

Title	アバターデータセットに対するトポロジー非依存解析を用いたVR空間におけるアバター生成システムに関する研究
Author(s)	日比野, 友博
Citation	
Issue Date	2024-03
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/19056
Rights	
Description	Supervisor: 宮田 一乗, 先端科学技術研究科, 博士

Doctoral Dissertation

Study of Avatar Creation System in VR Using Topology Independent
Analysis for Avatar Data Set

Tomohiro Hibino

Supervisor Kazunori Miyata

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Knowledge Science)

March 2024

Abstract

The objective of this study is to develop an avatar creation system on Virtual Reality (VR) spaces and to study effective interface for VR to enhance human activities and avatar creation in "Metaverse". In particular, I aimed to study the creation of anime-like avatars, which has traditionally been challenging.

In recent years, the evolution of Head-Mounted Displays (HMDs) and smart glasses has enabled us to experience the virtual world as a three-dimensional space. The virtual world, which takes place just like the real world, is referred to as the **Metaverse**. Interactions within VR spaces offer various merits and are expected to become increasingly major parts in communication in the future.

In the Metaverse, a 3D model acting as a user's alter-ego is named **avatar**. While there is demand for photo-realistic avatars in Metaverses to emulate the real world, there is also demand for anime-like avatars that allow users to embody characters, particularly in online games and social networking services (SNS). Currently, anime-like avatars are predominantly created manually by experts after a long period of works. The process needs a wide range of skills, from design to engineering, making it difficult to be automated. Thus, creating an anime-like avatar that has both individuality and high-quality appearance is a tremendous challenge.

Learning from experts' skills has been a major theme in machine learning. While remarkable achievements are being reported one after another in fields such as image generation, applications of these methods in the field of 3D computer graphics (3DCG) are limited to relatively simple objects. Particularly, as far as I know, no method has been established to generate high-precision anime-like avatars. A significant reason for this is the data structure inherent to 3D data. The structure of 3D data, including the number of vertices, their positions, and the order in which they are connected, is known as **topology**. As each piece of 3D data possesses a different topology, it is challenging to perform uniform analysis.

In this study, I developed a method to unify the differences in topology possessed by these 3D data and generate a high-quality data set. Specifically, I unified the diverse topologies of each model by standardizing the template at the low-polygon level. By matching the landmarks of the template among the training data, I was able to learn the features of each model while maintaining a common topology. Subsequently, I adapted to the training data while increasing the number of polygons, enabling us to extract features

with high detail. Similarly, for texture images, I took the same approach to unify different texture images into a common topology by performing template matching on UV maps. Mesh and texture data, which were originally closely combined with each training data, were extracted separately. They are allowed to operate independently and combined each other.

In addition, in this study I developed an application that allows users to generate avatars in a VR space by utilizing the created data set. By operating a controller in the VR space, the user can create an avatar while watching the avatar change in real time. The results of user studies showed that the interface for creating avatars within the VR space is more effective in reducing work time than the conventional interface for manipulating on 2D screen.

Next, to verify effective interfaces within the VR space, I developed three types of applications: a physical interface in which operations are completed in the controller, a pseudo-button interface that emulates a conventional mechanical device, and an interface that falls between the two. The results of user studies showed that the physical interface, in which operations are completed in the controller, is effective in reducing operation time and increasing the percentage of operations that the user concentrates on. On the other hand, however, it was also found that users' concentrated operation in the VR space could be burdensome depending on their custom, especially without appropriate guides to assist them.

In short, I established a topology-independent data analysis method for anime-like avatars, which are among the avatars needed for activities in the metaverse. I also developed an application that generates avatars in VR using this technology, demonstrating its advantages over conventional software. I also conducted detailed comparative studies on interfaces within the VR space, verifying the efficiency of the user interface and identifying its issues. Through these contributions, this research contributed to the development of knowledge science.

Keywords : Avatar, Computer Graphics, Creation Assistant, Human Computer Interaction, Virtual Reality

Acknowledgement

It was in April 2018 when I bought an HTC Vive.

I can vividly remember the excitement I felt when I set up the base stations and experienced VR world. While working during the day, I struggled to create avatars late into the night and received feedback from my friends in the VR world. During those days, I acquired various techniques that have become the fundamental ability to advance my current research. Though it has now become rare to talk about modeling with that friends, those days are still precious to me.

I will repeatedly emphasize in this thesis the advantages of Metaverse while wearing avatars. I have dedicated this doctoral course to research, believing that society will be better if everyone can use avatars as they like in many situations.

It was a significant challenge for me to start research in an entirely new environment and field. The climate here in Hokuriku is harsh, with lots of rain and snow, and many days with strong winds. At times, these harsh conditions seemed to amplify the loneliness and challenges I faced in my research. The open atmosphere of the Miyata Laboratory played a significant role in enabling me to conduct my research, I think. I would like to express my great gratitude to Professor Miyata Kazunori for creating such an environment. We would also like to thank Professor Mikami, Professor Nishimoto, Professor Yuizono, and Associate Professor Xie for their advice during the defense process.

Finally, I would like to express my deepest gratitude to my family, who understood this challenge and supported me to the fullest.

Contents

1	Introduction	1
1.1	Metaverse	1
1.2	Metaverse and Society	2
1.3	Metaverse and Avatar	6
1.4	Avatar and User	7
1.5	Questions and Composition	8
2	Related Works	9
2.1	3D Avatar Generation	10
2.1.1	Difference between photo-realistic and anime-like avatars	10
2.1.2	Character making system	11
2.1.3	CG and generative technique	14
2.1.4	Overview of Multivariate Analysis	14
2.1.5	New Generation Method Using Natural Language	16
2.1.6	Structure of 3DCG	16
2.1.7	Analysis for 3DCG	18
2.1.8	3D Avatar generation	20
2.2	VR device and Interface	21
2.2.1	VR and Tracking	22
2.2.2	VR and User Interface	25
2.2.3	The Possibility of Gesture Input	26
2.2.4	The Disturbance of VR Device Itself	27
2.2.5	Absence of Feedback	28
2.3	Relation to This Study	30
3	Data Set Generation	31
3.1	Training data	31
3.2	Extraction of Data	33
3.3	Pilot Study	33
3.3.1	Marching Cube Reconstruction	34
3.3.2	Manual Landmark Morphing	37

3.4	Subdivision Shrink Method	40
3.5	Learning Texture Data	46
3.6	Distribtuion of each data set	47
4	Development of VR Application	54
4.1	Purpose and Issue	54
4.2	Allocation for VR Controller	54
4.3	Overview of Application	56
4.4	User Study	58
4.4.1	Method	58
4.4.2	Result	61
4.4.3	Discussion	61
4.4.4	Problems to be Solved	63
5	Study on VR Interfaces	64
5.1	Implementation	64
5.1.1	Plan of Interface	64
5.1.2	Overview of Appliation	65
5.2	User Study	68
5.2.1	The Flow of the User Study	68
5.2.2	Result	71
5.2.3	Discussion	77
5.3	Additional Study	86
5.3.1	Method	86
5.3.2	Result	87
5.3.3	Detailed Interview	89
6	Conclusion	94
6.1	Data Set Generation	94
6.2	Application and User Study	94
6.3	Limitations and Further Study	95
6.3.1	Training Data	95
6.3.2	Landmark Auto Estimation	96
6.3.3	Facial Expression	96
6.3.4	Analysis Method	96
6.3.5	Better VR Devices	97
6.3.6	Appropriate Guide and Environment	97
6.3.7	Hair Style	98
6.4	Contribution for Knowledge Science	98
A	Appendix	112

List of Figures

1.1	Samples of remote meeting applications . (Left Up) Zoom https://zoom.us/ (Right Up) WebEx https://www.webex.com/ (Left Down) SpatialChat https://spatial.chat/ (Right Down) GatherTown https://ja.gather.town/	3
1.2	A virtual event on VR platform "cluster" https://www.moguravr.com/cluster-update-virtual-youtuber/	4
1.3	Virtual meeting in new style with 3D position	5
1.4	Virtual meeting in traditional style with 2D position	5
1.5	VTubers of "Hololive", a famous production. They are acted by real humans but have virtual bodies. https://hololive.hololivepro.com/	7
2.1	A sample image of bone hierarchy structure. The typical bone structure consists of hips, chest, spine, legs etc. The hierarchy commons among many 3D software and video games.	10
2.2	Character making system in Vide Game "Dragon QuestX" https://www.dqx.jp/online/playguide/making/	12
2.3	Overview of VRoid Studio, a conventional character making system. It has many parameter to choice and user can adjust them by slider interface.	13
2.4	The top and middle row are cited by https://cems.riken.jp/jp/laboratory/qmtrt . and show shapes in the same mathematical topologies. A teacup could deform into a donut while only expanding and contracting. The bottom row shows data in the same 3D topologies. All data have 16 vertices and the same order so the left deform into the right while only moving vertices.	17
2.5	The difference of data structure between 2D and 3D. In 2D, images are composed from arrays of pixel that are formulated as $f(x, y) = (r, g, b)$. In 3D, images are rendered from separated data sources as Mesh, Light, Camera, Material. Each source has each dimension and variable length.	18

2.6	Image of 6 Dof. The left image shows 3 Dof of positions. The right image shows 3 Dof of rotations (Euler angles).	22
2.7	Face tracking application "Animaze" also as known as "FaceRig" https://www.animaze.us/	23
2.8	Openpose, pose estimation from video images. https://github.com/CMU-Perceptual-Computing-Lab/openpose	23
2.9	VICON, a dedicated motion tracking system. It uses a suit filled with markers and camera system. http://www.vicon.jp/	24
2.10	Tiltbrush is a drawing software in 3D VR spaces. The user can draw a line anywhere in the air. https://www.tiltbrush.com/	26
2.11	The image of HMD device "VIVE Pro". The size of HMD are large over the user's head and the controllers's shape are like sticks. https://www.vive.com/	27
2.12	Cited by [92]. Haptic feedback device that emulates the recoil impact of swinging.	29
2.13	Cited by [93]. A sample of tangible interface. The mechanical form of the interface itself works as a system and can be touched and moved.	29
3.1	Major commerce cite "BOOTH", https://booth.pm/ja . . .	31
3.2	The image of data extraction of the training data. Complete avatar data was load in Blender GUI (A). This image shows the extraction of facial parts. facial mesh was separated and saved as another mesh data file (B). Facial texture was obtained as orthogonal rendering result (C).	33
3.3	Cited by [100] The total combination of vertex states is nominally $2^8 = 256$, but after symmetrically reduced, 15 patterns are effective. Meshes are restored according to the vertex states as the figure shows.	34
3.4	The result of reconstructing by marching cube method. The left is the reconstructed data and the right is training data. . .	35
3.5	A sample of morphing by marching cube method. The left is a simple cube and the right is the reconstructed data.	36
3.6	A sample of morphing by marching cube method. The left is the reconstructed data A and the right is another reconstructed data B.	36
3.7	Reconstructed results using marching cube method by part. Each training data is separated to parts (Ear, Hair and head).	37
3.8	Landmark setting and approximation in manual landmarking method. All vertices of the base mesh were approximated to another training data by manual work.	37

3.9	Variations generated by manual landmarking method. Variations are reorganized by PCA.	38
3.10	Proportional editing. When one vertex is moved, vertices in a predefined range move with it, and the movement is proportional to the distance from the selected vertex. The circle means the limit where the weight of editing gets 0.	39
3.11	Flow of the Subdivision Shrink method. First, I set a common low polygon template and approximate it to the training data. Then, by iterating subdivision surface surface and shrink wrap 6 times, I got a template with the details of the training data.	41
3.12	Template matching between different training data. The 90 landmarks of the template are brought closer to the corresponding points in the training data. Landmarks consist of lines that pass through each part of the face. The figure picks up lines through the eyebrows, upper edge of the eyes, lower edge of the eyes, nose, and mouth.	42
3.13	The specific process of subdivision surface. Each single polygon surface is separated into pieces recursively. The left is the original mesh. The middle is the $n = 1$ iteration. The right is the $n = 2$ iteration.	42
3.14	The sample of the process of shrink wrap. The polygons of the sphere (yellow) is fit to the cube (lightblue) keeping the same topology.	43
3.15	Procedure of subdivision shrink. The first template is the top left one and the 5th one is the down right. The template mesh gets high-polygoned learning the details of the training data. .	44
3.16	The normalization process. X,Y,Z axes mean horizontal, depth, vertical. X,Y axis are normalized so that their means set to 0. Z axis are min-max scaled so that their value range is -1 to 1.	45
3.17	Inverse transformation of texture image. Transform the template's UV map (upper left) to match the target image (lower right) obtained by parallel projection rendering of the training data (the result is upper right). Then, by performing the inverse transformation on the target image, I get a unified texture image in the form of template UV. (lower left)	47
3.18	The samples of interpolation of mesh. This sequence shows interpolation from the mesh that has feminine features (A) to the mesh has masculine features (B).	48
3.19	The samples of interpolation of texture. This sequence shows interpolation from the texture that has black eyes features (A) to the texture has blue eyes features (B).	48

3.20	Allocation of meshes and textures for two models. The difference in rows indicates the distribution of textures, with the top row showing the texture of Model A, the bottom row showing the texture of Model B, and the middle row showing a mixture of both. The difference in columns indicates the distribution of meshes, with the left column showing the mesh of Model A, the right column showing the mesh of Model B, and the middle column showing a mixture of both.	49
3.21	Variations of Mesh Principal Components (MPC). The top row represents MPC1, the middle row MPC2, and the bottom row MPC3. The left column shows changes at -3 S.D., the middle column represents the mean, and the right column indicates changes at +3 S.D.	50
3.22	Variations of Texture Principal Components (TPC). The top row represents TPC1, the middle row TPC2, and the bottom row TPC3. The left column shows changes at -3 S.D., the middle column represents the mean, and the right column indicates changes at +3 S.D.	51
3.23	The obtained mesh data set of all the training data.	52
3.24	The obtained texture data set of all the training data. The order of the training data of the same as Figure 3.23	53
4.1	An overview of the application. The participant stands in front of the virtual avatar in VR space. The participant wear HTC Vive headset and controllers connected to PC.	56
4.2	(Left) A participant using my system. (Right) The image the user sees via HMD display. The user can feel the avatar as if they are in front of the user.	57
4.3	The relation between controller and attributes. The left hand relates body deformation, the right hand relates face deformation. The left rotation axes correspond to PC1, PC2, PC3 of body, and the vertical position corresponds to the height of the avatar. The rotation 3 axes correspond to PC1, PC2, PC3 of face, and the vertical position correspond to the scale of the avatar.	57
4.4	cumulative explained variance ratio of PCA. The left 3 are FPC (Face Principal Component), The right 3 are BPC (Body Principal Component)	58
4.5	Comparison of the time to complete. The left is 2D-UI and the right is VR-UI.	61

