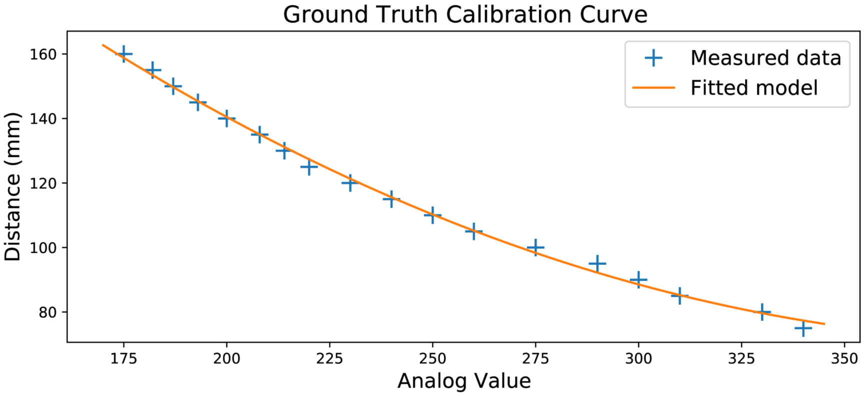
where v is the analog value and h the associated distance in millimeters. The regression plot is illustrated in Figure B.5b. As the sensor is fixed on the bottom of the mannequin h directly represents the height of the inner part of the torso. The compression depth induced by the hand movement is then $d = r - h$, with r the rest position height described in Figure B.3.1.

Tracker data evaluation setup

We programmed the microcontroller to output the raw computed height through the serial port clocked at 115200 bauds in a continuous stream used as an input for the Unity3D application. As the data stream from the microcontroller and the Unity3D application are not synchronized and have different refresh rates, we run the algorithm handling sensor values in a separate thread to avoid completely filling the Serial port buffer or introducing time mismatching (Figure B.6). The raw data are then exported into two separate files, but with the same time reference shared



(a) *Calibration bench used to calibrate the integrated mannequin depth sensor used as ground truth*



(b) *The calibration curve of the Time Of Flight Sensor. Analog values refer to the image of the voltage with the linear mapping* $0 = 0V$ *and* $1023 = 5V$.

Figure B.5 – *Calibration of the internal probe covering the whole set of possible compression for this mannequin*

between the two output files.

These data from this sensor are only used to assess our setup and are just stored for post-analysis.

As we can see in Figure B.7b, we configured a refresh rate for the integrated sensor much higher ($\sim$ 887 Hz) than the one we have from the HTC Vive tracker ($\sim$ 40 Hz). As both samples have a different refresh rate and as a short delay still exists between the two samples, pre-processing had to be applied to these raw data. To temporally re-synchronize both samples, we started by under-sampling the sensor data and extracted a short sub-sample of 5s from the tracker and sensor data at the beginning of the sample. Then we computed the cross-correlation between these two curves and kept the maximum point defining the best number of frames to shift to get the best

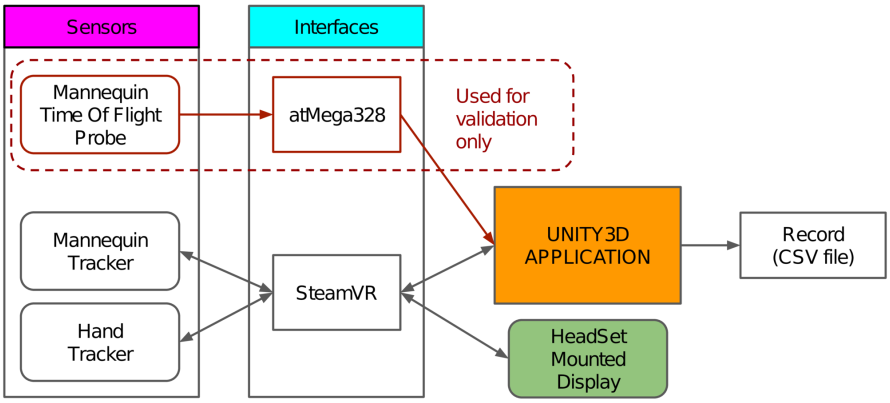


Figure B.6 – *System architecture*

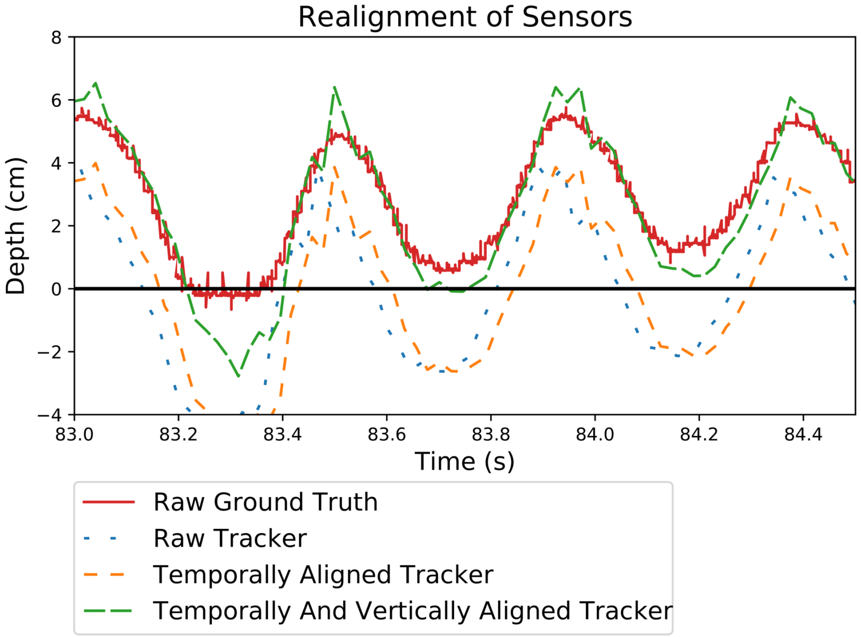
alignment (corresponds to the orange dashed curve from Figure B.7a). A second pass removes the vertical offset by subtracting the difference of the means over the sub-sample (in dashed green Figure B.7a).

Then we went back to raw inputs and, as the values from the internal probe are over-sampled compared to the tracker values, they are uniformly averaged for a second time, but this time around, each synchronized tracker sensor values timestamp to smooth the curve and get a sample with only one-time grid reference shared by both sensors curves (in orange on Figure B.7b). For a fair comparison, the tracker value is capped to zero (dashed green Figure B.7b) when hands are above the mannequin surface, as the internal probe cannot measure their position in such a context. As we measured the same frequency information with both sensors, we were able to slice samples into single "pushes" sub-samples using local lower extrema from the sensor probe as a "push" delimitation (vertical orange lines in Figure B.7b). Then we computed each push amplitude from both sensors and obtained an average absolute error of 1.20 cm (i.e., the tracker measures an amplified movement) with a standard deviation of 0.60 cm based on a dataset of 1758 pushes. This represents a mean ratio Tracker Amplitude over Sensor Amplitude of 1.30 with a standard deviation of 0.18.

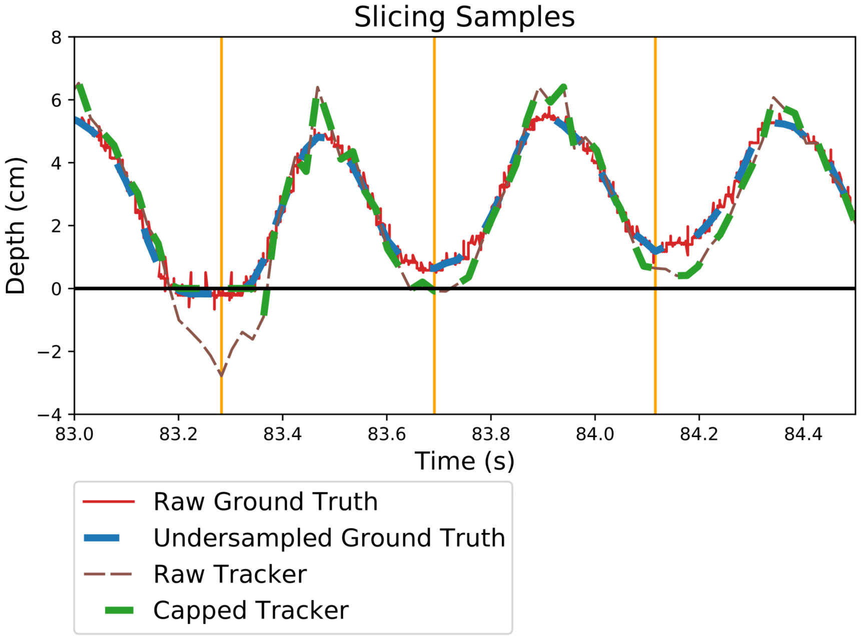
A noticeable artifact visible on the tracking curves (Figure B.8a) is the presence of spikes, especially when the direction of the movement changes as previously described in Figure B.3.1. Indeed, spikes are visible on the tracker curves (in orange) when the direction of the movement changes, whereas the reference value (in blue) does not present this issue.

This might be explained by the fact that when we wear the tracker, it is fixed with a slightly deformable strap, and when we change the direction of the movement, we can see that the tracker moves due to its inertia.

The difference between the maximum value read from the reference and the tracker for each push is plotted in red in Figure B.8b. Likewise, the difference between the minima of both sensors is plotted in green, and the filling areas represent the standard deviation. As we can see, both



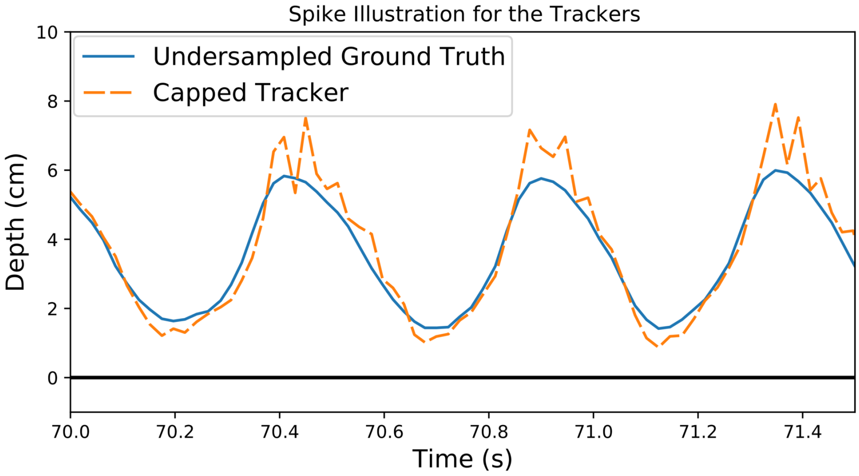
(a) *Realignment process. The black horizontal line represents the height when the mannequin is not pressed.*



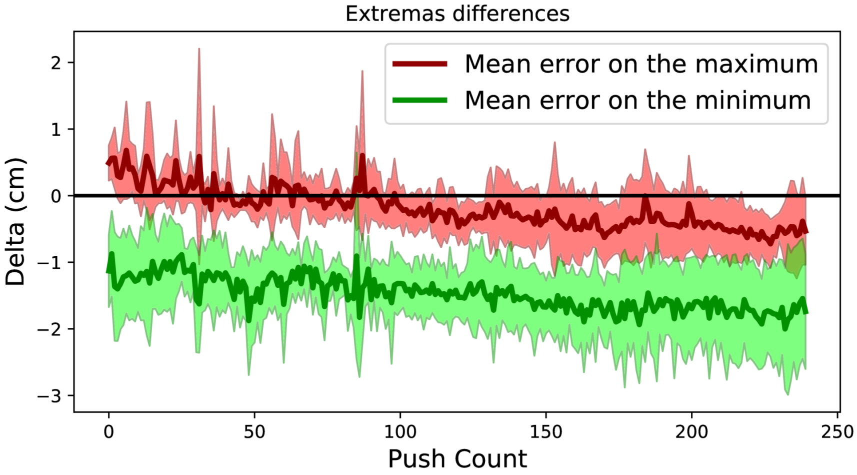
(b) *Under-sampling and splitting process. The black horizontal line represents the height when the mannequin is not pressed.*

Figure B.7 – *Illustration of the pre-processing applied on the raw data from the analysis of the results of the experiment*

curves follow a similar trend giving hints about a drift occurring over the number of pushes; the tracker erroneously reports getting closer to the ground as the number of pushes increases. These drifts led us to change the computation of frequencies for the offline analysis using the extrema delimitation described in Figure B.3.2 as it is more robust than the hysteresis method (Figure B.4) used to display the online feedback.



(a) *Spikes*



(b) *Extrema drifts between mannequin integrated probe values and tracker data on seven sessions*

Figure B.8 – *Illustration of artifacts on the tracker curve*

B.3.3 Visualization

We use Unity3D for integrating the components of our CPR training system (Figure B.6). Our system offers two visualization choices depending on the training focus on either emphasizing the sense of presence or finely guiding the CPR skill acquisition process. The minimal scene consists of the 3D environment, including at least the virtual victim's body and the user-tracked virtual hand. An additional display of performance indicators for the frequency and the amplitude can be generated in real-time within the first PV (Figure B.2 right).

Motivation for computing the virtual victim torso deformation

We consider that displaying a moving user's virtual hand during the massage is critical for inducing a strong agency. Indeed seeing such a movement is the main information the user has about the massage process; for this reason, we chose to display the user's virtual hand colocated with the actual hand location. An immediate consequence is the necessity to deform the virtual victim's torso accordingly to prevent the virtual hand to sink-in into a rigid virtual torso, thus potentially creating a break in presence Burns et al. (2006).

Torso deformation

The amplitude signal is used to drive the torso deformation as it faithfully expresses the compression depth achieved by the user on the virtual torso. We retained a purely geometric approach for the compression as the searched key effect is to prevent interpenetration rather than computing a physically-realistic shape deformation (Figure B.9). The full deformation reflecting the current amplitude is only applied to the torso mesh center.

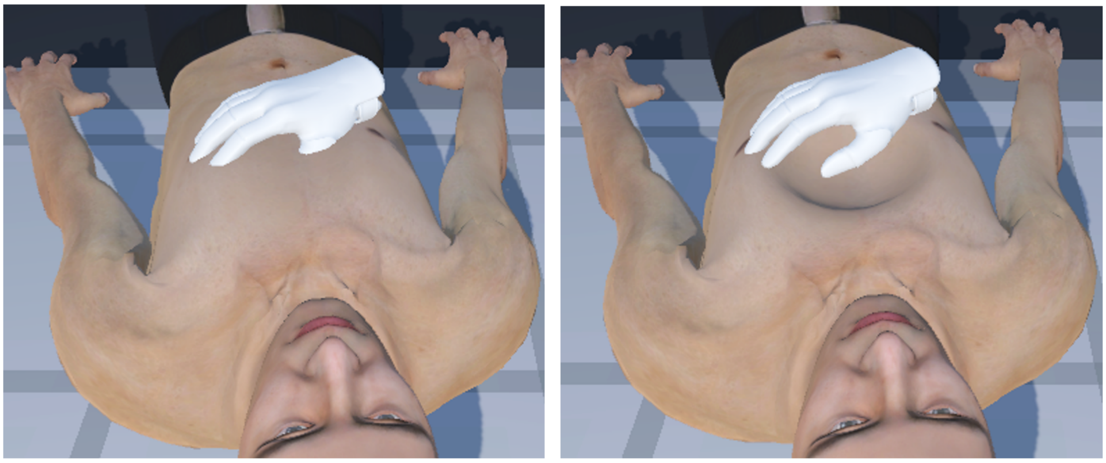


Figure B.9 – *Hand-torso interaction without (left) and with the simplified geometric deformation (right). Note that the displayed hand is on top of the (undisplayed) other hand for performing the cardiac massage. See the video for the first PV.*

Performance indicators

The cardiac massage amplitude and frequency can be displayed on-the-fly with individual gauges fixed above the virtual victim, as visible in Figure B.2 right. The optimal values are centered for gauges and colored green. Gauge markers are initialized at zero and are refreshed after each compression. If the user stops to perform CPR, a timer will automatically reset these markers to zero. Gauges indicate values between 100 and 140 compressions per minute and between 1 and 11 cm for the amplitude.

B.4 Experimental evaluation

B.4.1 Hypotheses

The purpose of this study is firstly to assess whether the minimal immersive setup that we have retained is sufficient to elicit a sufficient level of presence and Embodiment (through the agency score). We used the IPQ (I-group Presence Questionnaire) presence questionnaire Schwind et al. (2019) and the embodiment questionnaire, adapted from Gonzalez-Franco and Peck (2018), to match our specific haptic interaction context (available in appendix) to assess these hypotheses. Scores are then normalized by summing and dividing the result by the maximum score possible to ensure that normalized scores are within $[0,1]$.

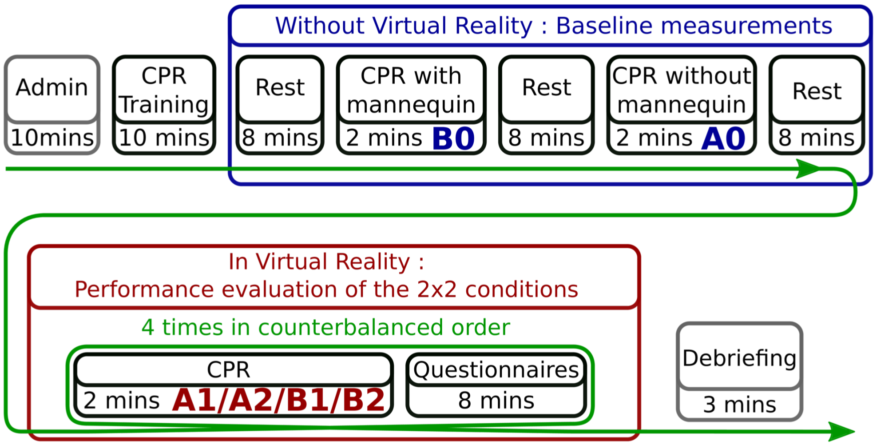
The second goal of the study is to determine the impact of two factors on the massage skill Performance (frequency and amplitude): Haptic feedback with a CPR mannequin (without and with) and CPR Performance feedback (without and with) in the HMD (Figure B.10b).

Thus we formulate the following hypotheses :
H1 - We expect a normalized presence score to be ≥ 0.5
H2 - We expect a normalized agency score to be ≥ 0.5
H3 - The use of the mannequin leads to better performance than without the mannequin because the haptic interaction induced by the mechanical property of the mannequin sustains more the regularity of the movement compared to a movement in free-space
H4 - The combination of the mannequin use with the real-time performance indicators leads to a better performance compared to no performance indicators
H5 - Displaying the performance within the HMD reduces the level of presence compared to the context without the performance indicators

B.4.2 Method

Given the potentially wide differences among subjects in terms of initial expertise in CPR massage, we decided to first provide a training session to all of them with the CPR mannequin in the regular context of such training, i.e., without VR. We then measured the performance (amplitude and frequency) in two successive contexts, with and without the mannequin, to establish their baseline (Figure B.10a top line).

After establishing the two baselines (**A0** and **B0**), the subjects enter a sequence of four CPR massage sessions in VR, each consisting of a two minutes massage (corresponding to **A1**, **A2**,



(a) *Experimental protocol*

Haptic Feedback (Mannequin) / Visual Feedback	Without	With
Baseline (no VR)		
No live performance display	**A0**	**B0**
Trial (in VR)		
No live performance display	**A1**	**B1**
Live performance displayed	**A2**	**B2**

(b) *Experimental evaluation with two factors: Haptic feedback (without and with the physical mannequin) and Performance feedback (without and with the amplitude and frequency gauges)*

Figure B.10 – *Experimental main timeline and conditions*

B1 or **B2** from Figure B.10b) followed by 8 minutes rest (Figure B.10a bottom line). Massage order is counterbalanced in order to prevent bias from a potential training effect. Each rest period is dedicated to the filling of questionnaires described previously.

B.5 Results

A total of twelve subjects participated in the experimental evaluation (age within [16,56], median 27.5, four female). One subject sample was corrupted, and one female subject stopped the experiment due to insufficient physical force to interact with the CPR mannequin. All subjects were right-handed, and six reported being familiar with VR while only two indicated an average experience of CPR massage. The experience is part of a project approved by the Swiss National Science Foundation, and subjects signed a consent form and received CHF20 as compensation for their time.

B.5.1 Presence and embodiment levels

We used non-parametric RankSum tests with a Bonferroni correction on $p-$values to determine whether the aforementioned hypotheses were verified.

Results of these tests are plotted in Figure B.11a for the assessment of the haptic feedback, Figure B.11b for the agency (both based on Gonzalez-Franco and Peck (2018)), and Figure B.11c for the three presence components from the IPQ Schwind et al. (2019).

As expected, significant differences found for haptic scores (**A1** - **B1**, $p = 0.025$ and **A2** - **B2**, $p = 0.004$) show the positive impact of the physical mannequin on haptic feedbacks perceived by participants. Surprisingly, haptic scores in non-mannequin conditions are higher than expected. We suspect that a combination of multiple factors, including the deformable avatar and the self hands contact, might explain such scores.

B.5.2 Performance quality

Concerning the evaluation of the performance: the Shapiro test rejected the hypothesis of the normality of samples across all conditions from the dataset (cf. Appendix). Thus we also applied non-parametric tests (RankSum) with corrections for $p-$values (Bonferroni). Results are displayed in Figure B.12.

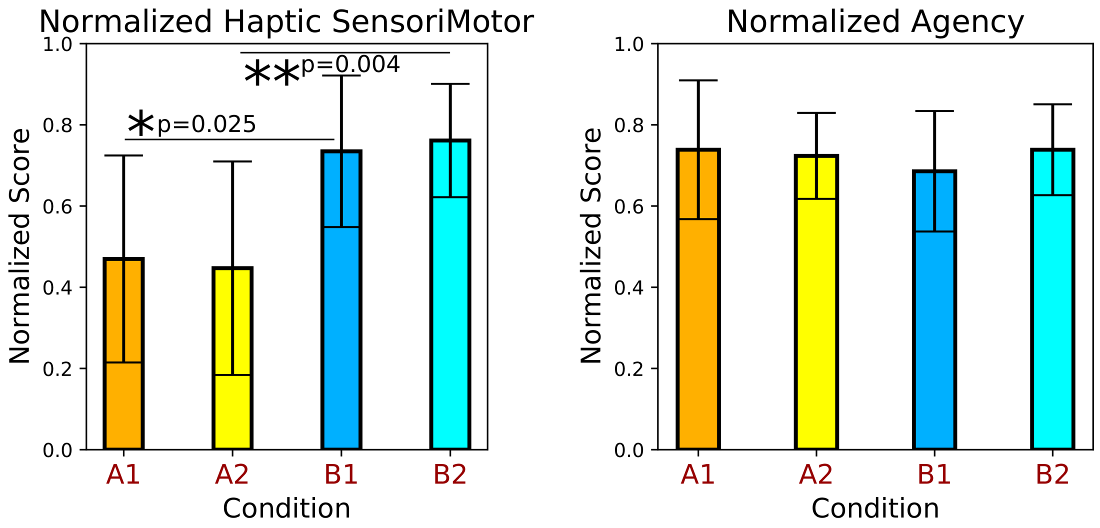
In all conditions without the live displayed performances, we observed a significant difference between with and without the tangible mannequin (**A0** - **B0**, $p = 0.0039$ and **A1** - **B1**, $p = 0.0126$). Conversely, when live performance is displayed, we did not observe a significant difference (i.e. **A2** - **B2**). This suggests that whenever guided with the displayed performances, the performed frequency is closer to the target value.

For the amplitude, the difference observed (**A1** - **B1**, $p = 0.0019$ and **A2** - **B2**, $p = 0.0003$) shows that even guided, used still struggles to reach the correct amplitude without the help of the mannequin.

B.6 Discussion

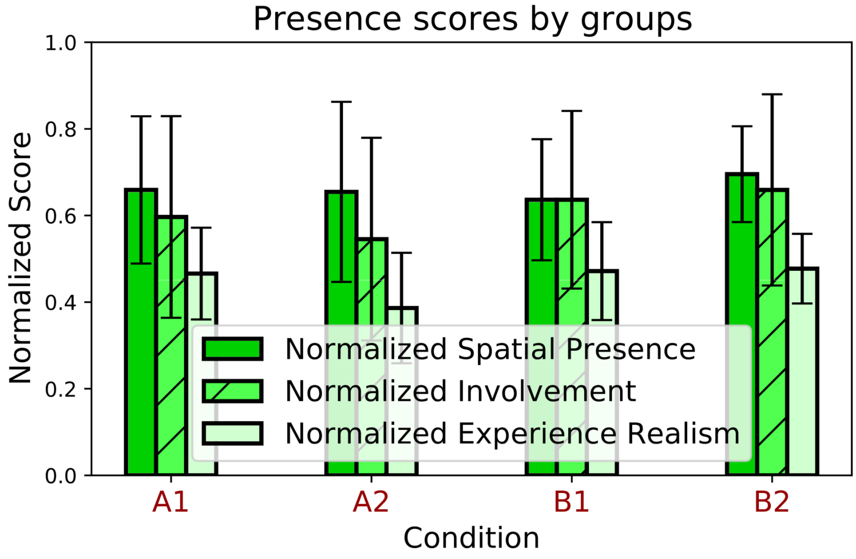
Regarding the experimental evaluation with subjects, our hypotheses were partially confirmed as follows:

H1 - The presence level delivered by the IPQ questionnaire is decomposed into three components: experience realism, spatial presence, and involvement. Only the scores of spatial presence and involvement succeed to be in the upper half (i.e., ≥ 0.5) of the presence scale hence offering only a partial confirmation of H1. Indeed, the proof of concept design was far from being realistic (e.g., single-hand display).