5.1	An overview of operation C method. The left is an orthogonal perspective of the system. The right is point of view of a user. A user operate parameters by only controller's Euler angle. There is no additional pseudo-button in VR space and the user can operate all parameters with controller.	66
5.2	An overview of operation B method. The upper row is orthogonal perspectives of the system, the left is with the left sided pseudo-button, whereas the right is with the right sided. The lower is point of view of a user. A user can operate parameter using pseudo-buttons and virtual laser pointer. Green button means the parameters of the face mesh. Blue button means the the parameter of the face texture.	67
5.3	An over view of operation H method. The left is an orthogonal perspective of the system. The right is point of view of a user. The cubes means what parameters a user is selecting. The upper 3 cubes means the face mesh, whereas the lower 3 cubes means the face texture. Increase/decrease of the parameter can be operated with controller.	68
5.4	A Sample of overlay graph of the user study. Horizontal axis means the flow of time, Vertical axes are 0,1 normalized proportion of parameters. Parameters are principal components of BS(Blend Shape) 1,2,3 and TW(Texture Weight) 1,2,3 and avatar height(PosY).	72
5.5	A Sample of trajectory graph of the user study. Horizontal axis are the participant's posX (left-right direction) and posZ (front-back direction). The center red circle is the where avatar stands. The black circle and the white circle is the participant's start and end position. The trajectory where the participant moved is the sequence of blue line. The trajectory color gets pale as the time flows.	73
5.6	Boxplot of method and duration. C method have significantly decreases duration than other methods.	76
5.7	Boxplot of method and valid. C method have significantly increase valid value than other methods.	77
5.8	Total results of the participant 1. The intensively operation areas are marked by yellow circle.	79
5.9	Total results of the participant 3. The operation intensively areas are marked by yellow circle.	84
5.10	Total results of the participant 5. The intensively operation areas are marked by yellow circle.	85
5.11	The examples of the created avatars by additional study. . . .	88

5.12	The Results of Participant 1 Task 1 C method on the additional study. The intensively operation areas are marked by yellow circle.	89
5.13	The Results of Participant 1 Task 2 C method on the additional study. The intensively operation areas are marked by yellow circle.	90
5.14	The Results of Participant 8 Task 2 C method on the additional study. The intensively operation areas are marked by yellow circle.	91
5.15	The Results of Participant 8 Task 2 H method on the additional study. The intensively operation areas are marked by yellow circle.	91
A.1	Total results of participant 1 of the user study in Chapter 5 .	113
A.2	Total results of participant 2 of the user study in Chapter 5 .	114
A.3	Total results of participant 3 of the user study in Chapter 5 .	115
A.4	Total results of participant 4 of the user study in Chapter 5 .	116
A.5	Total results of participant 5 of the user study in Chapter 5 .	117
A.6	Total results of participant 6 of the user study in Chapter 5 .	118
A.7	Total results of participant 7 of the user study in Chapter 5 .	119
A.8	Total results of participant 8 of the user study in Chapter 5 .	120
A.9	Total results of participant 9 of the user study in Chapter 5 .	121
A.10	Total results of participant 10 of the user study in Chapter 5 .	122
A.11	Total results of participant 11 of the user study in Chapter 5 .	123
A.12	Total results of participant 12 of the user study in Chapter 5 .	124

List of Tables

2.1	The summary limitations of 3D analyses	20
2.2	The advantages and disadvantages of tracking methods. "Acc" means accuracy, "Eco." means economy (lower cost).	25
3.1	The training data used for data set generation.	32
4.1	The changes of shape based on PC. The upper 3 rows show the changes about face, and the lower 3 rows show the changes about body. I extend the range of PC from ± 3 SD to ± 6 SD to exaggerate the changes. The changing of body shapes are shown from the side to reveal its depth change.	59
4.2	The result of the questionnaire	60
4.3	Time to complete work	60
5.1	Summary of the 3 types of operations method.	69
5.2	The orders by participants. Every single characters stands for methods. C means Controller method. H means Hybrid method. B means Button method. B* means Button method with mirror arrangement.	69
5.3	All measured data of the user study	74
5.4	All questionnaire results of user study	75
5.5	Table of ANOVA Probabilities. Each column are conditions, Turn, Method, and their interaction. Turn condition means difference by 1st,2nd,3rd studies par a participant. Method condition means difference by C,H,B methods. * star means probabiliy under 5 % so it has significant difference.	76
5.6	Correlation Matrix of measuread data and questionnaire answer	77
5.7	The results of the additional study. Eval means the overall evaluation of the study (questionnaire answer). Each method columns means the duration time (s).	87

5.8 Table of ANOVA Probabilities of the additional study. Each column are conditions, Turn, Method, and their interaction. Turn condition means difference by 1st, 2nd, 3rd, 4th, 5th and 6th studies par a participant. Method condition means difference by C,H,B methods. * star means probabiliy under 1 % so it has significant difference. 87

Chapter 1

Introduction

In this chapter, I will discuss the overall picture of the metaverse, its definition, origins, and relationship to society. I will also discuss the concept of avatars, which are the alter egos of users in the metaverse and discuss how avatars play an important role in the metaverse. In addition, I will also discuss the different roles of photo-realistic and non-photo-realistic avatars.

1.1 Metaverse

In recent years, the evolution of Head-Mounted Displays (HMDs) and smart glasses has enabled us to perceive a world rendered in computer graphics (CG) as a real three-dimensional space. A space where communications and economic activities occur, including selling or buying merchandise or paying service fees, is referred to as Virtual Reality (VR). In VR space, it is possible to communicate with others and conduct economic activities just like in the real world. This concept is known as the **Metaverse** [1].

The term "Metaverse" is a portmanteau of "meta-" meaning "beyond", and "-verse" meaning "universe". It signifies a world that transcends the existing physical universe. The term first appeared in the 1992 science fiction novel "Snow Crash" [2]. The motif of "another world" frequently appears in many science fiction works. For example, from Japanese media including the "Ghost in the Shell" and "Sword Art Online" series, as well as the "Matrix" series that are famous worldwide. Even before the popularization of novels and films as known today, mechanisms to create "another world" evidently exists in human mind. Some text says that religious paintings from prehistoric times, the Lascaux cave paintings, is a primitive form of VR experience [3]. From this perspective, humans have always possessed the desire to escape to "another world". With the development of VR technology, computers are

now able to realize this imagination.

In the paragraph above, the term "virtual" is referred almost interchangeably with "simulated" or "imaginary". Predominantly, these applications are confined to anime-like films and movies for dramatic effect. However, the utilities of the metaverse extend beyond the confines of these fictional representations. Originally, the term "virtual" implies an aspect of "substitution in essence". Hence, a metaverse can also serve as a means to alleviate the inconveniences of reality and to add new value to it. Direct societal benefits are arguably more pronounced in these aspects. One typical example is the technology of "tele-existence" or "tele-operation" in robotics context [4], where one can simulate a presence in a physically distant location through robotic intermediaries. Moreover, technologies that add a sense of reality to the physical world by overlaying CG images are known as Augmented Reality (AR) or Mixed Reality (MR). These technologies hold promising implications for road navigation and skill learning and can potentially give incalculable effects on the real world.

Therefore, while the metaverse is based fundamentally on CG technology and creates a world distinct from the reality, it is not conceived as a world independent from the physical reality. Instead, it can be understood as a new organic world that evolves in dynamic connection with the physical reality.

1.2 Metaverse and Society

The field that is included by metaverse is too vast. Since it fundamentally substitutes for the existing world, from culture to economic activities. In particular, this study aims to discuss the human relationships that are changing with metaverse.

The influence of metaverse on human relationships is essential. As it is often said, "appearance accounts for 90% of the impression", it's clear that physical appearance holds the most substantial information at the first impressions and subsequent communication in human relationships. While it is very difficult to alter one's appearance in the real physical world, VR world enables it complete freedom in this regard. This represents a revolutionary shift from traditional human relationships. When experiences such as changing one's gender, becoming a completely different individual, or transforming into a non-human entity become possible, it is supposed that self-awareness and identities will fundamentally evolve.

Moreover, metaverse offers freedom from physical constraints. It redefines various sensory activities, such as dining and traveling, which were previously deemed meaningful only when experienced in **reality**. All these

activities would be increasingly replaced in the near future. In such VR world, the values associated with human social activities are likely to undergo significant transformation. It is not only about leisurely pursuits, but also fundamental activities such as communication and work are expected to become essentially different. Research related to VR also includes a wide range of social activities and relationships, such as navigation [5] [6], therapy [7] [8], efficiency improvement [9] [10], medical training [11], and many others.

By the way, the world is increasingly promoting remote activities for various social reason. The outbreak of a novel infectious disease has expedited the adoption of tools such as remote meetings, Zoom, and WebEx, implemented to reduce physical contact (Figure 1.1) [12] [13] [14]. These tools, lacking in physicality and spatial expression, can be considered a form of VR in the sense of providing alternative realities. Furthermore, VR platforms like "Cluster" are realizing live events and such in VR spaces (Figure 1.2). In this manner, it is supposed that a significant proportion of societal activities will transition towards remote and VR environments in the future. VR SNS like VRChat and Horizons are already VR platforms in practical use among many users, and such metaverse services are expected to become a major means of communication in the future [15].

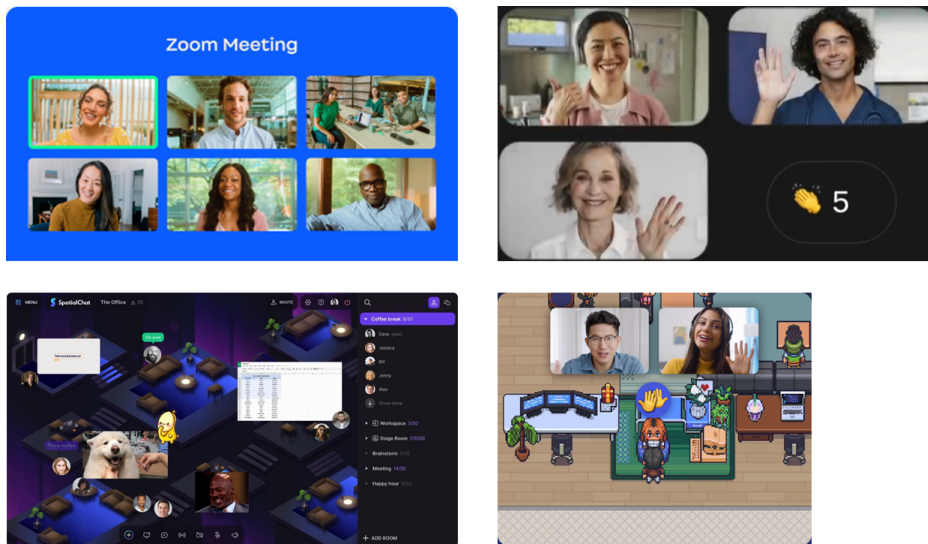


Figure 1.1: Samples of remote meeting applications . (Left Up) Zoom <https://zoom.us/> (Right Up) WebEx <https://www.webex.com/> (Left Down) SpatialChat <https://spatial.chat/> (Right Down) GatherTown <https://ja.gather.town/>



Figure 1.2: A virtual event on VR platform "cluster" <https://www.moguravr.com/cluster-update-virtual-youtuber/>

As a subtheme of this study, entitled "Virtual Meeting", I implemented a meeting environment with a three-dimensional arrangement, an arrangement that is difficult to achieve in the presence of gravity (Figure 1.3). A three-dimensional meeting arrangement, where all speakers are equally distant from each other, is theoretically represented as a tetrahedron shape, but this is not feasible in reality. And I implemented a meeting environment with plane arrangement as a traditional style (Figure 1.4). These two environments were implemented as the worlds of VRChat. I conducted a user study to compare these two environments: 8 participants were divided into 2 groups, the one undergo plane to 3-dimensional and the other undergo 3-dimensional to plane, and asked to discuss two agendas in the same order. The user study results indicated that, despite the meetings with the three-dimensional arrangement taking more time, the questionnaire responses did not reflect this. Instead, they gave the impression that the meetings were shorter. Thus, it was discovered that meetings in a three-dimensional setup made participants feel as if the time spent was shorter compared to traditional meetings in a flat arrangement.

Thus, the effect of the metaverse is not merely an increase in convenience, but one that has the potential to change the entire activity involved in human interaction.



Figure 1.3: Virtual meeting in new style with 3D position



Figure 1.4: Virtual meeting in traditional style with 2D position

1.3 Metaverse and Avatar

Within the metaverse, a 3D character that serves as one's alter ego is necessary. This is referred to as an "avatar". Ideally, an avatar should reflect the user's physical movements and expressions, but there is no necessity for it to exactly emulate the real world. Here, based on the function of avatars, I would like to discuss them in two broad categories: **photo-realistic avatars** and **non-photo-realistic avatars**.

Realistic avatars are primarily used as substitutes for the real world. Their main applications include remote meetings and remote work. The expected function here is to emulate reality while ignoring physical barriers, which can be considered an extension of the real existence. As previously mentioned, in "Tele-Existence", it is desirable for avatars to directly reflect real human beings.

However, since VR comprises 3D data, there is no obligation to simulate reality. It is possible to have a completely different form from reality. One can choose for a playful appearance when meeting friends or adopt a different look when playing a game. Particularly in MMORPGs, one might be required to transform into a fantastical being different from reality. In this context, I would like to refer to such avatars as **anime-like** avatars. Unrealistic avatars, with their anime-like style, have a unique appeal that sets them apart from photo-realistic avatars, and they play an important role in many forms of entertainment. The effect of wearing and acting out an avatar is not merely role-playing a being who is not that person, but has also been reported to have psychological effects that change a person's perception of self [16] [17] [18] [19]. Some research reports that the behavior through an avatar in VR can affect the same behavior in the real life. This phenomenon is known as the Proteus Effect [20] [21]. Becoming a virtual character in metaverse would be a whole new pleasure for humanity, and would even change one's perception of self [22].

The roles these avatars play are each significant in their respective directions. It would be inappropriate to attend an important business meeting as an anime character, just as it would be a mismatch to bring one's real physical self into a fantasy game world. In other words, these categories are not strictly separated but can be chosen or combined according to the situation. As demonstrated by the entities known as VTubers, there are those who maintain a fantastical appearance while performing jobs similar to idol streamers, not entirely divorced from reality (Figure 1.5). Such technology has the potential to create new culture that is different from the existing idol and streaming culture [23]. Combining physical elements such as voice and gestures with imaginative elements like appearance enables a performance

suitable for the required situation, a practice likely to become commonplace within the metaverse. In other words, in a metaverse society, people will be able to choose their personalities via avatar strategically [24].

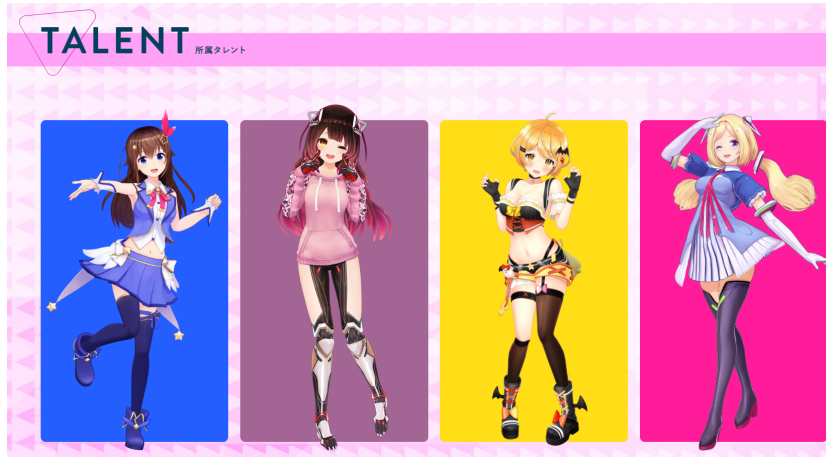


Figure 1.5: VTubers of "Hololive", a famous production. They are acted by real humans but have virtual bodies. <https://hololive.hololivepro.com/>

1.4 Avatar and User

In a metaverse society, users typically belong to multiple communities. Therefore, users need to have multiple avatars in order to use strategic identities as described before. And most of them are fictional and have no relationship to the real body. In such places, it would be of great benefit for users to be able to create many anime-like avatars as they like.

Ducheneaut says [25], the purposes for which users create avatars can be broadly classified as follows: to stand out, to conform to trends, and to idealize oneself. While the first and the second are common motivations in real-world fashion, the last one become a prominent element in a metaverse society. The distinctive feature of anime-like avatars is that a user can modify the original shape of the face and the body to achieve the ideal shape. Because of this characteristic, it has been reported that users sometimes spend a large amount of time creating avatars [26]. In addition, in some case, it is possible to make large and drastic changes from the real body, such as changing semi-beast [27] or changing gender [28].

As can be seen from these facts, it can be seen that the demand for anime-like avatars is potentially very large. They can be used in a fictional

world that is completely separate from real life, or even in the real world, they can be used to adjust an impression on others strategically. Therefore the variations are almost infinite, and means to customize them need to be developed from many aspects.

1.5 Questions and Composition

The Major Research Question (MRQ) of this study is, **How to make it easier for everyone to create anime-like avatars, which is essentially needed for activities in the metaverse?**

To achieve this goal, this study was conducted according to the following contents.

- Chapter 2 is **Related Works**. I introduced related works on 3DCG technologies related to avatar generation and interfaces using VR devices. I defined these Subsidiary Research Question (SRQ)s.
 - **How to generate data set and analyze for anime-like avatars?**
 - **Is it able to conduct a new VR interface superior to the conventional ones?**
 - **What it the essential usability of the VR interface?**
- Chapter 3 is **Data Set Generation**. I described how to develop an effective analysis method for the unified 3D data analysis of different topologies, which has been an essential difficulty point in 3D data analysis.
- Chapter 4 is **Development of VR Application**. I have developed a VR application that users can experience avatar creation. In addition, I conducted a user study to prove its superiority over conventional character making system.
- Chapter 5 is **Study on VR Interfaces**. I conducted a detailed study of VR interfaces. It proved that a VR-dedicated interface has many advantages, but at the same time, without proper guidance, it can be burdensome to users.
- Chapter 6 is **Conclusion**. I summarized the results of this study, outlines the limitations and future challenges. And, I summarized the contributions of this study to knowledge science.

Chapter 2

Related Works

In this chapter, I discuss related works on this study. First, I describe the overall technical characteristics of avatars and then describe the 3DCG technology used to synthesize them. In the latter half of this chapter, I will discuss tracking technologies for moving avatars in VR spaces and the relationship between these technologies and the effective user interface.

2.1 3D Avatar Generation

2.1.1 Difference between photo-realistic and anime-like avatars

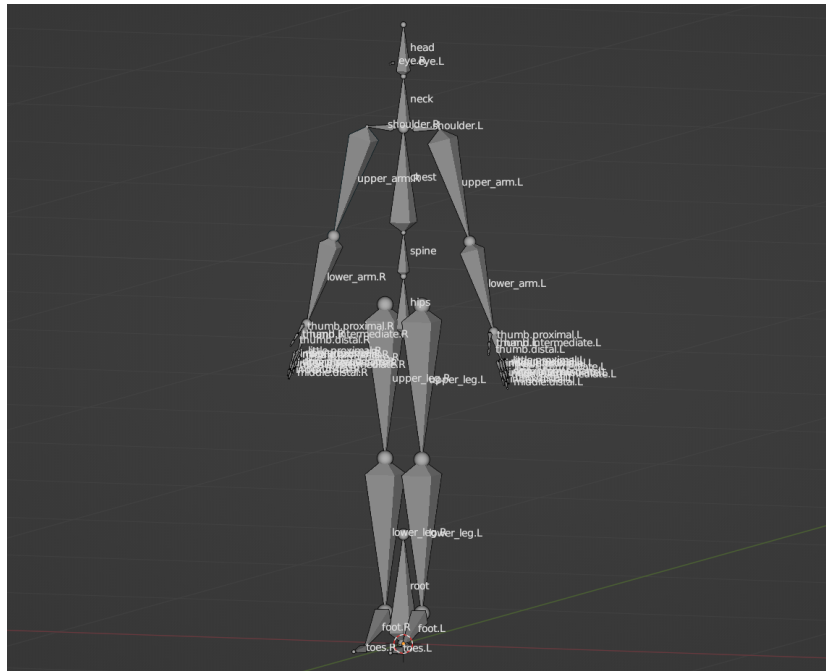


Figure 2.1: A sample image of bone hierarchy structure. The typical bone structure consists of hips, chest, spine, legs etc. The hierarchy commons among many 3D software and video games.

In this subsection, I will discuss the technological differences between photo-realistic avatars and anime-like avatars. Avatars are essentially 3D model data that follow a bone hierarchy structure known as "humanoid". They have the bones corresponding to the human joints, such as hips, chest, spine, legs, etc. (Figure 2.1). By combining them with motion tracking data, avatars can move in the VR space in the same way as users do in the real life.

Inevitably, the hierarchy of the bone is common to all humans. Therefore, by adding variations (e.g. height, body thickness) based on these bone structure, the body variations of most human can be represented. The SMPL model [29] is a powerful tool for creating realistic avatars, as it can generate differences in human body shape and pose with a small number of parameters. To turn real objects into 3D models, I can apply a technique called

photogrammetry [30]. This method estimates the 3D model from the portraits of the target object taken from multiple directions. At present, it is technically possible to create a photo-realistic avatar that reflects the features of a real human [31] [32] [33].

However, these methods cannot be applied to anime-like avatars. The reason is that anime-like avatars have locally super-deformed features. For example, the eye lines may be too sharp, or some parts of the body may be unnaturally emphasized. Also, their outfits can be so extravagant as to be impossible in reality, so they have much wider variety of style than photo-real ones. Such features are difficult to define as a consistent model with simple parameters. Unlike real humans, it is usually difficult to prepare portraits from multiple directions. Therefore, for creating anime-like avatars, it is essential to have 2 steps. First, a professional illustrator comes up with setting drawings like two-view diagrams. Second, a professional 3D creators faithfully turn them into 3D. Obviously, this requires an enormous amount of work time and communication by multiple professionals.

2.1.2 Character making system

Creating a 3D model is a highly advanced and specialized task. It involves a wide range of operations, such as creating an image sketch, generating a vertex mesh, creating materials that determine the texture of the surface, and rigging, which links joints to the movements of the model. Each of these tasks requires amounts of skills, and learning to do all of them yourself requires a vast investment of time. As the use of avatars becomes more popular, the user demand will continue to expand, but it is not practical for all users to acquire these skills. Therefore, a system is needed to make it easier for users without skills to create avatars.



Figure 2.2: Character making system in Vide Game "Dragon QuestX"
<https://www.dqx.jp/online/playguide/making/>

The most widespread approach to creating these anime-like avatars currently is a method called "character making system" where users operate a large number of parameters to create their preferred character (Figure 2.2). The typical ones are seen in famous video game series, "Dark Souls" and "Monster Hunter", etc. In fact, this method has been generalized in many video games, and as long as players are willing to invest time, they can create their preferred avatar within the limits set by the character creation system [34].

Even in old 2D video games, there were cases where characters were essentially created by changing hair color or changing the images of specific parts. But in these cases, the options are limited, so the variations are extremely limited.

Nintendo's MiiVerse is a pioneering example of creating anime-like models as 3D alter egos. This allows for considerable freedom in not only hairstyles and hair colors, but also the positioning of facial parts. Although it is recommended that the alter ego called Mii resembles to the user, their drawing style is in a toon style, so it is strictly doesn't need to be photo-realistic. And, some user plays by recreating unrealistic characters by this system.



Figure 2.3: Overview of VRoid Studio, a conventional character making system. It has many parameter to choice and user can adjust them by slider interface.

Even with such system, it was impossible to express continuous feature. It only changes the positions and scales of discrete parts so cannot express the continuous change like face contour. To achieve this, it is necessary to link a specific group of vertices to a certain handle point so that they move in conjunction, which requires extremely fine adjustments. Currently, VRoid (Figure 2.3) is the software that makes it possible to do such continuous operation.

However, there are issues with this approach. First, the direction of the avatar must fall within the range provided by the creation system. In addition, for a beginner who has no specific image or anything to work with, manipulating the numerous parameters is a pain in itself. Even if it takes less time to learn to use all these tools compared to learning each professional tool, it still requires hours of struggle to master it.

In almost all of these character creations, many parts of the face can be adjusted. This is based on the fact that when humans recognize others, their appearance occupies a large part, and the face occupies a large part of the appearance. Neuroscience also supports the fact that human cognitive abilities are specialized for recognizing faces, and this is thought to have evolved as humans lead social lives [35].

The problem that many CG beginners will have is the deformation of meshes. The means of mesh editing are so limited while demands for changing the shape of the face, which is an important part of Avatar. It is also

important to provide texture to enrich the face. But, this requires illustration skills, and there are few beginners who have such ability. From this, supporting to deform the shape of the face and to create texture is the first key point in making character creation more efficient.

2.1.3 CG and generative technique

Computer graphics (CG) and automated generation exhibit an inherently harmonious relationship. During the dawning age of CG, the programs were operated in command-line user interface (CUI), primarily focusing on procedural content. So their automation was relatively easily applied. However, as CG advanced, it began to accommodate areas akin to illustrations.

In other words, CG evolved to articulate illustrations more intuitively as RGB data on a pixel canvas. In the contemporary era, one can create CG illustrations that are indistinguishable from reality using drawing devices such as graphics tablets on large canvases boasting resolutions of 4K or higher. Nevertheless, this type of CG has long been considered the domain of illustration professionals, with little room for computational intervention.

In recent years, with the rapid progress in deep learning, it has become possible to analyze and generate characteristics of illustrations including their theme and style. This has been accomplished by combining multiple layers of nonlinear functions, referred to as activation functions, enabling the analysis of nonlinear relationships that were difficult to analyze traditionally. These learning networks are often referred to as neural networks, owing to their resemblance to the signal transmission between neural cells in their numerous network structures.

Starting with Multilayer Perceptron (MLP) and convolutional networks, more advanced methods such as Long Short-Term Memory (LSTM) analysis [36] and transformers [37] have recently been developed, and their networks have become more sophisticated. Such methods used to require a huge amount of computation time because they need to repeat a huge amount of matrix computations. However, with the development of deep learning frameworks and the General-purpose computing on graphics processing units (GPGPU), deep learning has quickly become a powerful and popular tool.

2.1.4 Overview of Multivariate Analysis

Here, I will mention the overview of developments in multivariate analysis. Aggregating a large number of parameters and extracting the important ones is a major task in multivariate analysis. Simply discarding parameters

is called "projection" and is the most primitive means of extraction. Extracting the few important components is also important in designing the interface. This is because it allows the user to reduce the number of variables to be manipulated without reducing the degree of freedom. This enables us to extract an orthogonal linear composite of multiple components as new component, which can be interpreted as a projection on the normalized and rotated components in multiple dimensions. Principal component analysis (PCA) is a means of extracting such composite components in order of variance contribution, and is known as a simple and powerful multivariate analysis. Therefore, even in interfaces where many parameters must be manipulated, PCA can be used to achieve comparable operation with a small number of efficiently aggregated parameters [38] [39]. Component aggregation by PCA is not clustering with any intention, but when the analyzed data have obvious biases, they are often isolated by PCA. Therefore, eigenvalues from principal components can be a good indicator of data characteristics, and there are examples of such studies for face images [40] [41].

However, because such a transformation is no more than linear transformation, it has the weakness of not being able to aggregate parameters with nonlinear relationships.

The most major strength of multivariate analysis using neural networks is the ability to extract nonlinear relationships by relaying activation functions. The development of deep learning is diverse, ranging from simple tasks like recognizing digits in an image to generating images using autoencoders [42]. In particular, autoencoders that utilize latent space are advantageous as generative models. They reduce the dimensions of the training data during the neural network analysis. The reduced data can summarize the data effectively. This process parallels PCA in linear computation. The obtained bottleneck space, termed the latent space, helps extract multiple nonlinear features.

Furthermore, advanced methods utilizing Generative Adversarial Networks (GANs) [43] have produced remarkable results [44] [45] [46]. These models involve a mechanism whereby the generator and discriminator iteratively learn whether the data generated from noise is real or fake, thereby enhancing each other's performance. The generator resulting from such learning becomes an extremely efficient image generation model [47].

As a result of the evolution of these various techniques, generative models using deep learning can now almost perfectly replicate delicate nuances including an artist's style or a professional's touch.

2.1.5 New Generation Method Using Natural Language

Additionally, in the context of this major trend towards automated generation, it is worth mentioning the emergence of prompt models. Nowadays, with the aid of large language models (LLM) like BERT [48] or GPT [49], it is possible to generate illustrations from simple text prompts. Particularly, the Stable Diffusion method using diffusion models [50] stands out for its ease of use and precision. Consequently, creating illustrations has become an accessible endeavor, even for novices without any specialized skill.

Many practical applications of the "text-to-X" model using neural linguistic programming are currently being devised and implemented. The "text-to-image" model [51] [52] has already achieved highly accurate image generation and style transformation. As a natural extension, "text-to-video" [53] is already being presented in simple applications. Similarly, "text to motion" [54] is also available. The most pioneering example is "text-to-mesh" [55], which produces 3D objects. Moreover, "text-to-avatar" [56] already exists in photo-realistic fields. But there have not yet to produce avatars with video game quality that is the participant of this research. Avatars used in video games typically consist of between 10k and 100k polygons. Given that they can be observed closely from all angles and can be manipulated into any pose by the player, it's essential that they maintain a watertight mesh shape at all times. In addition, the bones must be rigged so that there is no collision or unnatural expansion when it mirrors the user's motion. Therefore, the performance requirements are extremely severe. However, I cannot deny the possibility that in the near future, anime-like avatars will be generated simply by a combination of few simple prompts.

2.1.6 Structure of 3DCG

The advancements in deep learning for two-dimensional images have naturally influenced the field of 3D. Generation of 3D data using deep learning is one of the hottest field of the CG research. However, the progress in this area is not as advanced as it is for 2D images. The main reason for this discrepancy is that the "topology" in 3D data is not unified.

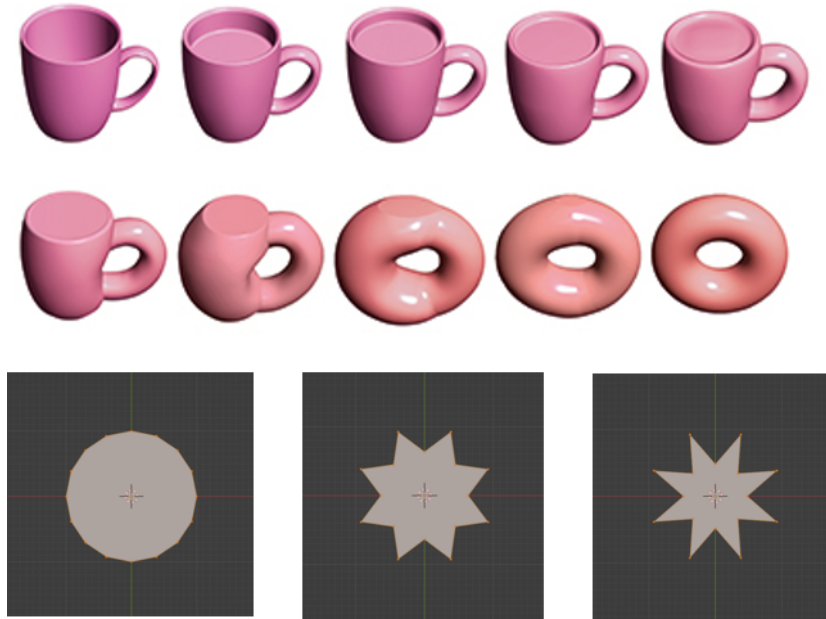


Figure 2.4: The top and middle row are **cited by <https://cems.riken.jp/jp/laboratory/qmtrt>**, and show shapes in the same mathematical topologies. A teacup could deform into a donut while only expanding and contracting. The bottom row shows data in the same 3D topologies. All data have 16 vertices and the same order so the left deform into the right while only moving vertices.