(a) *Haptic SensoriMotor scores*

(b) *Agency scores*

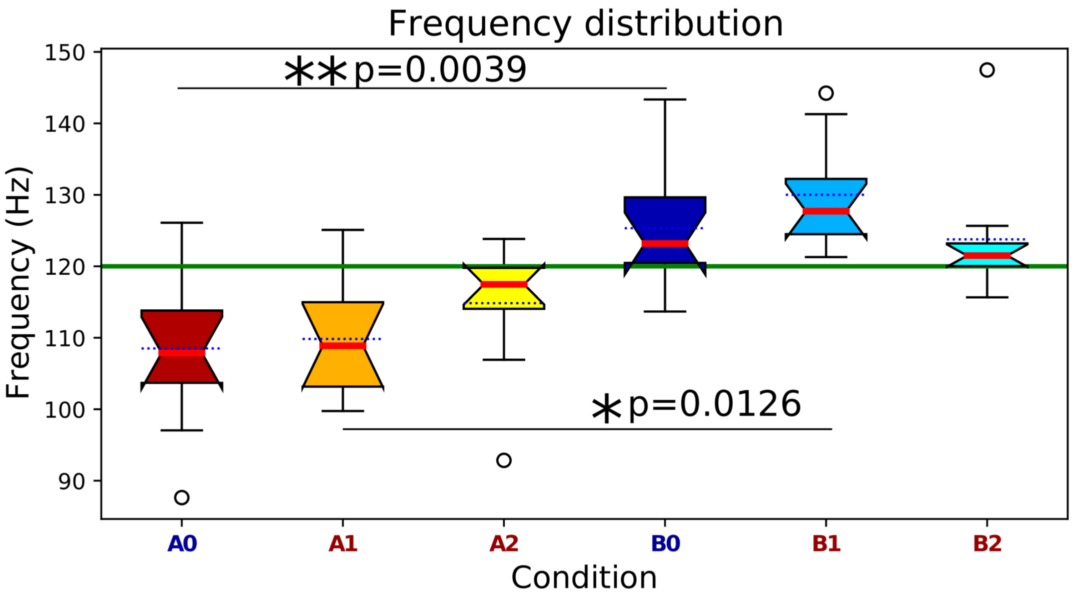


(c) *Presence scores*

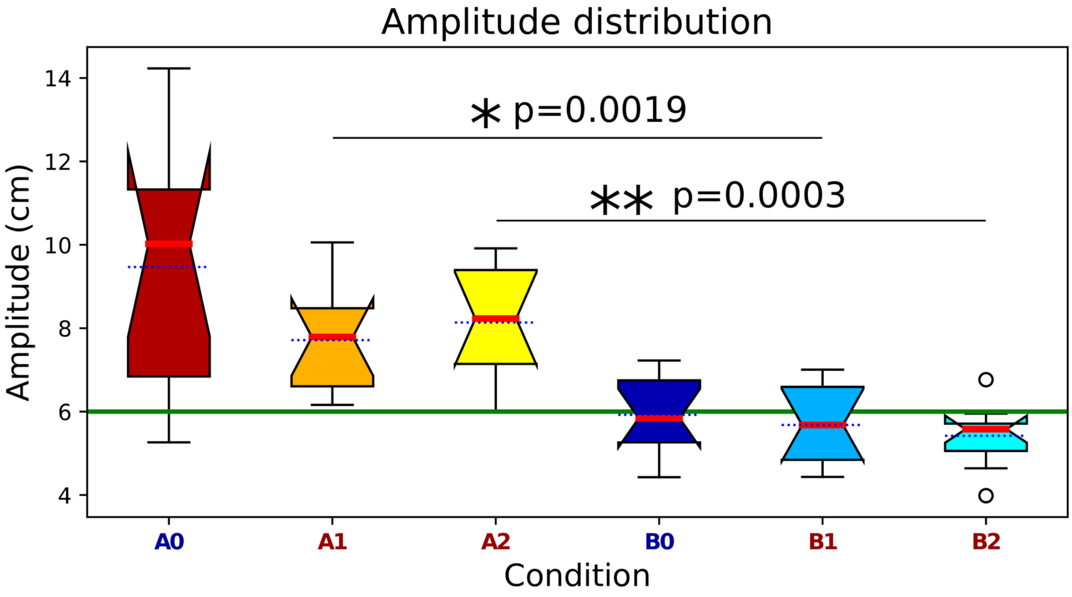
Figure B.11 – *Distributions of normalized scores from questionnaires (labels from Figure B.10b)*

H2 - As this single-hand representation did not penalize the embodiment score through its agency component, showing a consistently high level across all conditions, this hypothesis is accepted.

H3 - This hypothesis was validated as the integration of the real CPR mannequin in VR has a significantly positive impact on the massage performance quality compared to the conditions without mannequin (cf Figure B.12). Indeed, in a real-world context, the weight of the user combined with the resistance of the victim's torso constitute a dynamic system where the user only needs to give downward impulses to perform CPR. Additionally, this system links frequency



(a) *Frequency distribution of the result by groups*



(b) *Amplitude distribution of the results by groups*

Figure B.12 – *Plot of scores obtained by group (labels from Figure B.10b). Horizontal green lines represent the targeted value for the best CPR, and noches represent the 95% confidence interval of the median (the red line, dashed blue one represents the mean).*

and amplitude; thus, given the right frequency, it is easier to achieve the right amplitude as the former is more easily mastered. In the non-mannequin conditions, there is no such dynamic coupling; thus, after each impulse, users have to use their back muscles to lift up. Also, the amplitude/frequency link is different, explaining the observed wrong amplitude range while the frequency was correct. Thus even in the presence of the live displayed performance factor, it remains difficult to reach both targets (amplitude and frequency) at the same time.

Finally, no significant differences were found between the performance baseline **B0** and the mannequin conditions (**B1** and **B2**). This should not be interpreted negatively, though; simply put, one can see that performances were as good without and with VR, hence VR does not degrade performance. On the other hand, the presence scores show that VR has some potential to immerse the participant in a situation much closer to real life (but this was out of the scope of the present submission).

H4 - This hypothesis was initially hinted from the outcome of prior works from Semeraro et al. (2013) where authors noticed a significant difference between CPR performance with and without performance feedback. Surprisingly it was not validated in the mannequin condition as the performance display in the HMD was not bringing any added value to the performance quality.

H5 - Only the frequency performance quality shows a correct target value in the no-haptic condition, which suggests this partial benefit from the displayed information. Likewise, no significant reduction of presence level can be linked to the display of the massage information in the user field of view; this invalidates this hypothesis.

B.7 Conclusion

As this study is a proof of concept, many scenario events were not implemented (e.g., warnings if the hand is badly placed). Moreover, we only secured a minimal embodiment level with a single rigid hand representation rather than dealing with whole-body integration. Furthermore, performing CPR remains a physically demanding task, a point this study was not designed to address.

Our results show that the proposed minimal setup with single-hand tracking is sufficient to provide accurate frequency feedback. Unlike Semeraro et al. (2019), our results also show that the raw amplitude from the tracker needs to be scaled down by 0.77 (in our setup) to achieve a sufficient level of fidelity for the amplitude measurement. Thus, if we take this into account, the measurement of both performance criteria of a CPR massage can be done with a low-cost non-instrumented mannequin that offers the standardized haptic response.

As expected, this experiment clearly shows that the presence of a tangible mannequin provides better haptic feedback than without.

Finally, unlike our observation that a tangible mannequin is not necessary for training the correct frequency, this study strongly implies that a mannequin is required to train the correct amplitude range for a CPR massage.

B.8 Acknowledgement

This work was supported by the Swiss National Science Foundation, “Immersive Embodied Interactions in Virtual Environments” N°200020_178790. Special thanks to Mr. V. LEMAIRE from the company *Be! Rescuer* for his precious experience and expertise in resuscitation.

C How errors involving a finger swap in finger animation are perceived?

C.1 Demographic questionnaire

Question	Possible answers
Gender	Male Female Other I don't want to answer
Age	Integer
Height	Integer (centimeters)
Weight	Integer (kilograms)
Handedness	Right handed Left handed Ambidextrous
Main occupation	Text
Have you ever experienced "virtual reality" before?	Linear scale between 0 (No experience) and 7 (Daily use)
Do you practice sport?	Linear scale between 0 (No) and 7 (Daily practice)
Do you often type (piano/keyboards/etc.)?	Linear scale between 0 (No) and 7 (A lot)

Inverse Kinematics

Both arms and fingers IKs relies on the same principle: with known bones lengths (through a calibration process) and mechanical constraints (single joint for the elbow, the same flexion rotation for both proximal-intermediate and intermediate-distal joints Aristidou (2018)), we can determine joints rotations in the plan (the elbow or finger joints can only bend alongside a single axis). The final hand animation step is to realign the fingertips with their associated marker as described in chapter 2. This additional rotation follows a predefined curve based on the lateral position of the hand was added alongside the elbow-wrist axis (the elbow is not constantly stuck near the ribs when the target is within range). Finally, half of the swivel angle of the wrist is applied to the elbow joint to avoid mesh rigging issues.

Finger swap animation

When a swap is introduced, only the vertical motion of the *real source finger* (the finger the subject moves) is redirected onto the *displayed destination finger* (the finger the subject sees moving) to avoid potential lateral interpenetration.

At the beginning of the swap, the fingers' markers' positions are stored. Then, the lateral motion of the *displayed source finger*'s (i.e., what the users see for the source finger) motion is progressively shifted back to its initial position while the vertical motions used to animate both displayed fingers are progressively permuted. A custom anatomical correction is applied to scale the measured vertical movement to the size of the moving finger (e.g., when swapping the middle finger with the little finger).

The progressive removal of the lateral motion prevents the user from visually spotting the source finger while the swap is enabled and the lateral motion of the destination finger is not altered. Thus, if the subject moves the *real destination finger* (i.e., the finger initially not supposed to be moved by the user but still animated by the swap) laterally while the swap is active, the *displayed destination finger* will also move laterally, but its vertical motion will be the one of the *real source finger*. Conversely, if the user moves the *real source finger* laterally, nothing will move on screens, and if it moves vertically only the *displayed destination finger* will move.

The displayed fingers' positions are computed using Equation C.1 with :

- t refers to the rate of swap introduced and varies from 0 (motions are still fully congruent) to 1 (motions are fully permuted) with a step set to 0.4 per frame ($\simeq 31ms$) for the activation and 0.02 for the release ($\simeq 625ms$).

- $\vec{P}_{\text{xz}_{\text{ref}}}$ is the planar position of the *real source finger* tip when the swap is triggered

- $\vec{P}_{\text{xz}}$ is the planar position of the *real source finger* tip and $\vec{P}_{\text{xz}_d}$ the one for the *displayed source finger*

- h_{src} is the vertical component of the *real source finger* tip position and h_{dst} the vertical component of the *real destination finger* tip, with h_{src_d} and h_{dst_d} the *displayed* ones respectively

$$\begin{cases} \vec{P}_{\mathrm{xz}_d} & = & (1-t)\cdot \vec{P}_{\mathrm{xz}} + t \cdot \vec{P}_{\mathrm{xz}_{\mathrm{ref}}} \\ h_{\mathrm{src}_d} & = & (1-t)\cdot h_{\mathrm{src}} + t \cdot h_{\mathrm{dst}} \\ h_{\mathrm{dst}_d} & = & (1-t)\cdot h_{\mathrm{dst}} + t \cdot h_{\mathrm{src}} \end{cases} \tag{C.1}$$

The swap ends when the moving finger is placed back on the table or when the current button leaves the activation area. Such a method does not introduce any additional delay in the fingers' animation when a swap is introduced.

D Does scaling player size skew one's ability to correctly evaluate object sizes in a virtual environment?

Full reference:
Hartman, N., Delahaye, M., Decroix, H., Herbelin, B., & Boulic, R. (2020, October). Does scaling player size skew one's ability to correctly evaluate object sizes in a virtual environment?. In Proceedings of the 13th ACM SIGGRAPH Conference on Motion, Interaction and Games (pp. 1-6) https://dl.acm.org/doi/10.1145/3424636.3426908.

Does scaling player size skew one's ability to correctly evaluate object sizes in a virtual environment?

Neal HARTMAN*
neal.hartman@cern.ch
CERN
Switzerland

Mathias DELAHAYE*
mathias.delahaye@epfl.ch
EPFL
Switzerland

Hugo DECROIX
hugo.decroix@epfl.ch
EPFL
Switzerland

Bruno HERBELIN
bruno.herbelin@epfl.ch
EPFL
Switzerland

Ronan BOULIC
ronan.boulic@epfl.ch
EPFL
Switzerland

ABSTRACT

This study attempts to evaluate whether a navigation technique based on scaling the user's avatar impacts the user's ability to correctly assess the size of virtual objects in a virtual environment. This study was realized during the CERN Open Days with data from 177 participants over eighteen years old. We were able to observe well-established phenomena such as the effect of inter-pupillary distance (IPD) on perception of scale, as well as original results associated with scaling factor and avatar embodiment. We observed that the user is more prone to overestimate object sizes from the Virtual Environment (VE) when provided with an avatar, while scaling the IPD according to the scale of the user's avatar contributes to a reduction in the overestimation of object sizes within the VE.

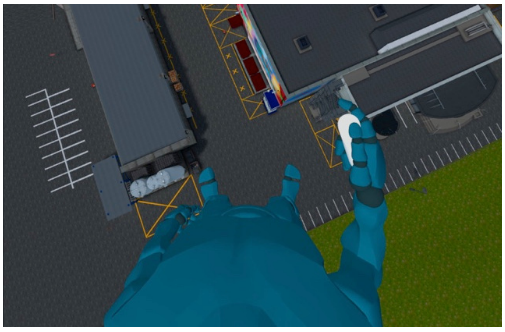


Figure 1: Illustration of locomotion in "Giant Mode" while navigating around the a virtual representation of CERN's main site

CCS CONCEPTS

• **Human-centered computing** → **User studies**; **Virtual reality**; **Interaction techniques**.

KEYWORDS

Virtual Reality, Embodiment, Avatar Scaling, User Interaction, Navigation

ACM Reference Format:
Neal HARTMAN, Mathias DELAHAYE, Hugo DECROIX, Bruno HERBELIN, and Ronan BOULIC. 2020. Does scaling player size skew one's ability to correctly evaluate object sizes in a virtual environment?. In *Motion, Interaction and Games (MIG '20), October 16–18, 2020, Virtual Event, SC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3424636.3426908

1 INTRODUCTION

Virtual Reality (VR) experiences often demand that players navigate in order to discover and interact with the environment. However, current technical limits prevent players from navigating over long distances while maintaining a link between the real, physical movement performed and the one displayed in the headset. Different techniques have attempted to bridge the gap between the physical movement of the user and the locomotion of the player within the VE, from treadmills [Fung et al. 2006], rolling spheres, and sliding on a curved floor with slippers [Warren and Bowman 2017] to traditional techniques like navigating with joysticks, as in traditional video games, or teleportation [Bozgeyikli et al. 2016]. But all of these techniques have serious drawbacks. Solutions that require additional hardware limit the potential number of users who can experience them, and teleportation does not allow continuous movement, which can be fatiguing and disorienting when traveling over long virtual distances.

Several researchers and developers have recently begun to experiment with a novel navigation technique which attempts to mitigate these disadvantages. This technique involves allowing the user to change scale, or euphemistically, to become a Giant, in order to be able to continuously traverse more distance with less disruption. One such early system, named GulliVR [Krekhov et al. 2018] is used as the basis for this paper. While GulliVR used the "Giant mode" technique, it did not employ an avatar, which many subsequent studies have done, including a hybrid approach by the same authors which combines a Giant avatar with teleportation [Cmentowski et al. 2019].