At this subsection, it is crucial to clarify the concept of "topology" which is a key point in this research. Topology is a term in mathematics used to categorize shapes based on their fundamental essence of continuity. It essentially encapsulates the invariant structure of shapes, unaffected by scaling or relocation as shown in Figure 2.4 the top and middle row. In the context of 3D computer graphics (3DCG), topology refers to the count of vertices and their arrangement as shown in Figure 2.4 the bottom row.

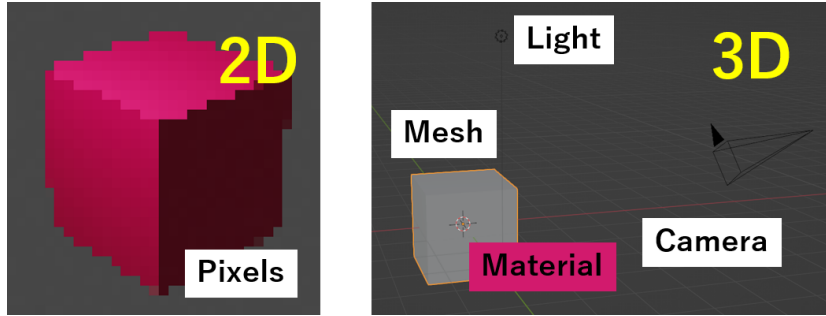


Figure 2.5: The difference of data structure between 2D and 3D. In 2D, images are composed from arrays of pixel that are formulated as $f(x, y) = (r, g, b)$. In 3D, images are rendered from separated data sources as Mesh, Light, Camera, Material. Each source has each dimension and variable length.

In 2D computer graphics (2DCG), if images are of the same size, their data length is fixed by the pixel count and the number of colors. This fixed structure is extremely convenient for multivariate analysis. However, 3DCG does not follow a data structure based on the basic unit of this type of space (known as voxel). In 3DCG, the shape is specified by dictating the positions of vertices and their sequential connections. Even if the data appears identical in a visual way, the density of the composing vertices may differ. Additionally, even if the count and position of vertices are the same, their connecting order may differ (Figure 2.5).

Naturally, the topology is not always the same among multiple data sets, and the structure is individual to each data set. This makes it difficult to apply multivariate analysis such as PCA or deep learning as is. In other words, since it is assumed that the topology of 3D data is not unified, a brand-new method to analyze across different topologies is needed.

2.1.7 Analysis for 3DCG

In this subsection, I discuss methods to analyze 3DCG in a topology-independent way.

The most naïve method involves conducting an analysis based on voxels (unit cubes) in 3D space [57] [58] [59] [60]. This approach is an extension of pixels in 2D, making it easily interpretable to humans and simple to implement in deep learning networks. However, voxel-based techniques encounter a limitation in achievable resolution, as spatial memory grows at an order of $O(n^3)$, increasing exponentially. In addition, mesh formation by voxels is fundamentally unsuitable for the natural representation of smooth objects because their normals are orthogonal to the lattice axis. So, replicating voxels

at a granularity that appears natural to the human eye requires an unrealistically extensive computation. Current methods tend to be voxel-based but incorporate strategies to reduce computational order.

If the objects to be generated are similar, for example, limited to human face[61] or human body[62], the method of fixing the template mesh and looking only at the transformations between them is powerful. However, this method only works well for objects with simple topology, such as cubes-like or spheres-like shape, but as the geometry becomes more complex, there are limits to its expressiveness and completeness. That is, either the template becomes overly complex and loses generalizability, or it cannot be guaranteed to be water-proof (with no hole). There are also advanced methods that estimate the original mesh based on the training data [63] [64]. There is also PolyGen [65], a combination of such methods and state-of-the-art deep learning that enables analysis by modeling the mesh generation procedure using transformers, but these applications are limited to simple objects such as chairs and airplanes. These methods using deep learning are hard to be applied to complex models like human avatar.

Another typical 3D format is the point cloud[66] [67] [68] [69]. Because of their high affinity with data obtained from depth sensors such as LiDAR, point cloud-based techniques have performed well in applications such as topographic map or urban models reconstruction [70]. In addition, point cloud-based techniques are efficient in terms of memory savings and can depict objects with extremely simplified data volume compared to mesh or voxel data. However, point clouds do not have mesh information. Therefore, the technique cannot be applied to avatars, for which mesh and rig information is essential. Therefore, it is necessary to develop a technique to estimate them from the point cloud, but it is not an easy task to generate a watertight mesh from the point cloud without any breakdowns. In other words, this technique has advantages in terms of generating objects that are seen from a distance, such as background objects, but its application to avatar generation is difficult.

There are also other challenging factors in the analysis of 3D data. In standard rendering, information such as vertices and meshes lose all continuity through the rendering process. As a result, it becomes theoretically impossible to compute backwards the original parameters from the two-dimensional rendered image. This has made it difficult to apply multivariate analysis. However, in recent years, methods for analyzing the back propagation of parameters from the rendering results by estimating intermediate parameters are being established. The beginning of such methods is a technique called Neural Rendering [71] [72]. It ingeniously defines the relationship between the rendering result and the mesh as pseudo-continuous

Table 2.1: The summary limitations of 3D analyses

Method	Refs	limitations
Voxel Based	[57] [58] [59] [60]	need many computational memory
Mesh Based	[61] [62] [63] [64] [65]	low variations
Point Cloud	[66] [67] [68] [69]	lack of geometry information
Implicit Function	[73] [74]	lack of detail
Neural Rendering	[71] [72][75] [76] [77] [78]	lack of detail

by adjusting the rasterization mechanism. This allowed the application of deep learning to the 3D data and rendering results, which had a discontinuous relationship. There is also research that signed distance functions (SDF) [73] or any other implicit form [74]. By learning the entire shape as an implicit function, the point where the positive-negative changes is formed into a surface. Implicit functions serve as topology-independent parameters, which makes them suitable for unified polynomial calculations. However, the application of these methods is limited to the reproduction of simple models like chairs or airplanes. Implicit methods require finer sampling and more parameters to describe uneven and highly variable shapes. So, the current methods are not expressive enough to reconstruct anime-like avatars of 10 K or more high polygon with complicated shapes like hair, finger and deformed face parts.

Currently, techniques for estimating parameters from rendering results, such as NeRF [75] [76] [77] [78], are evolving and becoming generalized. This technique estimates the parameters underlying volume rendering by considering the position and angle of the camera transforms from the rendering results obtained from multiple viewpoints. With this method, it is possible to reproduce the surroundings, landscapes, and room conditions from a few portraits taken with a smartphone [79]. However, due to the nature of volume rendering, it is very powerful at reproducing 3D objects at large scales. On the other hand, it is not suited to reproducing the detail of small objects. In other words, it is impossible to obtain a high level of detail by applying such a technique directly to an avatar in an anime-like style.

The summary of the above limitations are shown in Table 2.1.

2.1.8 3D Avatar generation

When it comes to methods for generating 3D models of faces, there has been a considerable amount of research in the long history of reconstructing 3D faces from single images. This is, for example, like estimating a photo-realistic 3D model from a self-portrait. These methods can estimate bones

or landmarks from a picture with high accuracy [80] [81]. However, the model reconstructed from the bone information is photo-realistic, and it is impossible to produce the various features of an anime-like avatar.

Of course, there are some synthesis methods targeting 3D avatars, especially those in anime-like style. However, anime-like avatars incorporate cartoon-like expressions in their faces, their bodies are often deformed, and their variations are wide-ranging and often discontinuous. As a result, the parameters and resolution needed to express fine details are more extensive compared to photo-realistic avatars. Because of these technical limitations, research attempting to automatically generate anime-like avatars has been limited to those that fix the basic model [82]. Therefore, a mechanism that can extract features while preserving the wide range of detail in various models has not yet been established.

A method has also been established to automatically generate avatars in anime-like style by using machine learning to perform fitting to match the training data with an anime [83]. However, the range of application of such methods is limited because the variation of the generated results is limited to the range that can be created by the tool. No method has yet been established that can extract features from a wide range of features of various 3D models while preserving their details.

In conclusion, even with machine learning techniques, robustly analyzing complex shape variations of anime-like characters is challenging, and it's necessary to establish methods capable of accomplishing this.

2.2 VR device and Interface

One of the major advantages of using VR devices is the ability to treat the VR controller as an input device. This has great potential, unlike conventional interfaces, as it allows 3D space to be perceived as the real world and enables intuitive handling of physical inputs. However, there are new drawbacks to its use that are not present in conventional devices. In this section, I will discuss advantages and disadvantages of VR interface.

2.2.1 VR and Tracking

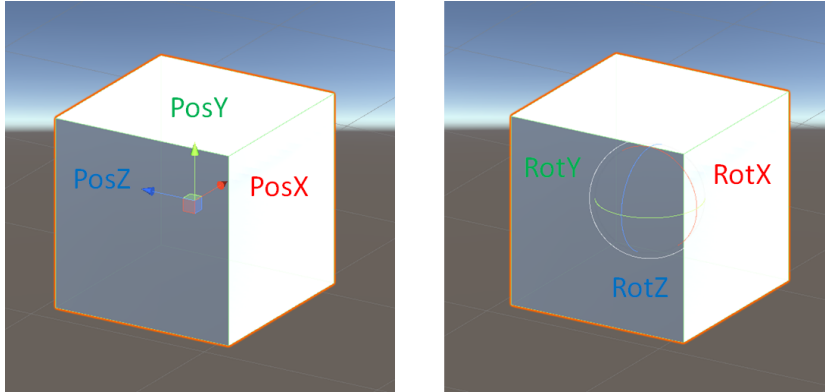


Figure 2.6: Image of 6 Dof. The left image shows 3 Dof of positions. The right image shows 3 Dof of rotations (Euler angles).

How user input is reflected in the VR space is also an essential work of VR. VR technology has a wide range of applications in education, cognition, medicine, and entertainment. But, it is not easy to accurately reproduce the real world in VR space.

One predominant problem is the difficulty associated with accurately detecting users' motions. This process of detecting and replicating user movements in the VR space is termed as "tracking." The data obtainable from VR devices are limited. A majority of VR devices are equipped with a HMD and controllers for each hand. Typically, the data from this tracking yields 6 degrees of freedom (DoF) for each device, effectively measuring a three-dimensional position and three-dimensional Euler angles (Figure 2.6). Yet, for precise tracking, additional nuanced information is indispensable. This includes gaze direction, facial expressions, mouth movements, hand and finger gestures, and the intensity of a grip, among others. Such nuances are paramount especially for intimate communication. While auxiliary devices like LeapMotion and Animaze (FaceRig) facilitate such measurements, they necessitate advanced and swift image and pattern recognition (Figure 2.7). Achieving a level of fluidity that mirrors real-life communication often remains elusive.



Figure 2.7: Face tracking application "Animaze" also as known as "FaceRig"
<https://www.animaze.us/>

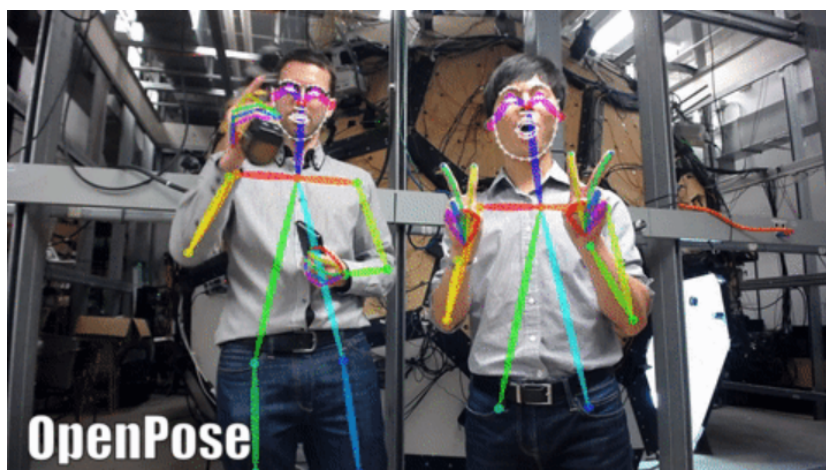


Figure 2.8: Openpose, pose estimation from video images. <https://github.com/CMU-Perceptual-Computing-Lab/openpose>

Estimating body pose presents another essential challenge. Similar to facial expressions, pose expression play a crucial role in communication. In technical instructions or dance, replicating the exact movement of the body becomes paramount. However, estimating the pose solely based on the position and angle of the head and hands proves unfeasible. As a potential solution, technologies similar to OpenPose [84] exist, which capture full-body images and estimate poses (Figure 2.8). Nevertheless, these too require substantial computational resources. Identifying a specific individual in video

image containing multiple participants and continuously detecting its pose without discrepancies is a highly sophisticated computational task. Even with the application of deep learning techniques, achieving flawless performance remains a hard challenge.



Figure 2.9: VICON, a dedicated motion tracking system. It uses a suit filled with markers and camera system.<http://www.vicon.jp/>

As a fundamental solution, there exists a method of attaching sensors to all the major joints of the body. When professional dancers undergo motion capture, they typically wear specialized suits adorned with markers all over their bodies. Additionally, equipment like VICON, Perception Neuron, which employs multiple physical sensors for full-body tracking, is also available (Figure 2.9). However, such equipment is often expensive and requires dedicated studio spaces, making them less accessible for non-professional user. Some additional sensors to the waist and legs, aim to provide full-body tracking data. Yet, the raw data from these devices alone cannot accurately represent a full body tracking. These devices employ a technique known as "inverse kinematics" which calculates joint positions from the end point and estimates the pose using a minimal set of tracking data. However, they come with technical challenges, including potential computational errors that may distort the captured posture and calibration drifts resulting from accumulated inertia.

Method	Acc.	Eco.	Speed.	Examples
Camera + Image Recognition	×	○	△	OpenPose
Sensors + Estimation	△	△	○	HTC Vive Tracker
Dedicated System	○	×	○	VICON, Perception Neuron

Table 2.2: The advantages and disadvantages of tracking methods. "Acc" means accuracy, "Eco." means economy (lower cost).

In essence, whether employing image recognition, setting up a dedicated studio, or combining a few sensors with kinematics estimation, each method has its own set of advantages and disadvantages (Table 2.2). Separate from the characteristics of each method, there exists a trade-off in terms of which body parts require how much precision. For tasks like typing sentences, the movement of the fingertips demands extremely high sensitivity, with even a split-second interruption being unacceptable, though the measurement area remains confined. On the other hand, when detecting dance movements, millimeter-level precision might not be necessary, but the area of measurement becomes vast. In real-world interface design, beyond the trade-offs related to costs, considerations must also be made balancing the performance characteristics of devices with the actual measurement target.

2.2.2 VR and User Interface

Designing user interfaces in VR spaces is a critical and challenging issue. In a VR environment where one can physically manipulate the 3D space directly, actions such as throwing objects, striking, or swinging down controllers can be replicated exactly, allowing for extremely intuitive operations. On the other hand, continually tracking the absolute positions with high precision is difficult. Moreover, actions directly linked to bodily movements are limited in their degrees of freedom, making it challenging to replicate an efficient UI, like densely packed buttons on a keyboard. While there are methods to pseudo-replicate such devices in VR space using a relatively wide area, these often result in crude operations using laser pointers, making it hard to perform actions accurately without stress.