---

*Both authors contributed equally to the paper

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MIG '20, October 16–18, 2020, Virtual Event, SC, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8171-0/20/10…$15.00
https://doi.org/10.1145/3424636.3426908

Other researchers have also found that the "Giant Mode" navigation technique is particularly promising. [Abtahi et al. 2019] examined three means of boosting a user's navigation speed: speed gain, walking as a Giant, and scaling up the user (as a Giant) while maintaining the head at the ground level; users preferred the simple Giant mode. A variant of walking as a Giant involves flying like Superman, which puts the user's eyeline at the same level, but with somewhat different characteristics - i.e. no Giant avatar and an inter-pupilary distance (IPD) that would be associated with a human. This technique, examined in [Piumsomboon et al. 2018], found that scaling IPD when changing a user's eye-height more strongly modifies the user's perception of scale of the environment; effectively, to be a Giant, one must see like a Giant. The researchers involved did not use an avatar in their study, but did highlight the value it would have in giving natural cues to the user during navigation.

With the advent of new, low-cost devices, VR experiences are now accessible to a broader public than ever before, and with less hardware investment, making the technique of navigating like a Giant attractive to an increasing number of users.

Due to the perceptual manipulations necessary to use this technique, however, we decided to study whether it would impact the user's judgement of the scale in the VE compared to the typical scale distortion observed in VEs in general.

2 RELATED WORK

2.1 Scale in a Virtual Environment

Perception in virtual environments, and specifically judgement of distance and scale within those environments, has been studied for decades. It has long been observed that a user's judgement of scale is impaired when using a virtual environment. While there is little commonality in the numerous experiments that have evaluated this effect, the overwhelming conclusion is that users underestimate distance (and by extension scale) in a virtual environment relative to the real world. The magnitude of this effect varies heavily with experimental conditions, technology used to view the virtual environment (type of device, field of view, binocularity, resolution), and the details of the virtual environment (image quality, richness, texture, lighting, real-world visual cues, and experience). The underestimation of scale ranges from a factor of 2, as observed through egocentric observations of distance to an object by untrained observers [Knapp 1999], to roughly 20%, as determined by a study of architects that were given virtual or real tours of a museum, and then asked to evaluate distances (height of a ceiling, length of a wall) [Henry and Furness 1993].

2.2 The Body (or Avatar) as a Scale Reference

It has been shown that the body acts as a scale reference for the outside world, both in near personal space and beyond [Van Der Hoort et al. 2011]. In the "Being Barbie…" study, the user is attributed, through a video-relay system, the body of a mannequin ranging in size from 30cm to 4m. The user is asked to judge both distance and object scale, and the results show that the user mis-estimates scale based on the body that is attributed. The smallest body results in an over-estimation of real-world scale, while the largest body results in an under-estimation. The results vary from approximately a factor of 2 mis-estimation in the small-body case, to a factor of about 1.5 in the large-body case. Interestingly, even the normal case results in an under-estimation of scale (though slightly lower than the amount measured in the large-body case), echoing other studies that have shown that it is not the qualities of the environment that affect this error, but the intrinsic nature of representation through a virtual display.

2.3 Vision Characteristics as a Scale Reference (Superman)

The roles of eye height and inter-pupillary distance (IPD) have also been studied in the literature in the context of estimation of scale of a virtual environment [Kim and Interrante 2017]. In a simulation with no bodily representation (i.e. no avatar) there are two possibilities for assuming an eye level above the normally-expected human height: either flying (if the IPD is conserved at normal human size, i.e. Superman) or growing (if the IPD scales with the position above ground level, i.e. Giant). These two conditions are compared in the study "Superman vs. Giant," [Piumsomboon et al. 2018] where it is found that, in the mode without IPD manipulation (Superman) half of the users felt that their body size was larger (even though no body is visible) and half thought that it was normal, yet flying. Conversely, in the mode with IPD manipulation, users judged themselves to have a Giant body more than 90% of the time [Piumsomboon et al. 2018].

In examining whether this estimation of body size (even though no body was visible) had an effect on the estimation of environment sizes, it was found that indeed the manipulation of IPD led to a skew in the estimation of scale; smaller IPD resulting in over-estimation, and larger IPD resulting in under-estimation, as is congruent with the previously discussed experiments. This effect has also been found in other studies where IPD and height have been manipulated both up and down, i.e. Dwarf vs. Giant, which show that indeed manipulation of height as well as IPD result in a change in ability to estimate scale. However, with relatively small changes in height (i.e. plus or minus 50cm), the effect of increasing eye height seems to have small effects, whereas decreasing it is more pronounced [Kim and Interrante 2017]. The opposite effect was found in a similar study which used the same 50cm offset, where the significance was found in reducing eye height and not increasing it [Leyrer et al. 2015]. This second study also analysed the role of an avatar in these two conditions, and it found no significant effect.

While these studies of eye height and IPD show some conflicting results, they all used relatively modest adjustments in IPD and height, as well as feature rich virtual environments, where many environmental size cues were available.

2.4 Summary and Contributions

Prior research shows one clear conclusion: a user's ability to judge the scale of an environment presented through virtual means can be manipulated through three key variables:

- The size of the user's inferred or visibly-attributed body (avatar)
- The user's eye height above ground level
- The user's IPD

While some elements, like magnitude of the manipulation, visual richness of the environment, placement of objects in the visual field, etc., may have an impact on magnitude and sensitivity of the results, it is almost universally observed that increasing the key variables listed above leads to an under-estimation of the scale of the virtual environment, whereas decreasing them has the opposite effect.

Our contributions hinge on assessing the impact of the "Giant mode" as a means of effective navigation. For this purpose we propose a system based on GulliVR [Krekhov et al. 2018]. This system allows the user to navigate in a VE in the direction of regard, which is intrinsically intuitive, by simply pressing a button on the hand controller. In order to move more quickly through the environment, and to obtain a reference point for the layout of the surroundings, a simple function allows the user to scale their avatar to Giant scale (and to move correspondingly faster through the environment, unimpeded by buildings and terrain). A controlled experiment evaluates user performance in a size assessment task within a between-group design including five combinations of the following three factors: avatar, Giant mode and IPD scaling.

3 EXPERIMENTAL DESIGN

3.1 Question

In order to make a technique like Giant navigation interesting, it must entertain scale shifts larger than those attempted in previous studies. Scale ratios of 20-40 have not been studied in depth. In addition, the effect of an avatar in such a large scale offset has not been studied, and by incorporating it as an additional variable, allows the comparison of five distinct groups. Lastly, such a navigation technique is naturally episodic and temporal. In this respect, there is no existing research which examines the effect of time, and changeability, on a user's estimation of scale in a virtual environment.

As such, the research question can be simply stated as: to what extent do the factors of avatar, eye height, IPD, and time have on a user's estimation of scale in a feature-rich, openly navigable virtual environment?

3.2 Hypotheses

Based on the literature, we expect to verify several well-supported hypotheses, as well as investigate new, untested ones related to time and avatar.

1. H1 - Users in the control groups (no Giant mode) will underestimate the scale of the VE as observed in general in virtual environments
2. H2 - Users in Giant modes will underestimate the scale of the VE in higher proportions than the control groups
3. H3 - Users with an avatar will show more under-estimation effect than those without an avatar
4. H4 - Users with larger IPD will show more under-estimation effect than those with normal IPD
5. H5 - Users who spend more time in Giant mode will show a stronger under-estimation effect

3.3 Groups

To study the impact of these factors, we divided the experiment into 5 groups as shown in Table 1. Time was studied as a fourth, pseudo-independent variable.

Table 1: List of the different groups used during the experiment

Group Name	Avatar	Giant Mode	IPD Scaling
Control	No	No	N/A
Avatar Control	Yes	No	N/A
Superman	No	Yes	No
Avatar Scale	Yes	Yes	Yes
Without Avatar Scale	No	Yes	Yes

3.4 Evaluation

In order to evaluate the accuracy of the perception of size within the virtual environment, we implemented several binary questions constructed as illustrated in Figure 3. The user is faced with a choice between a small figure and a large one. Neither response is correct, but one response is closer to the correct answer than the other. The user is informed of this fact at the outset, and is requested to make the choice that is closest to the correct answer. A response is considered correct when it corresponds to the figure that is closest to the true scale. When the contrary choice is made, it is considered incorrect, and the error direction is defined as either under or over, depending on which way the question is skewed. The list of the eight questions and their correct and incorrect answers, as well as error direction, is given in annex.

3.5 Experimental Protocol

The design of the VR experience allows users to navigate freely within a virtual environment and to decide (with the exception of the control groups) the amount of time that they spend in either normal or Giant scale. At the outset, the user is given a short tutorial, and for those in a group that allows scaling, is prompted to scale up and down once so that they are aware of this capability.

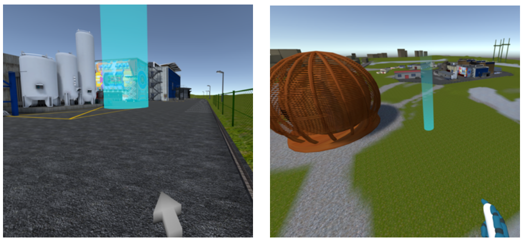
Once inside the VR experience, the user is guided by prompts, which invite the user to move through a series of checkpoints, each indicated by a directional arrow and a column of blue light, which is visible from either normal or Giant scales (Figure 2). Each checkpoint is associated with a virtual object in the VE.

Once at a checkpoint, the user is reduced to normal scale and oriented towards the object of interest at that checkpoint (or maintains normal scale in the control group) so that there is an equivalence of perspective between all experimental groups at this stage. The user is asked to judge the scale of the object in question, following the binary choice as previously described in Figure 3 according to subsection 3.4.

4 EXPERIMENTAL RESULTS

4.1 Results

We conducted our study at the CERN main site located in Meyrin, during the week-end of the CERN Open Days. During this period



(a) From the Normal Perspective (b) From the Giant Perspective

Figure 2: Navigation Markers

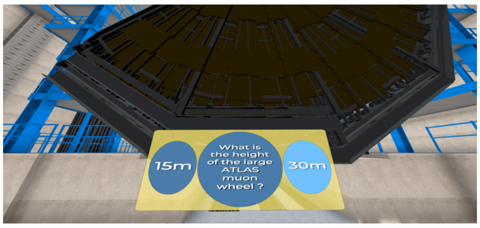


Figure 3: Scale question prompt, showing the binary choice between a small figure and a large one

we collected data from 342 participants, aged between 5 and 99, running twelve Oculus Go headsets in one room simultaneously.

4.1.1 Data preprocessing. In order to eliminate inconsistent data from the population, we applied three different filters. We removed results from participants under the age of 18, participants that did not complete the outdoor phase of the experience, and only considered responses and time spent at Giant scale for the outdoor part of the experiment, as the size scaling in indoor areas is non-uniform and depends on the height of the room's ceiling.

After filtering, the dataset contains 177 entries (96 male and 81 female).

Size accuracy was computed as the rate of correct answers over total questions. Given that each question had a designated "correct" answer and designated "error" (cf. annex) we were able to determine a normalized error direction for the user's size evaluation. A user that judges the correct object size for every question would receive an error direction of 0. Otherwise for each error we added +1 or -1 to the error trend depending on the direction of the error and then divided it by the number of errors.

As results are drawn from discrete values we only considered non parametric tests for the analysis.

Boxplot representations show the median (large red line in the middle of the notch), the mean (small dotted blue line), first and last quartile (colored area). Whiskers represent the contained population between $Q1 - 1.5 \cdot (Q3 - Q1)$ up to $Q3 - 1.5 \cdot (Q3 - Q1)$ while circles represent outliers (values beyond these whiskers). Finally, notches represent the 95% confidence interval of the median.

4.1.2 H1. To assess that the direction of the error of the scale estimation in control groups is lower than zero we started by aggregating data from the "Control" group with the "Avatar Control" group. Then we applied a Wilcoxon test to reject the null hypothesis that the evaluation error trend is not null. We obtained a p-value of 0.86 meaning that we cannot consider a significant difference in the obtained value from 0. Thus our data does not support the first hypothesis of the paper.

4.1.3 H2. In order to determine the potential effect of Giant mode we concatenated samples from the "Control" group with the "Avatar Control" group (the two without access to Giant mode) in one sample with the remaining groups in another sample. Then we compared both samples using the Ranksum test and obtained a p-value of 0.17. As as result we find that Giant mode alone is not sufficient to validate our second hypothesis.

4.1.4 H3. We used the same approach to check for a more pronounced scale underestimation for groups where subjects were provided with a virtual body (avatar). Thus we aggregated data from "Avatar Scale" and "Avatar Control" into one sample and the other groups in the comparative sample, obtaining a p-value of 0.040 (Ranksum) highlighting a significant difference between the two samples for the direction of the evaluation error. To retrieve which direction we performed a one-tailed Mann-Whitney U test giving us a p-value of 0.016, indicating that groups with an avatar presented a higher overestimation of virtual object sizes relative to non-avatar groups, soundly rejecting our third hypothesis. A plot of scores from these two samples are available in Figure 4.

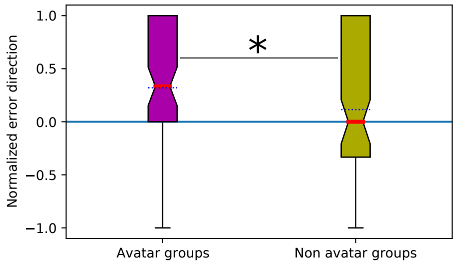


Figure 4: Comparison of normalized score of evaluation error direction between groups with an avatar and those without

4.1.5 H4. As above, to determine the effect from IPD on the underestimation of object size we concatenated data from "Avatar Scale" and "Without Avatar Control" groups into a first sample and remaining ones into a comparative sample. With a p-value of 0.010 for the Ranksum test we were able to highlight a significant difference between these samples. As before, the direction was then assessed using a one-tailed Mann-Whitney U test with a p-value of 0.004, showing that the group with IPD scaling (with or without avatar) tended to overestimate less the sizes of objects, thus validating our fourth hypothesis. These results can be seen in Figure 5.

4.1.6 H5. To test our final hypothesis we aggregated data from all groups where subjects experienced the Giant mode (i.e. all groups except "Control" and "Avatar Control"). This sample was split into two samples using the median of the time spent as Giant as a delimiter, which was calculated as approximately 60 seconds.