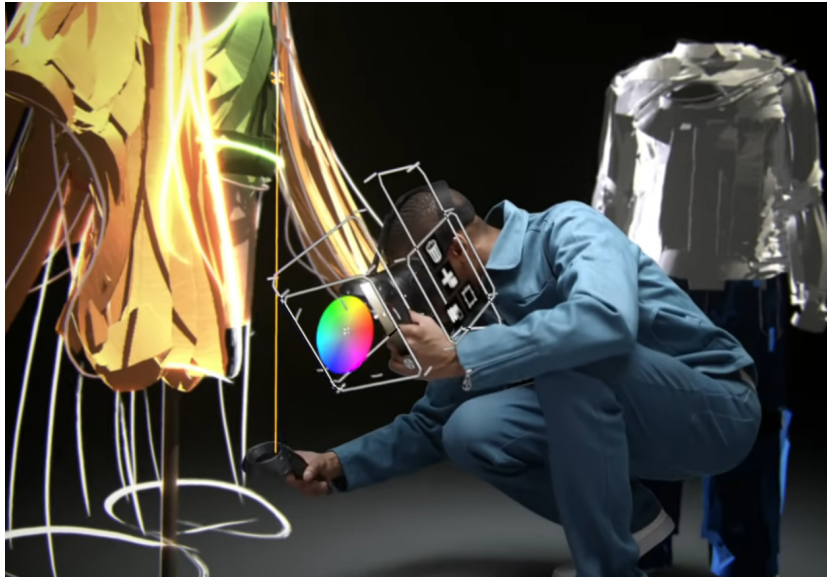


Figure 2.10: Tiltbrush is a drawing software in 3D VR spaces. The user can draw a line anywhere in the air. <https://www.tiltbrush.com/>

Another advantage of VR interfaces is that they free users from traditional monitor-type operations. Basically, current modeling software requires users to face a 2D monitor. However, working in this position is limited to work such as that of an illustrator. On the other hand, when creating sculptures in an atelier, the work requires more physical moving, which is not possible with a 2D monitor. However, by using VR equipment, it is possible to realize such a work environment. TiltBrush (Figure 2.10) is a pioneering example of a drawing software that uses such input, but it is still limited.

2.2.3 The Possibility of Gesture Input

Taking advantage of VR's merit of enabling physical input, interfaces using gesture-based input have also been devised. Movements such as stretching and throwing make it possible to transfer physical movements directly to the avatar, making it an extremely intuitive interface. However, this method requires the player to reproduce the movements required of the avatar. Walking and jumping require a great deal of effort, and in the first place, flying and somersaults may be physically impossible for the user to perform. For this reason, interfaces have been devised that use physical input but efficiently transcribe part of it into other movements. Examples of such research include methods that magnify the user's body movements and transmit them to the avatar [85], methods that use the avatar as a puppet and manipulate

it simulatively through finger movements [86], and methods that combine these elements[87].

In other words, when using physical input, it is necessary to consider whether it is appropriate to use the input as it is as a parameter. In the case of primitive movements such as stretching, throwing, etc., this is often appropriate, but in other cases, it is necessary to convert the physical input into a different output. This requires arbitrary adjustments, such as mapping finger movements to puppets, or mapping a specific bodily movement to another specific movement. There are still many unanswered questions as to what kind of mapping is appropriate, and this is an area where research is desirable.

2.2.4 The Disturbance of VR Device Itself



Figure 2.11: The image of HMD device "VIVE Pro". The size of HMD are large over the user's head and the controllers's shape are like sticks. <https://www.vive.com/>

In addition, the weight and size of the VR device itself cannot be ignored in the current situation. Currently, HMDs such as the VIVE Pro and Meta Quest are fairly large objects, and users cannot ignore their presence and disturbance. Wearing them requires a clear change of motion, and it is difficult to wear them as naturally as with glasses. This is especially true for controllers, where the VIVE controller is shaped like a large stick(Figure 2.11). The movement and fixation of such a large object in the air at various speeds with no support is unlikely in everyday life. Compared to many traditional

mechanical devices (paper and pen) that have evolved to reflect the precise movements of the user without stress, many VR devices are unnatural in both their hardware size and the movements they require of the user, and it must be said that many of them place a burden on the user.

2.2.5 Absense of Feedback

One of the reasons for the difficulty of continuous value input on VR devices is that it is difficult to have rich feedback as well. For example, when drawing a shape with a pen and paper in the physical world, the user can constantly recoil from the paper while knowing and fine-tuning the exact position of the pen tip. Also, when a user is typing on a keyboard, each time a key is pressed down, the user receives a recoil from the device, which helps the user know where the fingertips are located by tactile sensation alone. While many users can continue keyboarding without looking at their fingertips at all, not many would be able to do so without this kind of mechanical feedback [88] [89].

Such feedback is not only used for fine-tuning by the user, but is also involved in the very foundation of the interface itself. For example, when creating an interface for playing drums, it is essential to have feedback that allows the device to recoil back to its original position after the drum is struck. If there were no such recoil, the user would have to hold the device still in mid-air and then return it to its default position each time the drum is struck.

As noted, repeated stop-and-go cycles are very taxing on the human joints. Interfaces such as buttons and instruments alleviate this burden by automatically generating mechanical repulsion on the device side. An interface that requires the user to go to the trouble of simulating a stop-and-go would fundamentally undermine the quality of the drumming experience. Thus, feedback is essential for a user interface that is easy to input. However, it is physically difficult for a VR device alone to provide such feedback.

In response to this problem, there are methods to reproduce mechanical feedback by attaching an auxiliary device to the user [90] [91] [92], or to make the device itself more intuitive in terms of hardware. The latter, called tangibles interfaces [93] [94] [95], devise device geometry and feedback so that operations applied to the device are more intuitively linked to the operations one wants to perform (Figure 2.12, 2.13). However, these methods are specialized for use when the feedback to be expressed is clear, and there is currently no universal feedback reproduction method that can be used for any VR device.

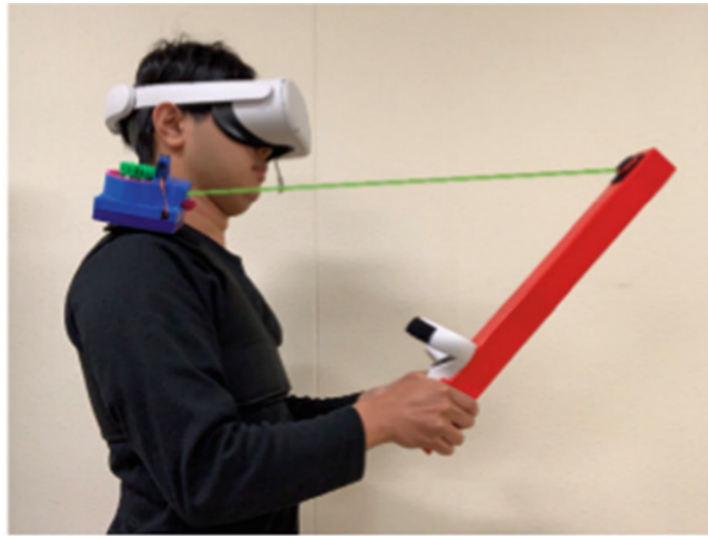


Figure 2.12: Cited by [92]. Haptic feedback device that emulates the recoil impact of swinging.

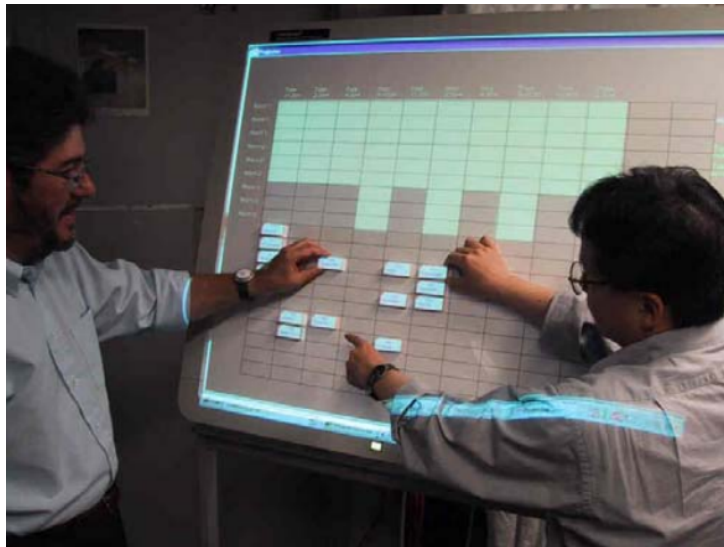


Figure 2.13: Cited by [93]. A sample of tangible interface. The mechanical form of the interface itself works as a system and can be touched and moved.

While VR currently serves almost perfectly as a display presenting visual and auditory information, as an interface for users to interact with the external environment, its functionality remains somewhat lacking.

2.3 Relation to This Study

The issues derived from related studies can be summarized as follows.

- There are few examples of automatic generation of anime-like avatars.
- There are lack of training data and the difficulty of analyzing 3D data with different topology.
- Although analysis techniques using deep learning are being developed, it is difficult to generate avatars with video game quality.
- Conventional character-making systems are too time-consuming.
- Interfaces using VR could be a great solution but is not yet unexplored enough.
- There are difficulties unique to VR such as the absence of feedback and the difficulty of fine control.

so, my research need to make the following contributions

- General data set generation and analysis for anime-like avatars.
- Proposal of new VR interface prior to the conventional ones.
- Evaluation of the essential usability of the VR interface.

Chapter 3

Data Set Generation

In this chapter, I discuss techniques for generating the data sets needed to generate 3D avatars. As discussed in Chapter 2, these require high-quality training data and topology-independent feature extraction techniques from them. As a pilot study, I tried several techniques and then refined them to develop an generic data set generation method.

3.1 Training data

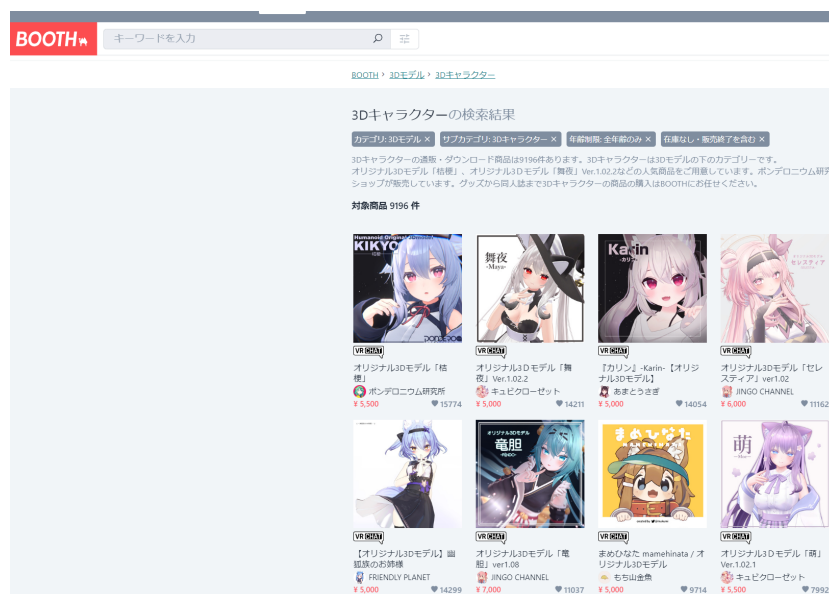


Figure 3.1: Major commerce cite "BOOTH", <https://booth.pm/ja>

Table 3.1: The training data used for data set generation.

sex	example	number
female		26
male		2
deformed		12

In this study, in order to address the image analysis problem, I aim to develop a method for extracting features from high-quality 3D models used as training data. Data sets are essential for machine learning. In the case of 2D images, a large amount of such learning data can be prepared from illustration data or photographs. Although some large data sets have been published in 3D, including many data of simple objects and photo-realistic humans [96] [97], the data for anime-like avatars is quite rare. However, with the recent development of metaverse culture, there has been a rapid increase in cases where users are selling 3D models (Figure 3.1), and many of which have video game quality.

Therefore, by using these data and extracting the details they possess, I aim to establish a system for automatically generating avatars. Due to the budget limitations, resulting in learning from a small amount of data.

In this study, I used 40 sets of data as training data. Although the sales data was overwhelmingly biased toward female avatars and it was difficult to find other types of avatars, a small number of male avatars and deformed avatars were added to the sales data in order to provide a wider range of generation. The distribution is shown in Table 3.1. The reason for the bias toward female avatars in the sales data is that they are less likely to give an aggressive impression and more likely to express emotions, which makes them more suitable for communication. In addition, most avatars were often equipped with objects that imitated cat or fox ears, either by default or as an option. Such attachments are called "kemomimi". The reason why kemomimi are preset on many avatars is thought to be because they are used to amplify emotional expressions in VR environments, where it is difficult to reflect real-time facial expressions [98] [99]. In this study, the

so-called human face, hair, and kemomimi were analyzed as separate objects.

3.2 Extraction of Data

The extraction of mesh and texture data from the training data is shown in Figure 3.2. First, the selling avatar data is loaded in Blender as training data. A complete avatar data has a structure of nude bodies, clothes, hair, and ornaments stacked structurally according to the bone hierarchy (Figure 3.2 A). Since the main target of this project is the face data, only the facial mesh was extracted from this data (Figure 3.2 B). First, all meshes other than the face were deleted on the 3DCG software "Blender."



Figure 3.2: The image of data extraction of the training data. Complete avatar data was load in Blender GUI (A). This image shows the extraction of facial parts. facial mesh was separated and saved as another mesh data file (B). Facial texture was obtained as orthogonal rendering result (C).

I applied another type of data extraction, and topology fixing processes were performed for the textures. The details are explained in the following sections.

3.3 Pilot Study

I conducted two pilot studies to discuss the versatility of the method for 3D data analysis.

3.3.1 Marching Cube Reconstruction

In pursuit of a topology-independent 3D reconstruction, I implemented implicit function learning using the Marching Cubes algorithm [100].

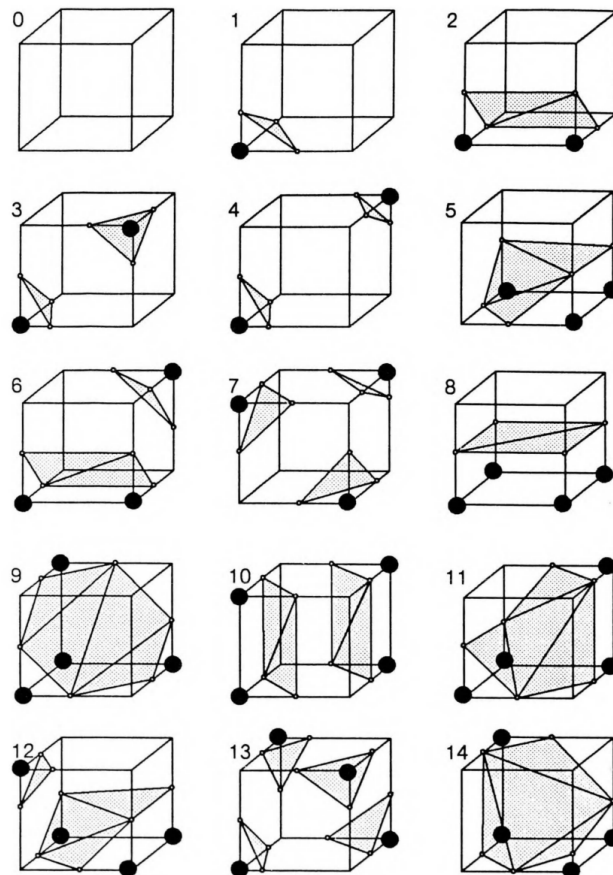


Figure 3.3: Cited by [100] The total combination of vertex states is nominally $2^8 = 256$, but after symmetrically reduced, 15 patterns are effective. Meshes are restored according to the vertex states as the figure shows.