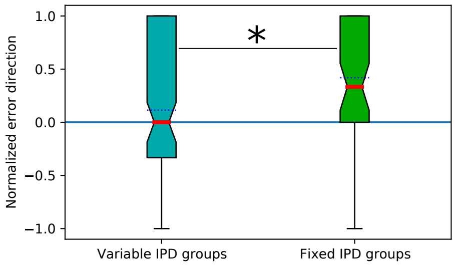


Figure 5: Comparison of normalized score of evaluation error direction between groups with variable IPD and those with a fixed IPD

The Ranksum test didn't provide us a p-value low enough (0.21) to highlight a significant difference between these two samples which doesn't allow us to validate our last hypothesis.

4.1.7 Other results.

Group - Size evaluation error trend. We also compared the distribution of error direction of the scale estimation between each group using Ranksum tests (10 in total). Using the Bonferonni correction we observed one significant difference between "Control" and "Avatar Scale" groups with an uncorrected p-value of 0.0022. With a one-tailed Mann-Whitney U test with a p-value of $7.3 \cdot 10^{-4}$ we found that subjects from the avatar scale group tended to overestimate object sizes more than ones from the control group. Plots of these scores are available in Figure 6.

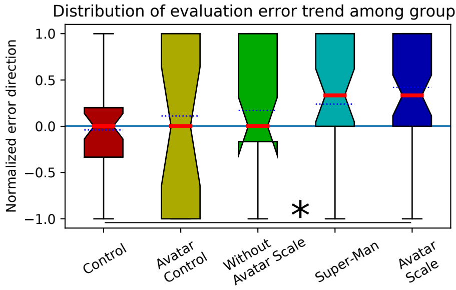


Figure 6: Comparison of normalized score of evaluation error direction between experimental groups

Grouped groups - Size evaluation error trend. In order to assess some of the more complex interactions at play between study variables, we compared the distribution of error direction amongst concatenated subgroups. Namely, we compared the avatar groups "Avatar Control", "Avatar Scale" and the remaining non-avatar groups using Ranksum tests (3 in total). Using the Bonferonni correction we observed one significant difference between the "Avatar Scale" and the remaining groups with an uncorrected p-value of 0.010. With a one-tailed Mann-Whitney U test with a p-value of 0.004 we found that subjects from the avatar scale group tended to overestimate object sizes more than ones from the remaining groups, as shown in Figure 7.

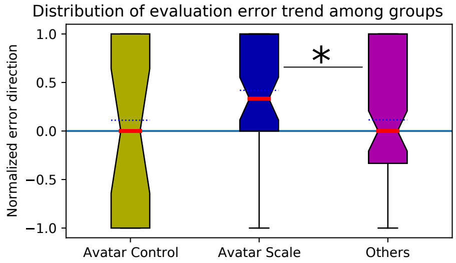


Figure 7: Comparison of normalized score of evaluation error direction between concatenated groups

5 DISCUSSION AND CONCLUSION

Most studies on the estimation of scale in virtual environments have been performed in highly controlled conditions. Our intention was to take an increasingly popular navigation technique - traveling at Giant scale - and assess whether it induces similar effects on scale estimation when employed in an open, real-world, self-directed, and feature-rich environment.

In contrast to our initial expectations, we showed that some commonly observed effects did not always present themselves. For instance, our first hypothesis assuming a general underestimation of object size for control groups (i.e. no Giant size) wasn't observed in this experiment. While we were not able to demonstrate uniform scale underestimation, as might have been expected, we did witness that users only showed about 60% size accuracy in their judgements, even though there was no predictable error direction. As presented earlier, it has been observed that in feature-rich environments [Henry and Furness 1993], size estimations may be skewed by as little as 20 percent. Given the feature-richness of our environment, and the wide range of self-direction allowed, it is perhaps not surprising that users demonstrated a less one-directionally skewed understanding of the scale of the environment.

Our second hypothesis, that scaling to Giant mode would skew size estimation more strongly, was not observed either. While at first glance this may seem surprising, given the large scale of the Giant mode we employed, it is perhaps to be expected in an experience where the user continually shifts from small to large scale. In addition, we forced users to return to 1:1 scale at every experimental question. While this ensured a uniform reference point, it meant that those using Giant mode were suddenly asked to evaluate the size of something from a vastly reduced relative scale, whereas the users who had been navigating at 1:1 scale witnessed no such change. The Giant mode users may have therefore interpreted their Giant size as the reference, and the 1:1 size as a "shrunken" body. This effect would have directly counteracted the underestimation effect, obscuring the role of Giant mode in size estimation, or, more importantly, showing that if a user can control their scale, perhaps they are not subject to the same scale distortion as might be expected.

Curiously, we showed the opposite of our third hypothesis, that subjects provided with an avatar would show more size underestimation than those without. This effect may again be explained by the construction of the experimental questions, as previously described. When the user is provided with an avatar, its presence may accentuate this effect. In order to test this hypothesis, we compared avatar subgroups which included this potential "shrinking" effect (Avatar Scale), without it (Avatar Control) and the remaining non-avatar groups. Indeed, we found that the Avatar Control group, see Figure 7, tended to overestimate, while the remaining groups did not demonstrate this effect, thus suggesting that our explanation for this effect is plausible.

Our hypothesis that variable IPD would result in decreased overestimation was readily validated by our experiment. In practical terms, this means that user scaling, as compared to the superman technique, maintains a better estimation of the true scale of the environment. This result may be unsurprising, given what we know about binocular vision, but its verification nonetheless reinforces the validity of the user-scaling approach to maintaining the most natural navigation in a virtual environment.

Moreover, the two previous results above agree with the additional observation that there is a statistically significant difference between the "Control" and "Avatar Scale" groups in terms of the error direction of size evaluation, as both differ in terms of the two relevant variables: IPD and the presence of an avatar.

Our last hypothesis, that the amount of time spent as a Giant would have an effect on size estimation, was not demonstrated. There may be a few reasons for that. First of all, users did not spend a large amount of time in Giant mode, with an average of approximately 47 seconds, out of an average total experience time of more than 6.5 minutes (so only approximately 10 percent time on average). Secondly, they were clustered quite narrowly around the mean, so distinctions between users may be difficult to observe. Lastly, the design of the experience, with constant switching from one scale to another, may in fact counteract the effect of time. Once a user becomes familiar with the two scales at play, they may not need to be exposed for a longer period in order for it to have a significant effect on their size judgement.

Conclusion. We observed trends which both contradicted and confirmed some well-known phenomena. Users with variable IPD showed a tendency to underestimate the VE scale compared to those with fixed IPD (Giant vs. Superman), but the net results amounted to over-estimation, rather than the underestimation taken to be the norm for users in a VE. The presence of an avatar indeed increased the magnitude of over-estimation, but again in the opposite sense to that predicted by the literature. Indeed, only the presence of both avatar and scalable IPD (Avatar scale group) provided results statistically significant when compared to the controls.

We presented a plausible explanation for the results we observed, namely an inversion of reference frame between normal and Giant, which suggests further work that could be done to better understand the variables at play in this novel navigation technique, particularly in the real-world case of feature-rich, open, freely-navigable virtual environments.

6 FUTURE WORK

The design of the experiment could be modified by including experimental groups where the user is not forced to normal scale at the time of experimental questioning, thereby examining the potential inversion of reference effect that was discussed. Time-based effects could also be examined by enforcing a specific amount of time in Giant mode for different experimental groups.

The complex nature of the interactions between the key variables examined here leaves ample room for re-imagining this experiment in different forms. With the increasing interest in viable navigation techniques in VR experiences, it seems that the utilisation of Giant scale navigation may increase, and that its effect on user perception and behaviour will become an increasingly interesting area for study.

ACKNOWLEDGMENTS

Thank you to the CERN Communication Group for funding this research. This work was also partially supported by the Swiss National Science Foundation, "Immersive Embodied Interactions in Virtual Environments" N°200020_178790.

REFERENCES

Parastoo Abtahi, Mar Gonzalez-Franco, Eyal Ofek, and Anthony Steed. 2019. I'm a giant: Walking in large virtual environments at high speed gains. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* 1–13.

Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. 2016. Point & Teleport Locomotion Technique for Virtual Reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '16).* Association for Computing Machinery, New York, NY, USA, 205–216. https://doi.org/10.1145/2967934.2968105

Sebastian Cmentowski, Andrey Krekhov, and Jens Krüger. 2019. Outstanding: A Multi-Perspective Travel Approach for Virtual Reality Games. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '19).* Association for Computing Machinery, New York, NY, USA, 287–299. https://doi.org/10.1145/3311350.3347183

Joyce Fung, Carol L Richards, Francine Malouin, Bradford J McFadyen, and Anouk Lamontagne. 2006. A treadmill and motion coupled virtual reality system for gait training post-stroke. *CyberPsychology & behavior* 9, 2 (2006), 157–162.

Daniel Henry and Tom Furness. 1993. Spatial perception in virtual environments: Evaluating an architectural application. In *Proceedings of IEEE Virtual Reality Annual International Symposium.* IEEE, 33–40.

Jangyoon Kim and Victoria Interrante. 2017. Dwarf or giant: the influence of interpupillary distance and eye height on size perception in virtual environments. In *Proceedings of the 27th International Conference on Artificial Reality and Telexistence and 22nd Eurographics Symposium on Virtual Environments.* 153–160.

JM Knapp. 1999. Visual Perception of Egocentric Distance in Virtual Environments unpublished Doctoral Dissertation. *Department of Psychology, University of California at Santa Barbara, Santa Barbara, CA* 93106 (1999).

Andrey Krekhov, Sebastian Cmentowski, Katharina Emmerich, Maic Masuch, and Jens Krüger. 2018. GulliVR: A Walking-Oriented Technique for Navigation in Virtual Reality Games Based on Virtual Body Resizing. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '18).* Association for Computing Machinery, New York, NY, USA, 243–256. https://doi.org/10.1145/3242671.3242704

Markus Leyrer, Sally A Linkenauger, Heinrich H Bülthoff, and Betty J Mohler. 2015. Eye height manipulations: a possible solution to reduce underestimation of egocentric distances in head-mounted displays. *ACM Transactions on Applied Perception (TAP)* 12, 1 (2015), 1–23.

Thammathip Piumsomboon, Gun A Lee, Barrett Ens, Bruce H Thomas, and Mark Billinghurst. 2018. Superman vs giant: a study on spatial perception for a multi-scale mixed reality flying telepresence interface. *IEEE transactions on visualization and computer graphics* 24, 11 (2018), 2974–2982.

Björn Van Der Hoort, Arvid Guterstam, and H Henrik Ehrsson. 2011. Being Barbie: the size of one's own body determines the perceived size of the world. *PloS one* 6, 5 (2011), e20195.

Lawrence E. Warren and Doug A. Bowman. 2017. User Experience with Semi-Natural Locomotion Techniques in Virtual Reality: The Case of the Virtuix Omni. In *Proceedings of the 5th Symposium on Spatial User Interaction (SUI '17).* Association for Computing Machinery, New York, NY, USA, 163. https://doi.org/10.1145/3131277.3134359

E User study of omnidirectional treadmill control algorithms in VR

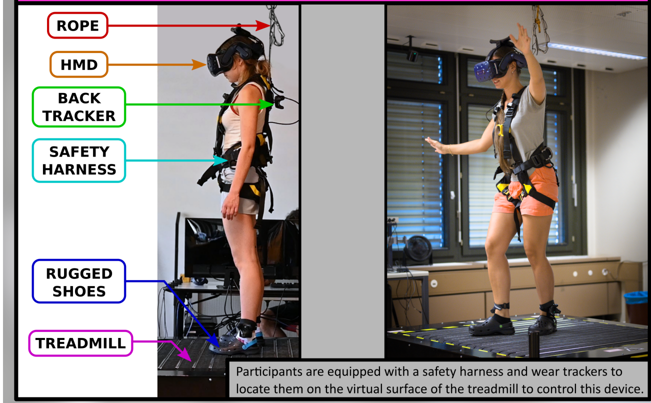
Full reference:
Delahaye, M., & Boulic, R. (2023, March). User study of omnidirectional treadmill control algorithms in VR. In 2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) (pp. 879-880). IEEE. https://ieeexplore.ieee.org/abstract/document/10108829.

USER STUDY OF OMNIDIRECTIONAL TREADMILL CONTROL ALGORITHMS IN VR

Mathias DELAHAYE - Ronan BOULIC

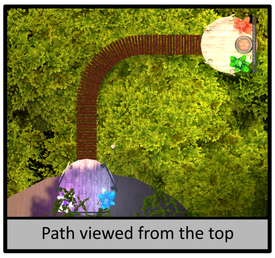
https://iig.epfl.ch

SETUP



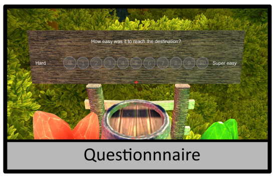
Participants are equipped with a safety harness and wear trackers to locate them on the virtual surface of the treadmill to control this device.

SCENE ILLUSTRATION



Path viewed from the top

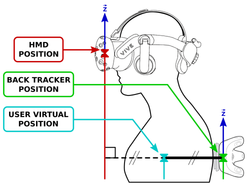
Start view

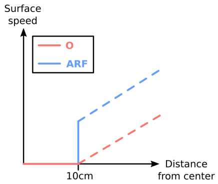


Questionnaire

The elevated suspended path acts as an implicit bias to tell participants to follow the trajectory. The canopy is also here to prevent fear of heights. Finally, questionnaires are directly integrated into the virtual environment.

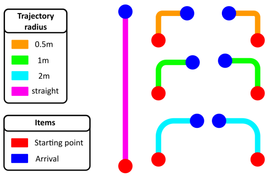
ALTERNATIVE ALGORITHM (ARF)

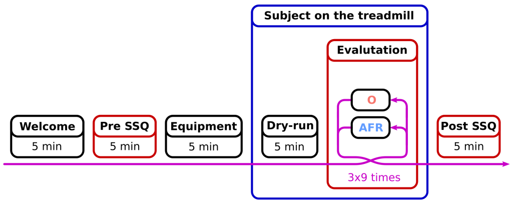




The center position of the user is computed using the back tracker. The local position center offset in this tracker's referential is computed at the calibration as the average position of the back tracker and the HMD's positions. A dead zone is implemented to avoid sending motor commands when the user mainly stands static in the middle. The core motion control loop (ARF) is sourced from kinematic equations from J. L. Souman, P. R. Giordano, I. Frissen, A. de Luca, and M. O. Ernst, "Making virtualwalking real: Perceptual evaluation of a new treadmill control algorithm" 2010. The original algorithm (O) is the one provided with the treadmill.