The Marching Cube is a technique that allows mesh generation from implicit functions by dividing the space into several grid cubes. This cube, which has 8 vertices, presents two scenarios at each vertex: the cube is either inside or outside the mesh of the implicit function. This inside/outside decision is defined as inside if the scalar of the implicit function is 0 and outside if it is 1. An appropriate threshold is used to create a boundary surface that separates the inside from the outside. If symmetry is not considered, there are $2^8 = 256$ possible combinations of vertices, but if symmetry is considered, it is limited to 15 possible combinations. By defining the mesh cross

section to be restored for those 15 combinations as shown in Figure 3.3, I can form a mesh that is the boundary surface from the implicit function. There are many applications in fields such as the illustration of organs in medical technology [101] [102].

Next, consider how to estimate the implicit function from the training data. For some training data, the space is divided into several grids. The finer this division is, naturally, the better the accuracy, but in this study, the space is divided into 256^3 cubes, 256 for each direction, in order to balance the computational cost. Next, for each vertex, SDF was calculated from the training data. The SDF is expressed as the distance from the nearest mesh, negative in the interior direction and positive in the exterior direction. The distance calculated in this way becomes an implicit function that returns a scalar value in three-dimensional coordinates, and the surface whose scalar value transitions from negative to positive represents the boundary surface of the training data. The distance functions were calculated using the SDF solver included in the libigl library¹.

Since these implicit functions share the same grid delimitation method, the synthesis of the implicit functions themselves can be achieved by linearly combining the distance functions for each coordinate. The implicit functions synthesized in this way can also be transformed into actual meshes by the marching cube by applying appropriate normalization. Using this method, I have shown that the marching cube can be used to restore the model A, which is the training data, to model B, which is another training data. Furthermore, a new model can be generated by combining them.

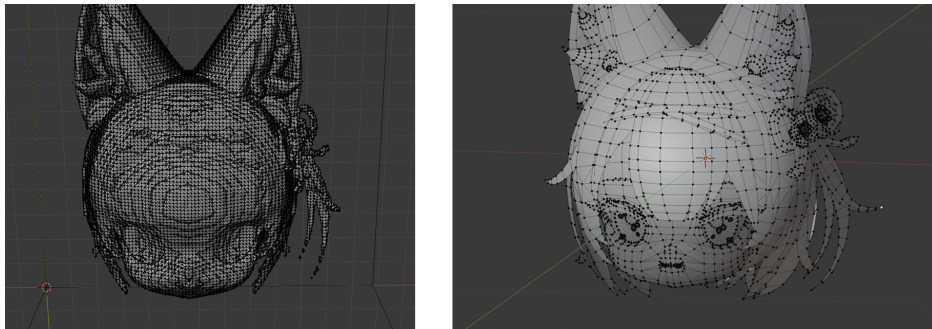


Figure 3.4: The result of reconstructing by marching cube method. The left is the reconstructed data and the right is training data.

Figure 3.4 displays the reconstruction results. The left side shows the reconstruction outcome using Marching Cubes, while the right side presents

¹<https://libigl.github.io/>

the ground truth. Although the mesh has been restored using Marching Cubes, the results are topology-independent. Consequently, these figures can transform in shape gradually by interpolating distance functions from one figure to another.



Figure 3.5: A sample of morphing by marching cube method. The left is a simple cube and the right is the reconstructed data.

Figure 3.5 serves as a sample in point, demonstrating the transformation from a simple cube to a learned face. Additionally, Figure 3.6 showcases another example, depicting the transition from a learned face A to another learned face B. In this way, while the method using Marching Cubes succeeded in replicating the faces from the training data to some extent, several challenges were evident.

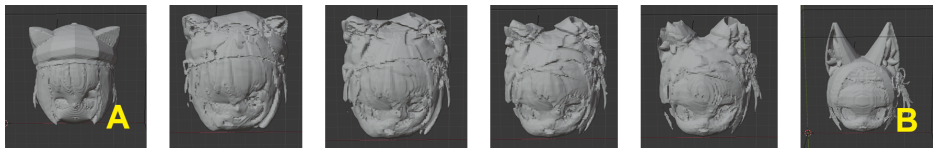


Figure 3.6: A sample of morphing by marching cube method. The left is the reconstructed data A and the right is another reconstructed data B.

It was notably difficult to depict intricate areas like hair, and frequently, defining such areas uniformly as a distance function led to morphological distortions. As a countermeasure, a method that separately analyzed sections was contemplated (Figure 3.7). However, in areas with pronounced features like eyeliner or the mouth region, a phenomenon occurred where details got lost. To prevent this, an increase in grid resolution was necessary, but this led to an exponentially increased computational order, proving to be virtually impossible.

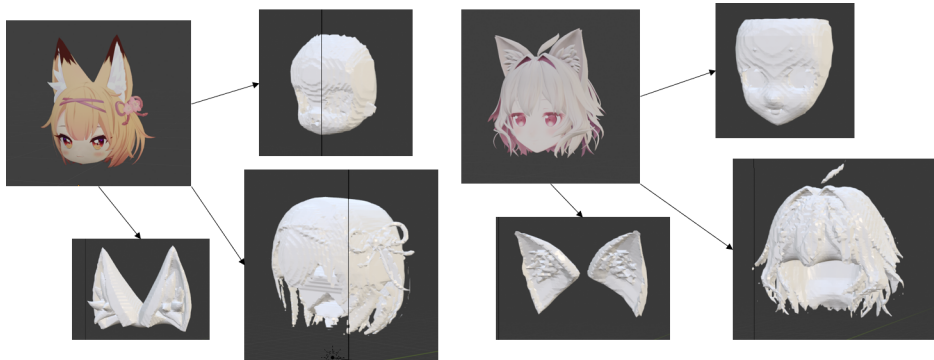


Figure 3.7: Reconstructed results using marching cube method by part. Each training data is separated to parts (Ear, Hair and head).

3.3.2 Manual Landmark Morphing

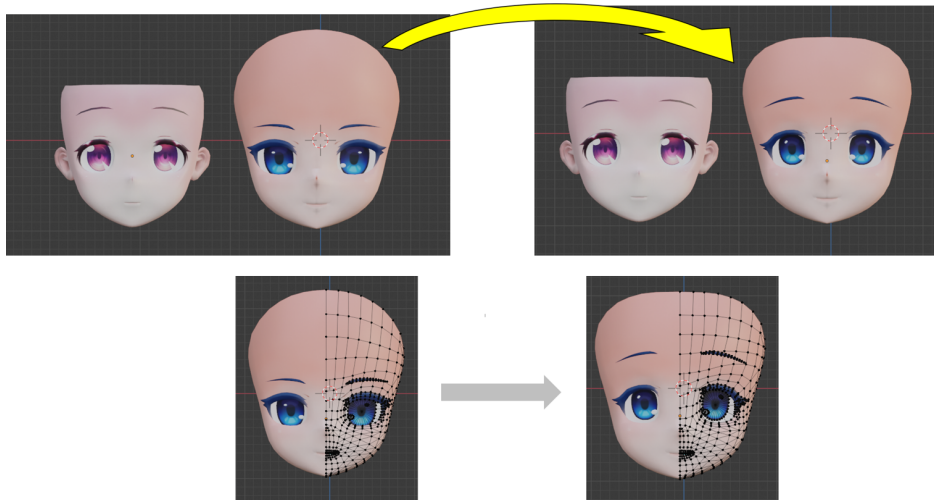


Figure 3.8: Landmark setting and approximation in manual landmarking method. All vertices of the base mesh were approximated to another training data by manual work.



Figure 3.9: Variations generated by manual landmarking method. Variations are reorganized by PCA.

Another approach was to fix the topology in the first place. Specifically, the method uses one set of training data as a reference, transforms it to match other training data, and records it as a shape key to extract the facial features of the avatar. This is a primitive mesh based method that fixes the template, but since the target is limited to the avatar's face, the fact that the generated results are bound to the topology of the template is not so much of a drawback. By fixing the template and using facial feature points as landmarks and matching these landmarks, morphological differences between the training data were recorded as shape keys.

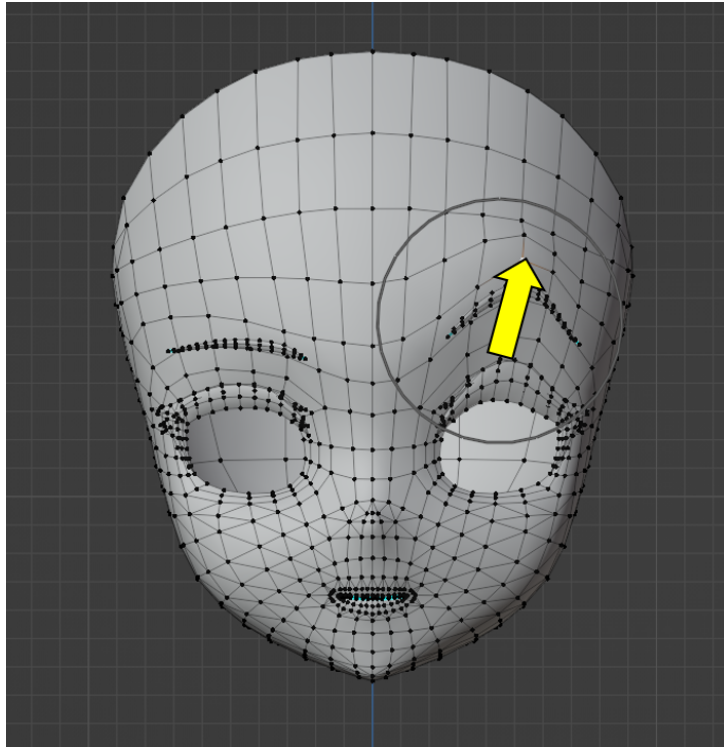


Figure 3.10: Proportional editing. When one vertex is moved, vertices in a predefined range move with it, and the movement is proportional to the distance from the selected vertex. The circle means the limit where the weight of editing gets 0.

The training data had roughly 2,000 vertices on average. This is an example where it would be nearly impossible to adjust everything manually, but if the landmarks are aligned by proportional editing (Figure 3.10), the deformation can be done with some accuracy. I did this by taking one of the models in the training data and deforming it to match the faces in the other training data (Figure 3.8). This deformation process was done manually in Blender and took about 30 minutes per data.

As a preliminary study, once about 10 shape keys had been set, I analyzed the data to confirm the accuracy of this method. For every shape key, a PCA was performed on its variation. These components effectively account for the variance of the faces in the training data. Figure 3.9 illustrates the difference generated by PC1 and PC2. The vertical direction represents PC1, which corresponds to the width of the face. The horizontal PC2 direction shows the difference in eye position, indicating that even with a small amount of sample data. It shows this approach succeeded to extract factors that successfully

explain the facial differences in the training data.

However, this method is not feasible for processing large amounts of data, as it takes a lot of time to adjust the landmarks. In addition, a more essential factor is the human blurring of fine adjustments, which makes the consistency of the method questionable.

From these pilot studies, I found a trade-off between the analysis of large amounts of training data and the preservation of detail. Naive analysis methods need computational power significantly, while methods requiring extensive manual adjustments fail to maintain speed and objectivity. Therefore, I found it necessary to develop an efficient method that combines both for effective data set generation.

3.4 Subdivision Shrink Method

As mentioned above, pilot studies have shown the need to establish a method for automating the learning of details of shapes including anime-like elements (exaggerated eyelines, cheek lines), while taking into account a certain degree of manual settings. In this study, I propose a method called the "Subdivision Shrink" to unify the differences in topology between models. This method involves adjusting the position of each facial part at a low-polygon level to match the training data, and then repeating the high-polygon transformation and approximation to reproduce the detail of the training data while maintaining a common topology.

The flowchart of the method is shown in Figure 3.11.

First, I carry out a process known as template matching for both the template and the training data. A low-polygon face template is prepared, which is set up with a common topology using a small number of vertices to match the shape (face or body) I want to learn. For this template, 90 landmarks (including the nose, mouth, chin, cheek peak, and eyeliner) are set and aligned to the corresponding points of the training data (Figure 3.12). This approximation was done manually.

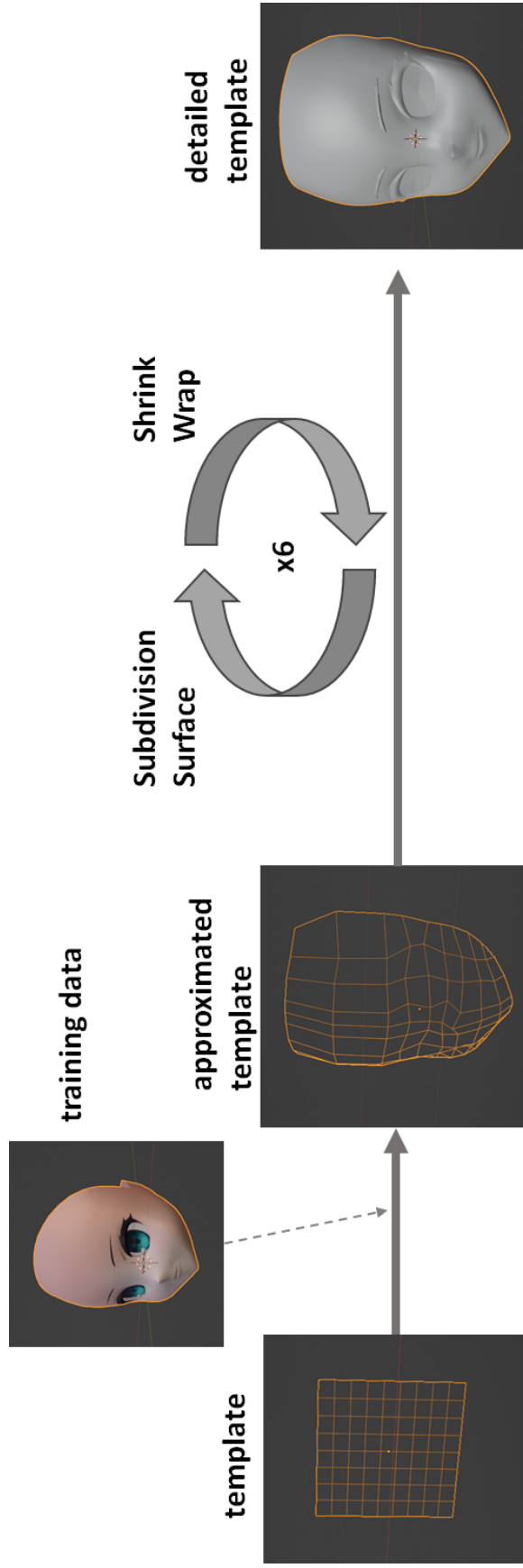


Figure 3.11: Flow of the Subdivision Shrink method. First, I set a common low polygon template and approximate it to the training data. Then, by iterating subdivision surface and shrink wrap 6 times, I got a template with the details of the training data.

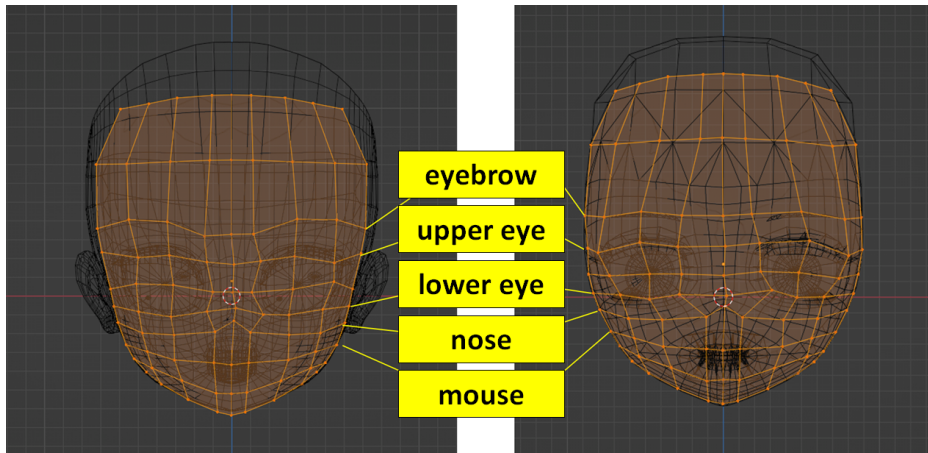


Figure 3.12: Template matching between different training data. The 90 landmarks of the template are brought closer to the corresponding points in the training data. Landmarks consist of lines that pass through each part of the face. The figure picks up lines through the eyebrows, upper edge of the eyes, lower edge of the eyes, nose, and mouth.

Subsequently, the template underwent subdivision surface and shrink wrap.

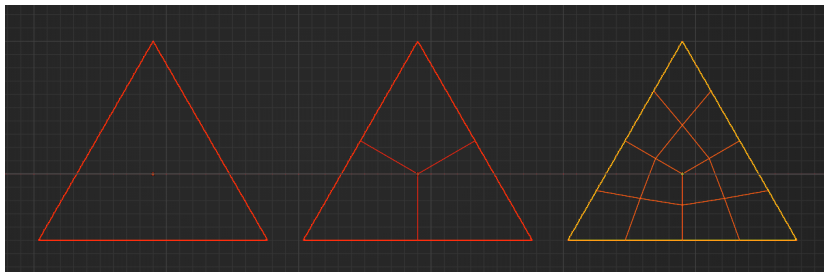


Figure 3.13: The specific process of subdivision surface. Each single polygon surface is separated into pieces recursively. The left is the original mesh. The middle is the $n = 1$ iteration. The right is the $n = 2$ iteration.

Subdivision surface is performed to enhance the expressiveness of the template, and the recursively divided polygons can conform to the training data at a more detailed scale. Here is the specific of process of subdivision as Figure 3.13. The original mesh is separated into pieces recursively. The rule is decided to any triangular polygons and quad polygons (Theoretically, all shapes can be represented by combining triangular polygons, but quad polygons are given special treatment because of their better usability for intuitive processing) as the following,

- Make new vertex at the barycenter position.
- Perpendicular lines from that vertex down to each edge of the original polygon

Subdivision surface process usually requires smoothing, such as Catmull-Clark method [103], but in this method, smoothing process rather undermines the objective of accurately tracking the training data, so a simple subdivision was performed as shown in Figure3.13.

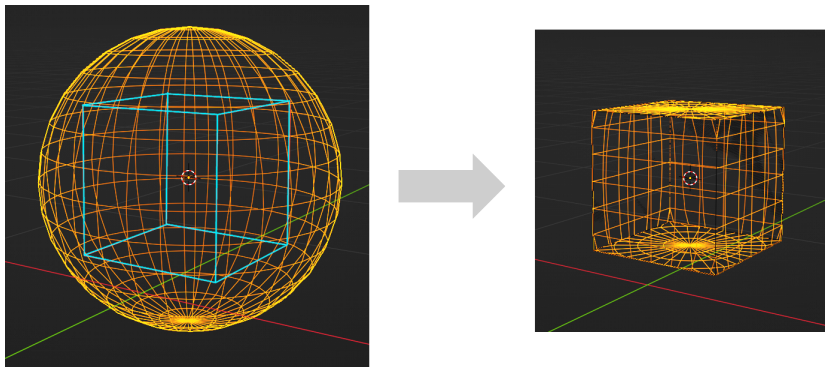


Figure 3.14: The sample of the process of shrink wrap. The polygons of the sphere (yellow) is fit to the cube (lightblue) keeping the same topology.

The shrink-wrap aligns the polygons of the template to the training data, with each vertex of the polygon being moved to a position closer to the surface of the training data. In this process, the vertices of each polygon of the original data are moved so that they fit on the target data mesh according to the normal direction of the original mesh. The sample of the process is shown as Figure 3.14.

By repeating this process combining subdivision surface and shrink wrap, I can reproduce the details of the training data while maintaining a common topology.

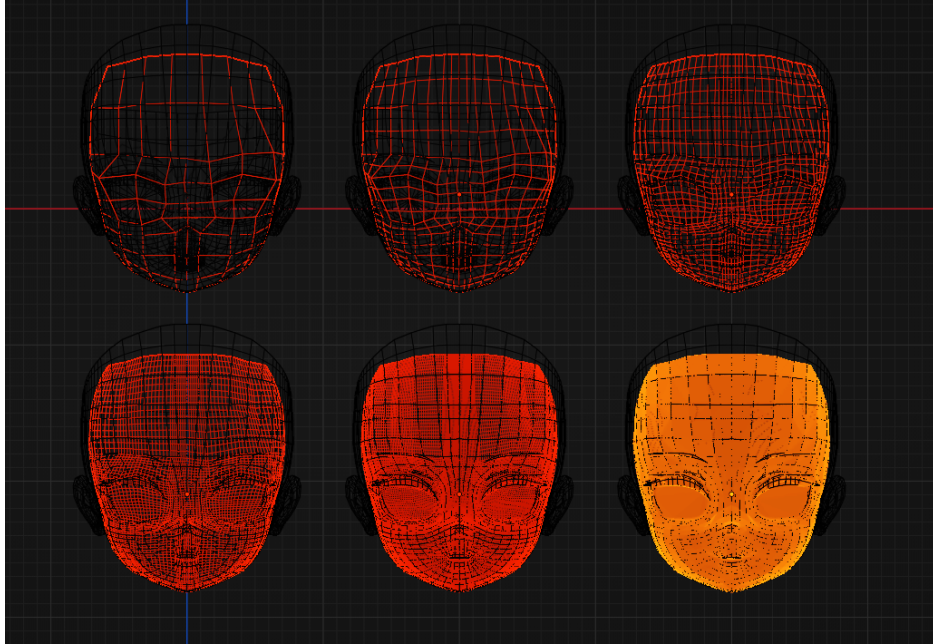


Figure 3.15: Procedure of subdivision shrink. The first template is the top left one and the 5th one is the down right. The template mesh gets high-polygoned learning the details of the training data.

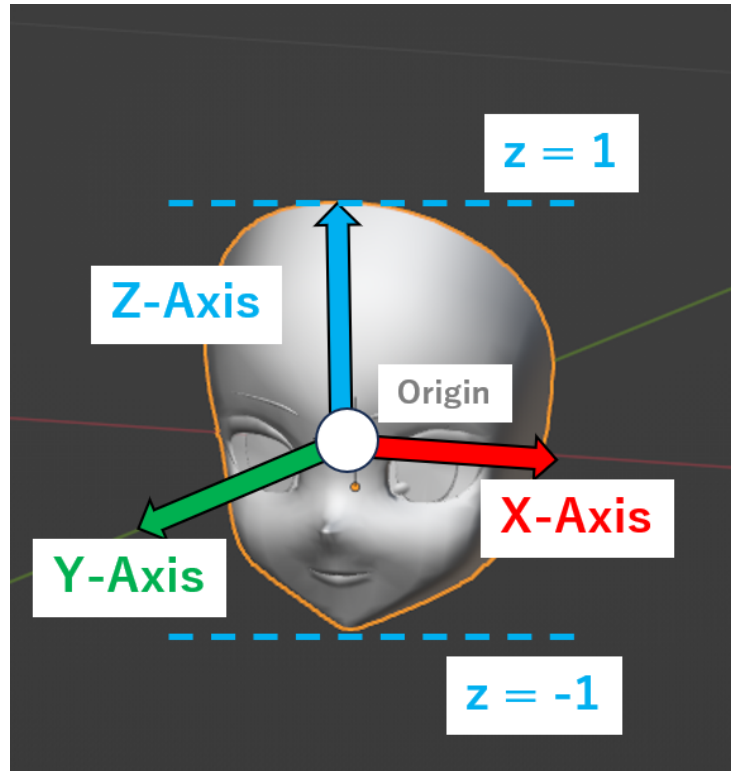


Figure 3.16: The normalization process. X,Y,Z axes mean horizontal, depth, vertical. X,Y axis are normalized so that their means set to 0. Z axis are min-max scaled so that their value range is -1 to 1.

Since the size of the face and the height of the forehead can vary by model, mesh scaling processing was performed. In the vertical direction (z-coordinate), it was scaled so that the lower and upper vertices' coordinates matched -1 to 1. For the depth (y-coordinate) and horizontal (x-coordinate) directions, an affine transformation was performed such that the average of each vertex coordinate is zero (Figure 3.16). Normalization process were performed as following:

$$\begin{aligned}
 s &= z_{max} - z_{min} \\
 x' &= \frac{x - \bar{x}}{s} \\
 y' &= \frac{y - \bar{y}}{s} \\
 z' &= \frac{z - \frac{z_{max} + z_{min}}{2}}{s}
 \end{aligned}$$

where bar indicates each mean of the coordinate.

With each subdivision surface, the number of vertices doubles exponentially. After several operations, memory limits can be reached. However, in this study, it was determined that six iterations (increasing the template’s vertex count from 90 to 296,001) provided sufficient accuracy in reproducing the training data.

The operations of template matching and iterations were carried out using the 3D software, Blender. Template matching was done manually, while subsequent iteration processing was conducted using a Python script within Blender.

3.5 Learning Texture Data

I developed a topology-independent extraction method for texture images. By editing the two-dimensional correspondence map (UV map) that dictates where the texture image is transferred on the mesh, I performed a process to standardize the topology.

To achieve this, I first prepared a fixed UV map as a template. Separately, it is necessary to extract textures from the training data. UV maps linked to the training data each have their uniquely adjusted UV maps, each inherently with its unique topology. Hence, direct learning from them is not feasible. As such, it becomes necessary to deconstruct the unique topology of the training data and perform operations to fit it into a template with a standardized topology.

Initially, I captured a data set for training using orthographic projection, creating the original portrait image (Figure 3.2 C). I loaded the training data in Blender and performed rendering. The material settings were configured in a mode without lighting, and the camera was fixed facing forward with orthographic projection. This yielded rendering results unaffected by shading. Although this resembles a simple facial photograph, I deformed the UV map of the template mesh and manually aligned the landmarks of the UV map with the landmarks of the original texture. This allowed for standardizing the topological differences each training data set held (Figure 3.17).

Here, the reverse procedure of the transformation that was performed on the UV map is applied to the portrait image. In other words, the inverse transformation was performed on the portrait image such that the UV map was transformed back to the template mesh. This resulted in an image that was transformed into the shape of the template mesh.

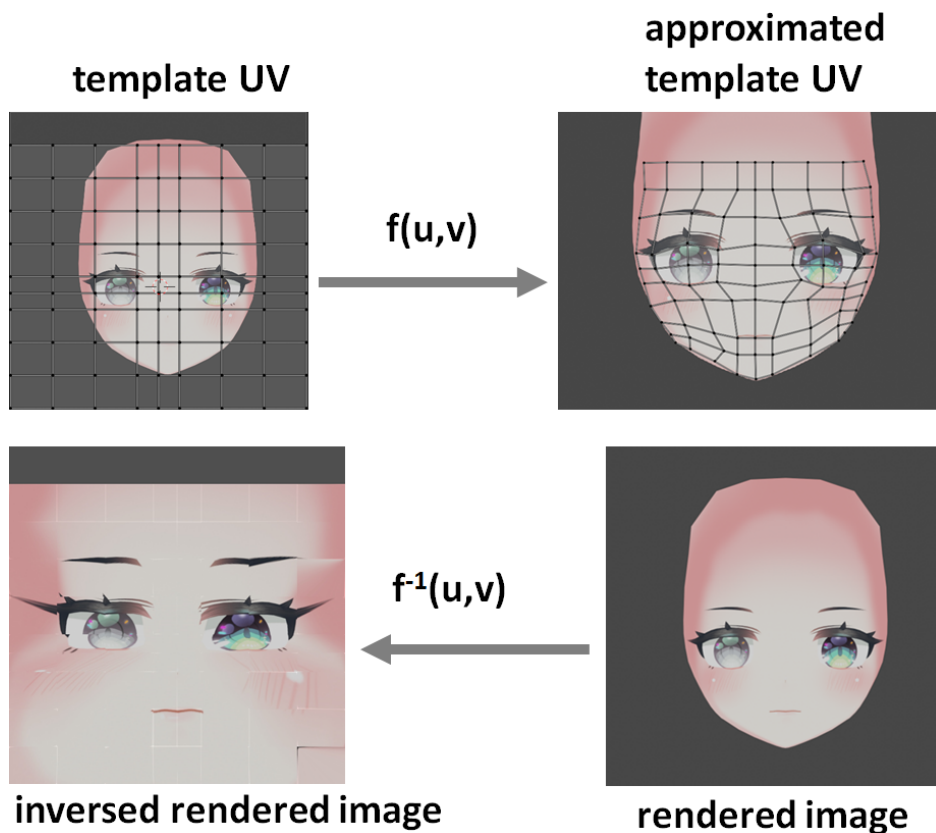


Figure 3.17: Inverse transformation of texture image. Transform the template’s UV map (upper left) to match the target image (lower right) obtained by parallel projection rendering of the training data (the result is upper right). Then, by performing the inverse transformation on the target image, I get a unified texture image in the form of template UV. (lower left)

Through this operation, the texture information, which was fixed to the dedicated UV maps in the training data, was uniformly transferred to the template, resulting in a common topology. By this standardization, multivariate analysis can be performed as image data with a common topology.

3.6 Distribtuion of each data set

As a result of these data set generation, I succeeded in generating a data set that has a common topology while still learning the features of the training data. First, the obtained mesh data are listed in Figure 3.23. The meshes are extracted for all $n = 40$ data. Each mesh learns in detail the features of the respective training data, but since they have a common topology, they

can be freely transformed and distributed. Examples of these are included below.

Using the obtained mesh data, I demonstrated that mesh transformation from one model to another can be done smoothly. Figure 3.18 shows a linearly complemented mesh of one model A (female) and another model B (male). The distribution ratio of model A is varied in five steps of 100, 75, 50, 25, and 0 %. From left to right, it can be seen that the properties of model A are gradually changing to those of model B.

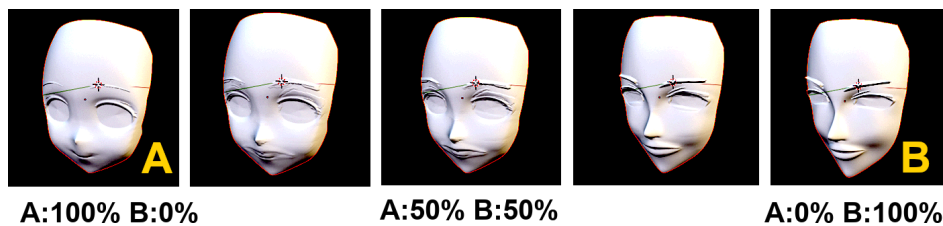


Figure 3.18: The samples of interpolation of mesh. This sequence shows interpolation from the mesh that has feminine features (A) to the mesh has masculine features (B).