EXPERIMENTAL PROTOCOL

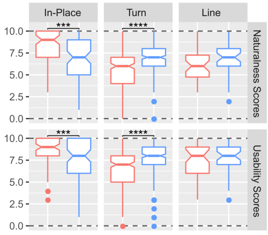


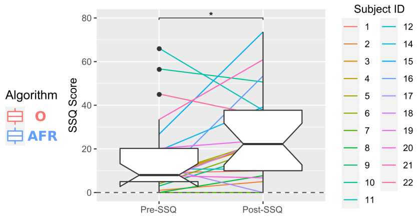


Participants are welcomed, give their informed consent, fill out a pre-SSQ questionnaire, are equipped, and then start the experiment. Participants have to follow the path disposed of in front of them (selected in a pseudo-randomized order) and then answer questions about usability/naturalness and iterate this with the different algorithms pseudo-randomly selected. Finally, participants fill out a post-SSQ questionnaire and receive compensation for their time.

SCAN ME

RESULTS AND CONCLUSION





Results show that AFR significantly performed better in terms of usability and naturalness for curved trajectories. No difference was observed on straight lines. On the other hand, O obtains significantly better scores for the naturalness of in-place trajectories. Indeed, when one moves for the in-place condition, one expects the treadmill to stay still. In the original algorithm, if the player moves slightly outside of the dead zone, the reaction is slight, whereas, with our implementation of the AFR, the control induces a small discontinuity. Finally, it was observed that walking on the treadmill did not fully alleviate the motion sickness as the post-SSQ score highlighted a significantly different level of motion sickness compared to the Pre SSQ scores. However, the final scores corresponded to the appearance of only minor symptoms.

Bibliography

3D, U. (2019). Unity 3d. https://unity.com/fr.

Aberman, K., Li, P., Lischinski, D., Sorkine-Hornung, O., Cohen-Or, D., and Chen, B. (2020). Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)*, 39(4):62–1.

Aggarwal, R., Mytton, O. T., Derbrew, M., Hananel, D., Heydenburg, M., Issenberg, B., MacAulay, C., Mancini, M. E., Morimoto, T., Soper, N., et al. (2010). Training and simulation for patient safety. *BMJ Quality & Safety*, 19(Suppl 2):i34–i43.

Al-Asqhar, R. A., Komura, T., and Choi, M. G. (2013). Relationship descriptors for interactive motion adaptation. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 45–53, New York, NY, USA. ACM.

Albert, L., Lance, F., Dubessy, M., Herbelin, B., Reymond, G., and Blanke, O. (2019). Real-time 360 Body Scanning System for Virtual Reality Research Applications. In *10th Int. Conference and Exhibition on 3D Body Scanning and Processing Technologies, Lugano, Switzerland*, pages 150–157. Hometrica Consulting.

Alexanderson, S., O'Sullivan, C., and Beskow, J. (2017). Real-time labeling of non-rigid motion capture marker sets. *Computers & Graphics*, 69:59–67.

Almousa, O., Prates, J., Yeslam, N., Gregor, D. M., Zhang, J., Phan, V., Nielsen, M., Smith, R., and Qayumi, K. (2019). Virtual reality simulation technology for cardiopulmonary resuscitation training: An innovative hybrid system with haptic feedback. *Simulation & Gaming*, 50(1):6–22.

Almusawi, A. R., Dülger, L. C., and Kapucu, S. (2016). A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242). *Computational intelligence and neuroscience*.

Andrews, S., Huerta, I., Komura, T., Sigal, L., and Mitchell, K. (2016). Real-time physics-based motion capture with sparse sensors. In *Proceedings of the 13th European Conference on Visual Media Production (CVMP)*, page 5. ACM.

Arduino (2019). *Arduino Uno Rev3 Schematics*. https://content.arduino.cc/assets/UNO-TH_Rev3e_sch.pdf.

Aristidou, A. (2018). Hand tracking with physiological constraints. *The Visual Computer*, 34(2):213–228.

Aristidou, A., Cameron, J., and Lasenby, J. (2008). Real-time estimation of missing markers in human motion capture. In *2008 2nd International Conference on Bioinformatics and Biomedical Engineering*, pages 1343–1346.

Aristidou, A. and Lasenby, J. (2013). Real-time marker prediction and CoR estimation in optical motion capture. *Visual Computer*, 29(1):7–26.

Aristidou, A., Lasenby, J., Chrysanthou, Y., and Shamir, A. (2017). Inverse kinematics techniques in computer graphics: A survey. *Computer Graphics Forum*, 37(6):35–58.

Baerlocher, P. and Boulic, R. (2004). An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 20(6):402–417.

Ballan, L. and Cortelazzo, G. (2008). Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes.

Balslev, D., Cole, J., and Miall, R. C. (2007). Proprioception contributes to the sense of agency during visual observation of hand movements: evidence from temporal judgments of action. *Journal of Cognitive Neuroscience*, 19(9):1535–1541.

Banakou, D., Groten, R., and Slater, M. (2013). Illusory ownership of a virtual child body causes overestimation of object sizes and implicit attitude changes. *Proceedings of the National Academy of Sciences of the United States of America*, 110(31):12846–12851.

Banakou, D., Kishore, S., and Slater, M. (2018). Virtually being Einstein results in an improvement in cognitive task performance and a decrease in age bias. *Frontiers in Psychology*, 9(JUN):917.

Baran, I. and Popovi, J. (2007). Automatic rigging and animation of 3D characters. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*.

Basset, J., Ouannas, B., Hoyet, L., Multon, F., and Wuhrer, S. (2022). Impact of self-contacts on perceived pose equivalences. In *Proceedings of the 15th ACM SIGGRAPH Conference on Motion, Interaction and Games*, MIG '22, pages 1–10, New York, NY, USA. Association for Computing Machinery.

Basset, J., Wuhrer, S., Boyer, E., and Multon, F. (2020). Contact preserving shape transfer: Retargeting motion from one shape to another. *Computers & Graphics*, 89:11–23.

Bergeron, H. E. (2019). A virtual reality system for realistic cardiopulmonary resuscitation training. *The University of Arizona*.

Blakemore, S. J., Wolpert, D., and Frith, C. (2000). Why can't you tickle yourself?

Blanke, O. (2012). Multisensory brain mechanisms of bodily self-consciousness. *Nature Reviews Neuroscience*, 13(8):556–571.

Blanke, O. and Metzinger, T. (2009). Full-body illusions and minimal phenomenal selfhood. *Trends in Cognitive Sciences*, 13(1):7–13.

Blanke, O. and Mohr, C. (2005). Out-of-body experience, heautoscopy, and autoscopic hallucination of neurological origin: Implications for neurocognitive mechanisms of corporeal awareness and self-consciousness. *Brain research reviews*, 50(1):184–199.

Boban, L., Delahaye, M., and Boulic, R. (2020). Partial Finger Involvement Reflects into Grasping Tasks Performance and Accuracy. In Kulik, A., Sra, M., Kim, K., and Seo, B.-K., editors, *ICAT-EGVE 2020 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments - Posters and Demos*. The Eurographics Association.

Boban, L., Strauss, L., Decroix, H., Herbelin, B., and Boulic, R. (2023a). Unintentional synchronization with self-avatar for upper- and lower-body movements. *Frontiers in Virtual Reality*, 4.

Boban, L., Strauss, L., Decroix, H., Herbelin, B., Boulic, R., et al. (2023b). Unintentional synchronization with self-avatar for upper-and lower-body movements. *Frontiers in Virtual Reality*, 4.

Bodenheimer, B., Rose, C., Rosenthal, S., and Pella, J. (1997). The Process of Motion Capture: Dealing with the Data. *In: Thalmann D., van de Panne M. (eds) Computer Animation and Simulation '97. Eurographics. Springer, Vienna.*, pages 3–18.

Botvinick, M. and Cohen, J. (1998). Rubber hands 'feel' touch that eyes see. *Nature*, 391(6669):756–756. Publisher: Nature Publishing Group.

Bourlard, H. and Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biol. Cybern.*, 59(4-5):291–294.

Bovet, S., Debarba, H. G., Herbelin, B., Molla, E., and Boulic, R. (2018). The critical role of self-contact for embodiment in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1428–1436.

BraydenManikin (2019). https://www.braydenmanikin.co.uk/.

Burns, E. and Brooks, F. P. (2006). Perceptual sensitivity to visual/kinesthetic discrepancy in hand speed, and why we might care. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '06*, page 3. ACM Press.

Burns, E., Razzaque, S., Panter, A. T., Whitton, M. C., McCallus, M. R., and Brooks, Jr., F. P. (2006). The hand is more easily fooled than the eye: Users are more sensitive to visual interpenetration than to visual-proprioceptive discrepancy. *Presence: Teleoper. Virtual Environ.*, 15(1):1–15.

Buttussi, F., Chittaro, L., and Valent, F. (2020). A virtual reality methodology for cardiopulmonary resuscitation training with and without a physical mannequin. *Journal of Biomedical Informatics*, 111:103590.

Caspar, E. A., Cleeremans, A., and Haggard, P. (2015). The relationship between human agency and embodiment. *Consciousness and Cognition*, 33:226–236.

Celikcan, U., Yaz, I. O., and Capin, T. (2015). Example-based retargeting of human motion to arbitrary mesh models. *Computer Graphics Forum*, 34(1):216–227.

Choi, K.-J. and Ko, H.-S. (2000). Online motion retargetting. *The Journal of Visualization and Computer Animation*, 11(5):223–235.

Cobos, S., Ferre, M., Sanchez Uran, M. A., Ortego, J., and Pena, C. (2008). Efficient human hand kinematics for manipulation tasks. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2246–2251.

Cohen, J. (1988). The effect size. *Statistical Power Analysis for the Behavioral Sciences*, pages 77–83.

Csikszentmihalyi, M. and Csikzentmihaly, M. (1990). *Flow: The psychology of optimal experience*, volume 1990. Harper & Row New York.

Cummings, J. J. and Bailenson, J. N. (2016). How immersive is enough? a meta-analysis of the effect of immersive technology on user presence. *Media Psychology*, 19(2):272–309.

Das, A. and Deb, S. (2016). A neural network-based methodology for inverse kinematics of a multi-finger robotic hand for gripping. *International Journal of Intelligent Systems Technologies and Applications*, 15:281–294.

David, N., Newen, A., and Vogeley, K. (2008). The “sense of agency” and its underlying cognitive and neural mechanisms. *Consciousness and cognition*, 17(2):523–534.

De Vignemont, F. and Fourneret, P. (2004). The sense of agency: A philosophical and empirical review of the “who” system. *Consciousness and Cognition*, 13(1):1–19.

Debarba, H. G., Chagué, S., and Charbonnier, C. (2020). On the plausibility of virtual body animation features in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1880–1893.

Delahaye, M. G., Zbinden, B., Herbelin, B., and Boulic, R. (2021). On the importance of providing a tangible haptic response for training cardiopulmonary resuscitation in virtual reality. In *ICAT-EGVE 2020-International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*, pages 17–25. The Eurographics Association.

Dourish, P. (2001). *Where the Action is: The Foundations of Embodied Interaction*. MIT Press.

Engbert, K., Wohlschläger, A., and Haggard, P. (2008). Who is causing what? the sense of agency is relational and efferent-triggered. *Cognition*, 107(2):693–704.

Falcao, C., Lemos, A. C., and Soares, M. (2015). Evaluation of natural user interface: A usability study based on the leap motion device. *Procedia Manufacturing*, 3:5490–5495. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015.

Falkenstein, M., Hohnsbein, J., Hoormann, J., and Blanke, L. (1991). Effects of crossmodal divided attention on late erp components. ii. error processing in choice reaction tasks. *Electroencephalography and clinical neurophysiology*, 78(6):447–455.

Farrer, C., Bouchereau, M., Jeannerod, M., and Franck, N. (2008). Effect of distorted visual feedback on the sense of agency. *Behavioural neurology*, 19(1, 2):53–57.

Feng, A., Casas, D., and Shapiro, A. (2015). Avatar reshaping and automatic rigging using a deformable model. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, MIG 2015*, pages 57–64.

Gallagher, A., McClure, N., McGuigan, J., Crothers, I., and Browning, J. (1999). Virtual reality training in laparoscopic surgery: a preliminary assessment of minimally invasive surgical trainer virtual reality (mist vr). *Endoscopy*, 31(04):310–313.

Gallagher, S. (2007). The natural philosophy of agency. *Philosophy Compass*, 2(2):347–357.

Galvan Debarba, H., Boulic, R., Salomon, R., Blanke, O., and Herbelin, B. (2018). Self-attribution of distorted reaching movements in immersive virtual reality. *Computers & Graphics*, 76:142–152.

Galvan Debarba, H., Bovet, S., Salomon, R., Blanke, O., Herbelin, B., and Boulic, R. (2017). Characterizing first and third person viewpoints and their alternation for embodied interaction in virtual reality. *PLOS ONE*, 12(12):1–19.

Gao, B., Lee, J., Tu, H., Seong, W., and Kim, H. (2020). The effects of avatar visibility on behavioral response with or without mirror-visual feedback in virtual environments. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 780–781.

Gehring, W. J., Goss, B., Coles, M. G., Meyer, D. E., and Donchin, E. (1993). A neural system for error detection and compensation. *Psychological science*, 4(6):385–390.

Glardon, P., Boulic, R., and Thalmann, D. (2006). Robust on-line adaptive footplant detection and enforcement for locomotion. *The Visual Computer*, 22:194–209.

Gleicher, M. (1998). Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 33–42, New York, NY, USA. ACM.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323.

Gonzalez-Franco, M. and Berger, C. C. (2019). Avatar embodiment enhances haptic confidence on the out-of-body touch illusion. *IEEE Transactions on Haptics*, 12(3):319–326.

Gonzalez-Franco, M. and Peck, T. C. (2018). Avatar embodiment. towards a standardized questionnaire. *Frontiers in Robotics and AI*, 5:74.

Grassia, F. S. (1998). Practical parameterization of rotations using the exponential map. *Journal of graphics tools*, 3(3):29–48.

Guo, S., Southern, R., Chang, J., Greer, D., and Zhang, J. J. (2015). Adaptive motion synthesis for virtual characters: a survey. *The Visual Computer*, 31:497–512.

Haggard, P. (2008). Human volition: towards a neuroscience of will. *Nature Reviews Neuroscience*, 9(12):934–946. Number: 12 Publisher: Nature Publishing Group.

Haggard, P. (2017). Sense of agency in the human brain. *Nature Reviews Neuroscience*, 18(4):196.

Han, S., Liu, B., Wang, R., Ye, Y., Twigg, C. D., and Kin, K. (2018). Online optical marker-based hand tracking with deep labels. In *SIGGRAPH*.

Harish, P., Mahmudi, M., Callennec, B. L., and Boulic, R. (2016). Parallel inverse kinematics for multithreaded architectures. *ACM Trans. Graph.*, 35(2):19:1–19:13.

Hasan, A., Ismail, N., Hamouda, A., Aris, I., Marhaban, M. H., and Al-Assadi, H. (2010). Artificial neural network-based kinematics jacobian solution for serial manipulator passing through singular configurations. *Advances in Engineering Software*, 41:359–367.

Heeter, C. (1999). Aspects of Presence in Telerelating. *CyberPsychology & Behavior*, 2(4):325–335.

Henmon, V. A. C. (1911). The relation of the time of a judgment to its accuracy. *Psychological review*, 18(3):186.

Herda, L., Fua, P., Plänkers, R., Boulic, R., and Thalmann, D. (2000). Skeleton-based motion capture for robust reconstruction of human motion. In *Proceedings of the Computer Animation*, CA '00, pages 77–, Washington, DC, USA. IEEE Computer Society.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Holden, D. (2018). Robust solving of optical motion capture data by denoising. *ACM Trans. Graph.*, 37(4):165:1–165:12.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Hoyet, L., Argelaguet, F., Nicole, C., and Lécuyer, A. (2016). “wow! i have six fingers!”: Would you accept structural changes of your hand in vr? *Frontiers in Robotics and AI*, 3:27.

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. (2016). Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Jaekl, P., Allison, R., Harris, L., Jasiobedzka, U., Jenkin, H., Jenkin, M., Zacher, J., and Zikovitz, D. (2002). Perceptual stability during head movement in virtual reality. In *Proceedings IEEE Virtual Reality 2002*, pages 149–155.

Javaheri, H., Gruenerbl, A., Monger, E., Gobbi, M., and Lukowicz, P. (2018). Stayin’alive: An interactive augmented: Reality cpr tutorial. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 365–368.

Jeannerod, M. (2003). The mechanism of self-recognition in humans. *Behavioural brain research*, 142(1-2):1–15.

Jeannerod, M. (2009a). *Cerveau volontaire (Le)*. O. Jacob.

Jeannerod, M. (2009b). The sense of agency and its disturbances in schizophrenia: A reappraisal. In *Experimental Brain Research*, pages 527–532. Springer.

Jeunet, C., Albert, L., Argelaguet, F., and Lécuyer, A. (2018). “do you feel in control?”: towards novel approaches to characterise, manipulate and measure the sense of agency in virtual environments. *IEEE transactions on visualization and computer graphics*, 24(4):1486–1495.

Jiang, K., Chen, H., and Yuan, S. (2005). Classification for Incomplete Data Using Classifier Ensembles. *Proc International Conference on Neural Networks and Brain, 2005. (ICNN&B 05)*, 1:559–563.

Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45.

Kamina, P. (2009). *Anatomie clinique*. Maloine.

Kannape, O. A. and Blanke, O. (2013). Self in motion: sensorimotor and cognitive mechanisms in gait agency. *Journal of neurophysiology*, 110(8):1837–1847.

Khanal, P., Vankipuram, A., Ashby, A., Vankipuram, M., Gupta, A., Drumm-Gurnee, D., Josey, K., Tinker, L., and Smith, M. (2014). Collaborative virtual reality based advanced cardiac life support training simulator using virtual reality principles. *Journal of Biomedical Informatics*, 51:49–59.

Kilteni, K., Groten, R., and Slater, M. (2012). *The Sense of Embodiment in Virtual Reality*, volume 21. MIT Press, Cambridge, MA, USA.

Kim, B.-H. (2014). An adaptive neural network learning-based solution for the inverse kinematics of humanoid fingers. *International Journal of Advanced Robotic Systems*, 11(1):3.

Kim, J.-S. and Park, J.-M. (2016). Direct and realistic handover of a virtual object. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 994–999. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Kokkinara, E. and Slater, M. (2014). Measuring the effects through time of the influence of visuomotor and visuotactile synchronous stimulation on a virtual body ownership illusion. *Perception*, 43(1):43–58. PMID: 24689131.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc.

Krugwasser, A. R., Harel, E. V., and Salomon, R. (2019). The boundaries of the self: The sense of agency across different sensorimotor aspects. *Journal of Vision*, 19(4):14–14. Publisher: The Association for Research in Vision and Ophthalmology.

Kulpa, R., Multon, F., and Arnaldi, B. (2005). Morphology-independent representation of motions for interactive human-like animation. In *Eurographics*.

Kwon, Y., Lee, S., Jeong, J., and Kim, W. (2014). Heartisense: A novel approach to enable effective basic life support training without an instructor. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 1699–1704, New York, NY, USA. ACM.

Langner, I., Henker, C., Steinhagen, K., Bülow, R., Langner, S., and Schmidt, C.-O. (2020). Can sacrum height predict body height, age, and sex? a large population-based mri study. *Forensic Imaging*, 21:200379.

Lanier, J. (1988). A vintage virtual reality interview.

LeapMotion (2019). Leapmotion. https://www.leapmotion.com/.

Lemaire, V. (2018). Introduction to cardiopulmonary resuscitation in virtual reality (vr) actions that save. In *Proceedings of Gamification and Serious Games Symposium 2018*, pages 22–23, Neuchatel, Switzerland. HE-Arc.

Li, J., McCann, J., Pollard, N., and Faloutsos, C. (2010). Bolero: A principled technique for including bone length constraints in motion capture occlusion filling. In *Proceedings of ACM SIGGRAPH / Eurographics Symposium on Computer Animation, Madrid, Spain*.

Li, R., St George, R. J., Wang, X., Lawler, K., Hill, E., Garg, S., Williams, S., Relton, S., Hogg, D., Bai, Q., and Alty, J. (2022). Moving towards intelligent telemedicine: Computer vision measurement of human movement. *Computers in Biology and Medicine*, 147:105776.

Logan, G. D. and Crump, M. J. C. (2010). Cognitive illusions of authorship reveal hierarchical error detection in skilled typists. *Science*, 330(6004):683–686.

Lyard, E. and Magnenat-Thalmann, N. (2007). A simple footskate removal method for virtual reality applications. *The Visual Computer*, 23:689–695.

Maister, L., Slater, M., Sanchez-Vives, M. V., and Tsakiris, M. (2015). Changing bodies changes minds: Owning another body affects social cognition.

Manganas, A., Tsiknakis, M., Leisch, E., Ponder, M., Molet, T., Herbelin, B., Magnenat-Thalmann, N., and Thalmann, D. (2005). Just in time health emergency interventions: An innovative approach to training the citizen for emergency situations using virtual reality techniques and advanced it tools (the vr tool). *The Journal on Information Technology in Healthcare*.

Manjrekar, S., Sandilya, S., Bhosale, D., Kanchi, S., Pitkar, A., and Gondhalekar, M. (2014). Cave: An emerging immersive technology – a review. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pages 131–136.

Manus-VR (2018). Manus vr - prime one. https://manus-vr.com/prime-one-gloves/.

Maran, N. J. and Glavin, R. J. (2003). Low- to high-fidelity simulation – a continuum of medical education? *Medical Education*, 37(s1):22–28.

Markley, F. L., Cheng, Y., Crassidis, J. L., and Oshman, Y. (2007). Quaternion averaging. *Journal of Guidance, Control, and Dynamics*.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., and Bethge, M. (2018). Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. Technical report, Nature Publishing Group.

MicroChip (2018). *atMega328 Datasheet*. https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf.

Microsoft (2019). Kinect. https://developer.microsoft.com/en-us/windows/kinect.

Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329.

Moeslund, T. B., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126.

Molet, T., Boulic, R., and Thalmann, D. (1999). Human motion capture driven by orientation measurements. *Presence: Teleoperators and Virtual Environments*, 8(2):187–203.

Molla, E., Galvan Debarba, H., and Boulic, R. (2017). Egocentric mapping of body surface constraints. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1.

Mourot, L., Hoyet, L., Clerc, F. L., and Hellier, P. (2022a). Underpressure: Deep learning for foot contact detection, ground reaction force estimation and footskate cleanup. *arXiv preprint arXiv:2208.04598*.

Mourot, L., Hoyet, L., Le Clerc, F., Schnitzler, F., and Hellier, P. (2022b). A survey on deep learning for skeleton-based human animation. In *Computer Graphics Forum*, pages 122–157. Wiley Online Library.

Mousas, C. and Anagnostopoulos, C.-N. (2017). Real-time performance-driven finger motion synthesis. *Computers & Graphics*, 65:1–11.

Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., and Theobalt, C. (2017). Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 10.

Multon, F., Kulpa, R., Hoyet, L., and Komura, T. (2009). Interactive animation of virtual humans based on motion capture data. *Computer Animation and Virtual Worlds*, 20(5-6):491–500.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA. Omnipress.

Nakul, E., Orlando-Dessaints, N., Lenggenhager, B., and Lopez, C. (2020). Measuring perceived self-location in virtual reality. *Scientific Reports*, 10(1):6802.

Nebel, J.-C. (1999). Keyframe interpolation with self-collision avoidance. In *Computer Animation and Simulation'99: Proceedings of the Eurographics Workshop in Milano, Italy, September 7–8, 1999*. Springer.

NeuronMocap (2018). Perception neuron. https://www.neuronmocap.com/products/perception_neuron.

Neyret, S., Navarro, X., Beacco, A., Oliva, R., Bourdin, P., Valenzuela, J., Barberia, I., and Slater, M. (2020). An Embodied Perspective as a Victim of Sexual Harassment in Virtual Reality Reduces Action Conformity in a Later Milgram Obedience Scenario. *Scientific Reports*, 10(1):1–18.

Niehorster, D., Li, L., and Lappe, M. (2017). The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research. *i-Perception*, 8(3).

Nielsen, T. I. (1963). Volition: A new experimental approach. *Scandinavian journal of psychology*, 4(1):225–230.

Oculus (2019). Oculus quest. https://www.oculus.com/blog/introducing-hand-tracking-on-oculus-quest-bringing-your-real-hands-into-vr/.

Osimo, S. A., Pizarro, R., Spanlang, B., and Slater, M. (2015). Conversations between self and self as Sigmund Freud - A virtual body ownership paradigm for self counselling. *Scientific Reports*, 5(1):1–14.

Padrao, G., Gonzalez-Franco, M., Sanchez-Vives, M. V., Slater, M., and Rodriguez-Fornells, A. (2016). Violating body movement semantics: Neural signatures of self-generated and external-generated errors. *Neuroimage*, 124:147–156.

Papagiannakis, G., Trahanias, P., Kenanidis, E., and Tsiridis, E. (2018). Psychomotor surgical training in virtual reality. *The adult hip-master case series and techniques*, pages 827–830.

Pavllo, D., Porssut, T., Herbelin, B., and Boulic, R. (2018). Real-time finger tracking using active motion capture: A neural network approach robust to occlusions. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pages 1–10.

Pavone, E. F., Tieri, G., Rizza, G., Tidoni, E., Grisoni, L., and Aglioti, S. M. (2016). Embodying others in immersive virtual reality: electro-cortical signatures of monitoring the errors in the actions of an avatar seen from a first-person perspective. *Journal of Neuroscience*, 36(2):268–279.

Pfeifer, R. and Bongard, J. C. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press.

PhaseSpace (2019). Phasespace impulse x2. https://phasespace.com/x2e-motion-capture/.

Piazza, T., Lundström, J., Kunz, A., and Fjeld, M. (2009). Predicting Missing Markers in Real-Time Optical Motion Capture. *LNCS*, 5903:125–136.

Porssut, T., Herbelin, B., and Boulic, R. (2019). Reconciling being in-control vs. being helped for the execution of complex movements in vr. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 529–537.

Porssut, T., Hou, Y., Blanke, O., Herbelin, B., and Boulic, R. (2021). Adapting virtual embodiment through reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics*.

Quam, D. L., Williams, G. B., Agnew, J. R., and Browne, P. C. (1989). An experimental determination of human hand accuracy with a dataglove. In *Proceedings of the Human Factors Society Annual Meeting*, volume 33, pages 315–319. SAGE Publications Sage CA: Los Angeles, CA.

Radianti, J., Majchrzak, T. A., Fromm, J., and Wohlgenannt, I. (2020). A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147:103778.

Rafferty, K., Nickleson, K., Devine, S., and Herdman, C. (2017). Improving the ergonomics of hand tracking inputs to vr hmds. In *International Conferences in Central Europe on Human Computer Interaction, Pilsen, Czech Republic*.

Raij, A., Kotranza, A., Lind, D. S., and Lok, B. (2009). Virtual experiences for social perspective-taking. In *Proceedings of the 2009 IEEE Virtual Reality Conference*, VR '09, pages 99–102, Washington, DC, USA. IEEE Computer Society.

RootMotion (2020). Final ik-vrik solver locomotion.

Rose, C., Cohen, M. F., and Bodenheimer, B. (1998). Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*.

Salomon, R. (2017). The assembly of the self from sensory and motor foundations. *Social cognition*, 35(2):87–106.

Salomon, R., Fernandez, N. B., van Elk, M., Vachicouras, N., Sabatier, F., Tychinskaya, A., Llobera, J., and Blanke, O. (2016). Changing motor perception by sensorimotor conflicts and body ownership. *Scientific Reports*, 6(1):25847. Number: 1 Publisher: Nature Publishing Group.

Sanchez-Vives, M. V. and Slater, M. (2005). Opinion: From presence to consciousness through virtual reality. *Nature Reviews Neuroscience*, 6(4):332–339.

Schröder, M., Maycock, J., and Botsch, M. (2015). Reduced marker layouts for optical motion capture of hands. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, MIG '15, pages 7–16, New York, NY, USA. ACM.

Schwind, V., Knierim, P., Haas, N., and Henze, N. (2019). Using presence questionnaires in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 360:1–360:12, New York, NY, USA. ACM.

Semeraro, F., Frisoli, A., Bergamasco, M., and Cerchiari, E. L. (2009). Virtual reality enhanced mannequin (vrem) that is well received by resuscitation experts. *Resuscitation*, 80(4):489–492.

Semeraro, F., Frisoli, A., Loconsole, C., Bannò, F., Tammaro, G., Imbriaco, G., Marchetti, L., and Cerchiari, E. L. (2013). Motion detection technology as a tool for cardiopulmonary resuscitation (cpr) quality training: A randomised crossover mannequin pilot study. *Resuscitation*, 84(4):501–507.

Semeraro, F., Ristagno, G., Giulini, G., Gnudi, T., Kayal, J. S., Monesi, A., Tucci, R., and Scapigliati, A. (2019). Virtual reality cardiopulmonary resuscitation (cpr): Comparison with a standard cpr training mannequin. *Resuscitation*, 135:234–235.

Sensoryx (2019). Vrfree glove system. https://www.sensoryx.com/product/vrfree_glove_system/.

Seymour, N. E., Gallagher, A. G., Roman, S. A., O'brien, M. K., Bansal, V. K., Andersen, D. K., and Satava, R. M. (2002). Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Annals of surgery*, 236(4):458.

Shao, L. (2016). Hand movement and gesture recognition using leap motion controller. *Virtual Reality, Course Report.*

Shapiro, A., Feng, A., Wang, R., Li, H., Bolas, M., Medioni, G., and Suma, E. (2014). Rapid avatar capture and simulation using commodity depth sensors. In *Computer Animation and Virtual Worlds*, volume 25, pages 201–211.

Shin, H. J., Lee, J., Shin, S. Y., and Gleicher, M. (2001). Computer puppetry: An importance-based approach. *ACM Trans. Graph.*, 20(2):67–94.

Sidhu, R. S., Park, J., Brydges, R., MacRae, H. M., and Dubrowski, A. (2007). Laboratory-based vascular anastomosis training: a randomized controlled trial evaluating the effects of bench model fidelity and level of training on skill acquisition. *Journal of vascular surgery*, 45(2):343–349.

Slater, M. (2003). A Note on Presence Terminology. *Presence Connect 3: 3.*

Slater, M., Banakou, D., Beacco, A., Gallego, J., Macia-Varela, F., and Oliva, R. (2022). A separate reality: An update on place illusion and plausibility in virtual reality. front. *Virtual Real. 3: 914392. doi: 10.3389/frvir.*

Slater, M., Pérez Marcos, D., Ehrsson, H., and Sanchez-Vives, M. V. (2009). Inducing illusory ownership of a virtual body. *Frontiers in Neuroscience*, 3(2):214–220.

Slater, M., Pérez Marcos, D., Ehrsson, H., and Sanchez-Vives, M. (2008). Towards a digital body: the virtual arm illusion. *Frontiers in Human Neuroscience*, 2:6.

Slater, M. and Steed, A. (2000). A Virtual Presence Counter. *Presence: Teleoperators and Virtual Environments*, 9(5):413–434.

Slater, M. and Wilbur, S. (1997). A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments. *MIT Press.*

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Stein, N., Niehorster, D. C., Watson, T., Steinicke, F., Rifai, K., Wahl, S., and Lappe, M. (2021). A comparison of eye tracking latencies among several commercial head-mounted displays. *i-Perception*, 12(1):2041669520983338. PMID: 33628410.

Steinicke, F., Bruder, G., Jerald, J., Frenz, H., and Lappe, M. (2010). Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):17–27.

Steptoe, W., Steed, A., and Slater, M. (2013). Human tails: Ownership and control of extended humanoid avatars. *IEEE Transactions on Visualization and Computer Graphics*, 19(4):583–590.

Sturman, D. J. and Zeltzer, D. (1994). A survey of glove-based input. *IEEE Computer Graphics and Applications*, 14(1):30–39.

Sutherland, I. E. et al. (1965). The ultimate display. In *Proceedings of the IFIP Congress*, volume 2, pages 506–508. New York.

Tian, N., Achache, K. H., Ben Mustapha, A. R., and Boulic, R. (2023). Egg objective characterization of cybersickness symptoms towards navigation axis. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 289–297.

Tian, Y., Meng, X., Tao, D., Liu, D., and Feng, C. (2015). Upper limb motion tracking with the integration of imu and kinect. *Neurocomputing*, 159:207–218.

Tkach, A., Pauly, M., and Tagliasacchi, A. (2016). Sphere-meshes for real-time hand modeling and tracking. *ACM Trans. Graph.*, 35(6):222:1–222:11.

Ullmer, B., Shaer, O., Mazalek, A., and Hummels, C. (2022). *Weaving Fire into Form: Aspirations for Tangible and Embodied Interaction*, volume 44. Association for Computing Machinery, New York, NY, USA, 1 edition.

Unzueta, L., Peinado, M., Boulic, R., and Ángel Suescun (2008). Full-body performance animation with sequential inverse kinematics. *Graphical Models*, 70(5):87–104.

Valve (2019). Valve index. https://www.valvesoftware.com/en/index.

Vicon (2019). Vicon shogun. https://www.vicon.com/software/shogun/.

Villegas, R., Yang, J., Ceylan, D., and Lee, H. (2018). Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8639–8648.

Vive (2022). Vive tracker v3.0. https://www.vive.com/us/accessory/tracker3/.

Vladimirov, G. and Koceski, S. (2019). Inverse kinematics solution of a robot arm based on adaptive neuro fuzzy interface system. *International Journal of Computer Applications*, 178:10–14.

Waegeman, T. and Schrauwen, B. (2011). Towards learning inverse kinematics with a neural network based tracking controller. In Lu, B.-L., Zhang, L., and Kwok, J., editors, *Neural Information Processing*, pages 441–448, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wen, W. (2019). Does delay in feedback diminish sense of agency? a review. *Consciousness and cognition*, 73:102759.

Wiley, D. and Hahn, J. (1997). Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Applications*, 17(6):39–45.

Wolpert, D. M., Ghahramani, Z., and Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882.

Yang, C., Liu, S., Lin, C., and Liu, C. (2020). Immersive virtual reality-based cardiopulmonary resuscitation interactive learning support system. *IEEE Access*, 8:120870–120880.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*.

Zhang, J., Chen, K., and Zheng, J. (2022). Facial expression retargeting from human to avatar made easy. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1274–1287.

Zhou, X., Sun, X., Zhang, W., Liang, S., and Wei, Y. (2016a). Deep kinematic pose regression. In *ECCV Workshop on Geometry Meets Deep Learning*, pages 186–201. Springer.

Zhou, X., Wan, Q., Zhang, W., Xue, X., and Wei, Y. (2016b). Model-based deep hand pose estimation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2421–2427. AAAI Press.

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. (1986). A hand gesture interface device. *SIGCHI Bull.*, 18(4):189–192.

Zlatanova, S. (2002). Augmented reality technology. *GISt Report No. 17, Delft, 2002, 72 p.*

Glossary

BiE Break in Embodiment. iii, 6

BiP Break in Presence. 5

CoR Center of Rotation. xiii, 11, 65, 70, 71

CPR Cardiopulmonary Resuscitation. xvi, 3, 109, 120, 122, 123, 129, 131, 132, 134–136

DoF Degree of Freedom. 12

HMD Head-Mounted Display. xi, xvi, 3–5, 10, 17, 19, 45–47, 50, 51, 58, 61, 101, 108, 114, 119–121, 131, 136

IK Inverse Kinematic. iii, v, xiv, 12, 17, 24, 26, 42, 46, 63, 78, 79, 94, 95, 97, 110, 139

IMU Inertial Measurement Unit. 9, 11, 17, 21, 22, 109

IPQ I-group Presence Questionnaire. 131, 133

MoCap Motion Capture. iii, v–vii, 9, 11, 12, 17, 61, 97, 102, 112, 113, 115

Mocap Motion Capture. iii, vii, 9, 46

PI Place Illusion. 5

PSI Plausibility Illusion. 5, 114

PV Person Viewpoint. xi, xvi, 3, 46, 61, 101, 108, 111, 112, 114, 116, 130

SE spontaneous errors. 46, 47, 49, 51–53, 55

SoA Sense of Agency. 6, 7, 43–46, 57, 59, 61

SoE Sense of Embodiment. iii, vi, vii, 5, 6, 8, 13, 43, 45, 57, 111, 115

VE Virtual Environment. iii, v, vii, 3, 4, 7, 8, 14, 17, 21, 43, 61, 62, 111, 115

VR Virtual Reality. iii, v, vii, xi, xii, xvi, 1–9, 11, 13, 14, 17–20, 22, 24, 26, 28, 30, 32, 34–36, 38–40, 42, 43, 45–47, 50, 51, 55, 57–59, 61, 101, 109, 110, 114, 115, 119, 121–123, 131, 132, 134, 136

Mathias DELAHAYE

PhD Student at **EPFL** and Engineer from **Mines DOUAI** in Computer Sciences

https://delahaye-group.fr

mathias-delahaye

mathias.delahaye

delahaye.mathias

LANGUAGES

French

English

Spanish - Professional

INTERESTS

Computer sciences (Back-end, Architecture, Security, Micro-controllers, etc.)

Photography (Portrait, Object presentation, Landscapes, etc.)

Electronics (PCB design, SMD soldering, low power IoT devices, etc.)

Music (Piano, Synthesizers, etc.)

Sports (Mountain Bike, Climbing, Mountaineering, 4WD, Danse, etc.)

EDUCATION

PhD Student

2019 - 2023

École Polytechnique Fédérale de Lausanne (EPFL) - Lausanne, Switzerland

I worked on Finger-Level Hand Control and Polymorphic Embodiment within the **IIG** at EPFL lead by **Dr. BOULIC Ronan**.

Engineer from Mines DOUAI - Master

2015 - 2018

Mines DOUAI - Douai, France

I majored in Computer Science at Mines DOUAI

Classes Préparatoires aux Grandes Ecoles - License

2012 - 2015

Lycée Claude Bernard - Paris, France

Mathematics Physics branch

EXPERIENCES

Teacher Assistant

2019 - 2023

EPFL - Lausanne, Switzerland

As a Ph.D. student, I assisted **Dr. BOULIC Ronan** for **Virtual Reality** and **C++ programming** courses with **hands-on support** or **automated evaluation tools**.

Master project supervision

2019 - 2023

EPFL - Lausanne, Switzerland

As a PhD student I supervised semester projects

Internship as a PhD Student

10/2018 - 11/2018

ENSTA Bretagne - Brest, France

During one month I had the chance to work with **Pr. LAGADEC Loïc**, to learn and work with FPGAs overlays at ENSTA Bretagne.

Infrastructure and electronic designer

2017-04 - 2017-08

Unéole - **Euratechnologies, Lille, France**

Design of a system allowing a windmill to send data to a sever Design of an electronic board providing a simple feedback control over a brake system Technologies used :

- Raspberry Pi
- PHP, Python
- Electronics
- UART, I2C

Application designer

2016-04 - 2016-07

Soufflet Group - **Nogent-Sur-Seine, France**

Design of an application providing support for printers Technologies used :

- C# with the .NET framework
- Microsoft SQL
- Microsoft AD

REWARDS

Lauz'Hack 2019 - Contest Winner For Bobst Challenge
2019

Lauzhack - Lausanne, Switzerland

Hackaton organized in different challenges to solve in 24h proposed by other companies. The prize was won by implementing an innovative way to control a Bobst fictive production line using finger tracking (Leap Motion) combined with Eye tracking (Tobii).

IC - Teacher Assistant Award
2021

EPFL - Lausanne, Switzerland

Prize won for my involvement in the ICC Course, with the implementation of an online autograder website allowing students to submit their projects and get live feedback on their grades, and in the VR course for which I wrote the practical part with hands-on available here Prize won for my involvment in the ICC Course, with the implementation of an online autograder website allowing students to submit their projects and get a live feedback on their grade, and in the VR course for which I wrote the practical part with hands-on avaliable here: **hands-on support**.

First Prize Photo Contest
2022

Club Photo EPFL / RESCO - Lausanne, Switzerland

Frist Prize won a collection of four pictures from EPFL's cafetarias

PUBLICATIONS

Avatar error in your favor: Embodied avatars can fix users' mistakes without them noticing

Mathias Delahaye, Olaf Blanke, Ronan Boulic, Bruno Herbelin

PLOS One

Real-Time Neural Network Prediction for Handling Two-Hands Mutual Occlusions

Dario Pavllo, Mathias Delahaye, Thibault Porssut, Bruno Herbelin, Ronan Boulic

Computers & Graphics : X

Does scaling player size skew one's ability to correctly evaluate object sizes in a virtual environment?

Neal Hartman - Mathias Delahaye, Hugo Decroix, Bruno Herbelin, Ronan Boulic

MIG '20: Motion, Interaction and Games

On the Importance of Providing a Tangible Haptic Response for Training Cardiopulmonary Resuscitation in Virtual Reality

Mathias Delahaye, Boris Zbinden, Bruno Herbelin, Ronan Boulic

ICAT-EGVE 2020

Partial Finger Involvement Reflects into Grasping Tasks Performance and Accuracy

Loën Boban, Mathias Delahaye, Ronan Boulic

ICAT-EGVE 2020