In addition, I demonstrated that the same data set generation can be performed for textures. the all result of the texture data obtained is shown in Figure 3.24. The textures were extracted for all $n = 40$ data. Each texture in the training data has been deformed into a rectangular grid. This is similar to the state of unified topology data in a 3D model.

Figure 3.19 shows the linear completion of the textures of one model A (black eyes and eyelines) and another model B (blue eyes and eyelines). As in the mesh example, the change is divided into five steps, and it can be seen that the properties of model A gradually change to those of model B from left to right.

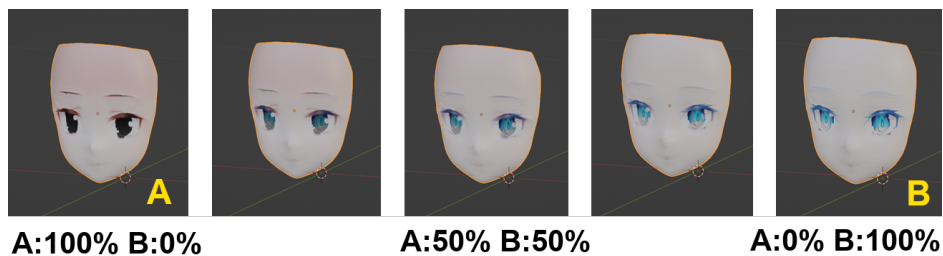


Figure 3.19: The samples of interpolation of texture. This sequence shows interpolation from the texture that has black eyes features (A) to the texture has blue eyes features (B).

In addition, since this method extracts mesh and texture features independently, it is possible to add the elements of each together arbitrarily (Figure 3.20). For example, if you have one model A and another model B, you can choose model A's for the mesh and model B's for the textures, or you can try the exact opposite allocations if you wish. In short, each element can be allocated as desired. This means that not only can each training data be allocated individually, but also the element X in one training data and the element Y in another training data can be conveniently allocated.[104]

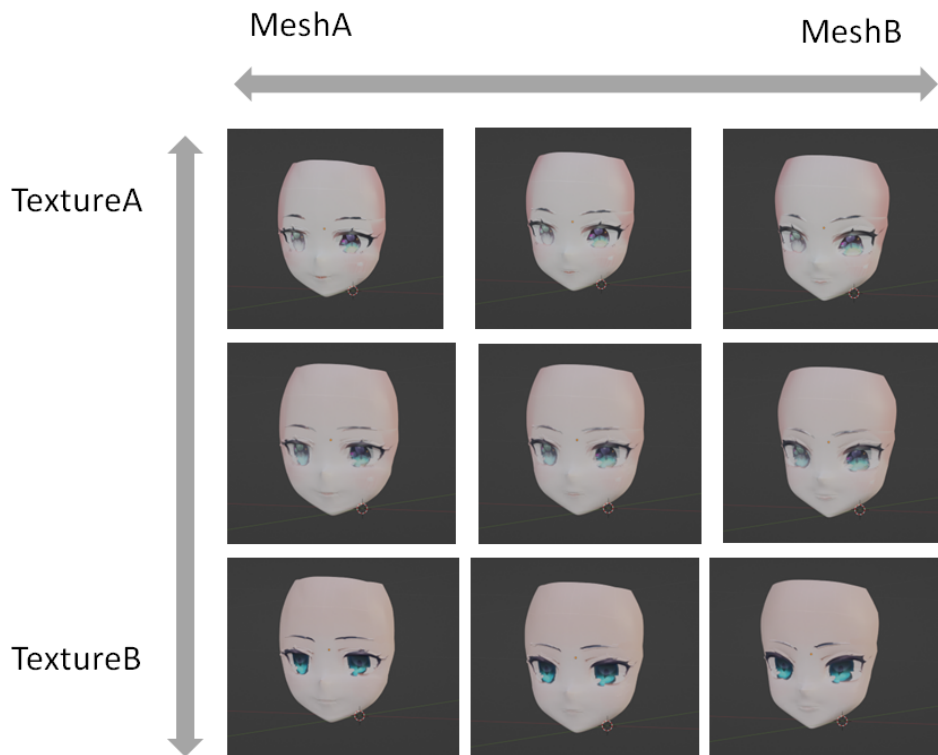


Figure 3.20: Allocation of meshes and textures for two models. The difference in rows indicates the distribution of textures, with the top row showing the texture of Model A, the bottom row showing the texture of Model B, and the middle row showing a mixture of both. The difference in columns indicates the distribution of meshes, with the left column showing the mesh of Model A, the right column showing the mesh of Model B, and the middle column showing a mixture of both.

These mesh and texture differences can be decomposed into principal components by PCA. Since the variation due to each principal component can be defined as the variation from the mean, the differences were stored in the form of shape key data for meshes and in the form of image data for textures.

By combining these, various meshes (Figure 3.21) and textures (Figure 3.22) can be redefined as a linear composite of each principal component.

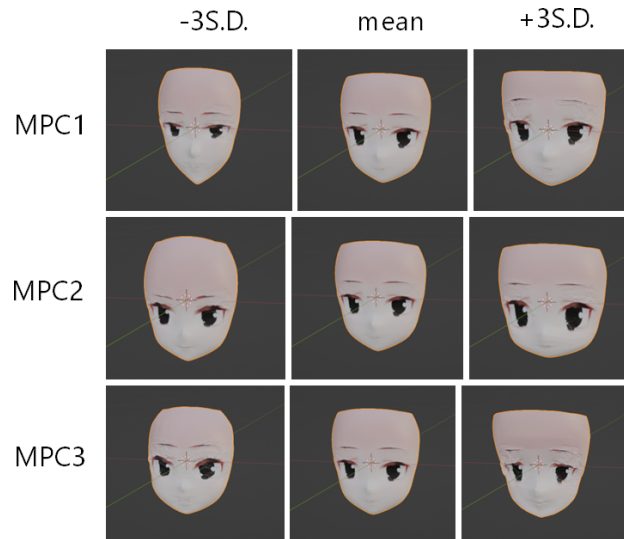


Figure 3.21: Variations of Mesh Principal Components (MPC). The top row represents MPC1, the middle row MPC2, and the bottom row MPC3. The left column shows changes at -3 S.D., the middle column represents the mean, and the right column indicates changes at +3 S.D.

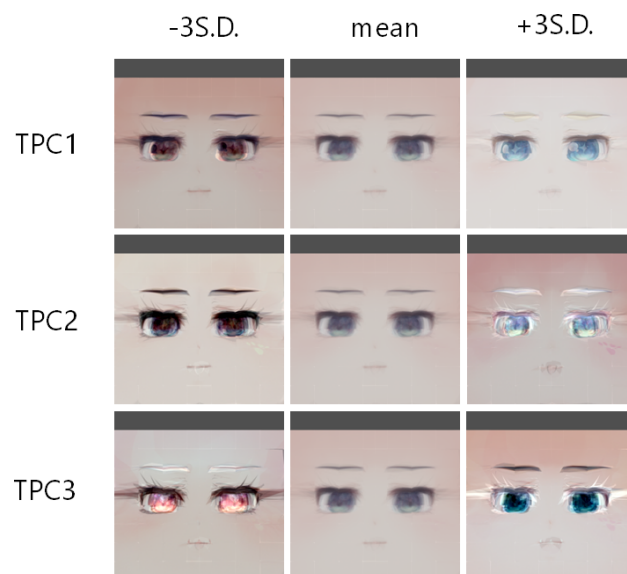


Figure 3.22: Variations of Texture Principal Components (TPC). The top row represents TPC1, the middle row TPC2, and the bottom row TPC3. The left column shows changes at -3 S.D., the middle column represents the mean, and the right column indicates changes at +3 S.D.

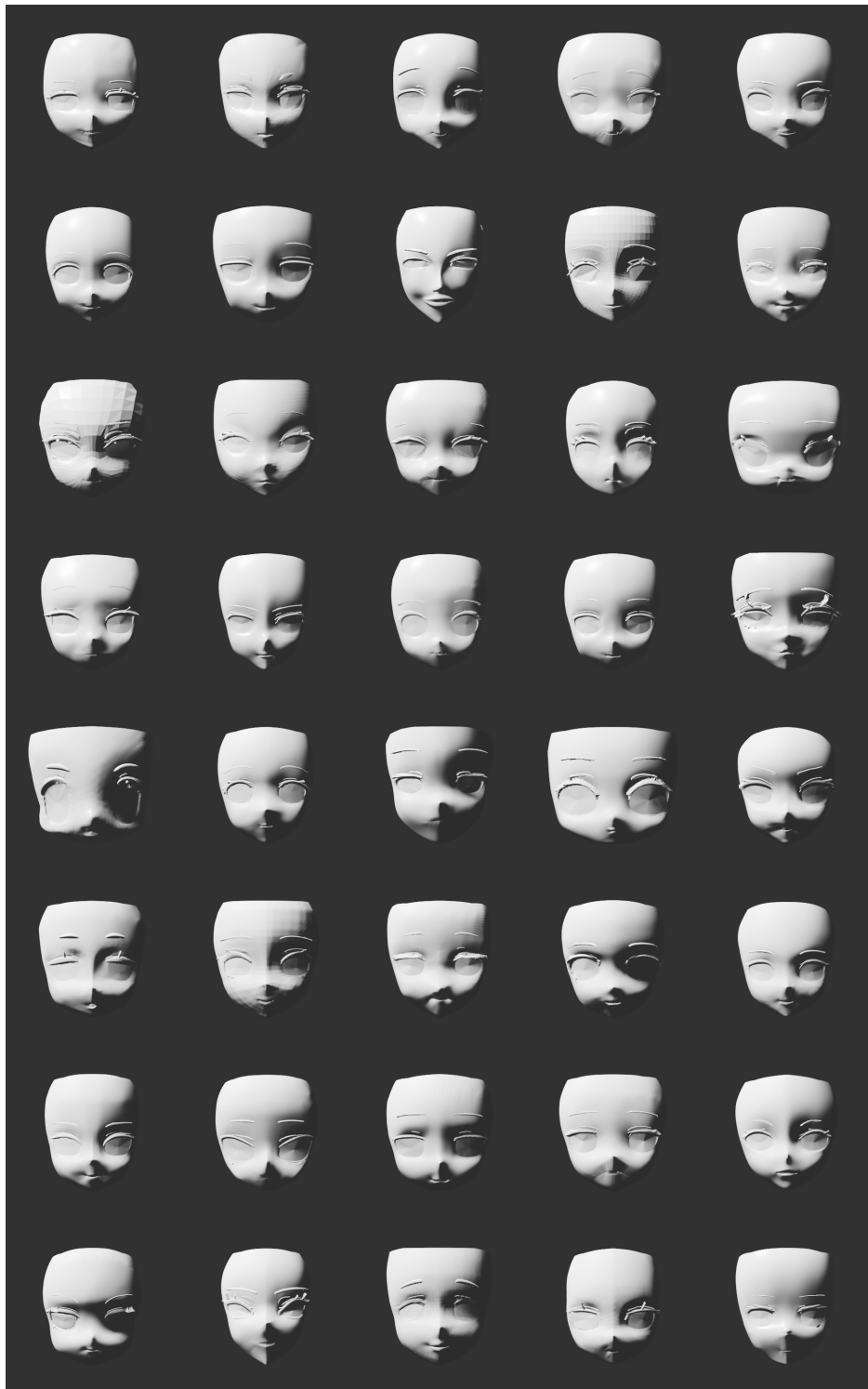


Figure 3.23: The obtained mesh data set of all the training data.



Figure 3.24: The obtained texture data set of all the training data. The order of the training data of the same as Figure 3.23

Chapter 4

Development of VR Application

In this chapter, I will discuss the implementation of analyzed data to VR application. The goal of this research is to eventually establish a system that allows users to create avatars easily. Based on the obtained data set, I developed applications that allows users to create avatars in a VR space. Unity, a gaming middleware, was used to develop the application.

4.1 Purpose and Issue

In Chapter 3, I have shown that new models can be generated by analyzing features from data from a large number of 3D models and distributing them in arbitrary proportions. I also showed that PCA can aggregate features into a small number of important components. I then created an application that allows users to actually generate avatars using those principal components. However, as discussed in Chapter 2, while users are more likely to provide physical input in a VR space, the type of parameters that can be manipulated is limited because the interface cannot be as dense and efficient as that of a conventional mechanical device. Therefore, the challenge in creating an avatar generation system is to realize an interface that can be freely generated based on limited physical input.

4.2 Allocation for VR Controller

In this application, I developed a method to solve this problem by assigning parameters to the rotation axis of the VR controller. First, the information that the VR device can input is the xyz position or xyz angle of the head,

left hand, and right hand. Of these, the position and angle of the head is completely dependent on the user's standing position and line of sight, making it impossible to manipulate as an independent parameter. The position of the hands is likewise strongly dependent on the user's own standing position. Since the player moves primarily in planar directions (horizontal and forward/backward), the vertical hand position can be detected as an independent parameter, regardless of the player's planar position. Detecting the relative arm position by detecting shoulder rotation could be considered, but this would require a large swinging motion of the arm, and there is concern about collisions between controllers and collisions with other objects in the real world, making this a highly dangerous motion. Therefore, the remaining parameters with a high degree of freedom are limited to the rotation of both hands. The angles of both hands can be manipulated independently of the other parameters by twisting the wrists. These motions are also highly compatible with everyday movements such as pitch direction (tapping), yaw direction (stroking), and roll direction (twisting keys).

Even for parameters that strongly depend on the user's own standing position, such as body position, a method that uses relative values is also conceivable. This is similar to the so-called gesture operation, where a threshold value is set and the parameter can be adjusted so that it only changes when the user makes a specific gesture. However, with such an algorithm using relative parameters, the optimal threshold value will change depending on the user's preferences and habits, making it difficult to find the optimal conditions for everyone.

The information that the controller (HTC Vive) can input also includes trigger and pad inputs. However, since these are discrete and single kind of operation, it is impossible to assign them to the operation of increasing or decreasing parameters. In order to operate parameters using these inputs, an additional pseudo-mechanical device (e.g., an up/down button that is controlled by a trigger) is essential.

Based on the above discussion, in this application, I decided to link the parameters with the absolute value of the Euler angle, which has a relatively high degree of freedom of operation and is analogous to a continuous parameter. This allows the user to specify any angle and reflect the parameters by independently controlling the wrist rotation, no matter what position the user is standing. The three angle parameters are limited, which is also consistent with the objective of summarizing the large number of parameter operations that were cumbersome in the conventional UI.

4.3 Overview of Application

This application serves as a VR avatar creation system, enabling users to deform the face and body of an avatar standing at the center of the VR space by manipulating a controller. The system overview is shown as Figure 4.1 and the actual scene is shown as Figure 4.2. The deformation of the avatar is executed via the controller, with the main components of the avatar's face and body changing based on the controller's rotations (Figure 4.3, Table 4.1). To leverage the benefit of VR, which allows for easy linkage between physical manipulation and parameters, the controller angle is associated with the principal components. This deformation is only applied while users press the controller's trigger, fixing the current state once the trigger is released. In this application, while the shapes of the face and body are adjustable, the hairstyle and face texture are fixed, as shown in Table 4.1..

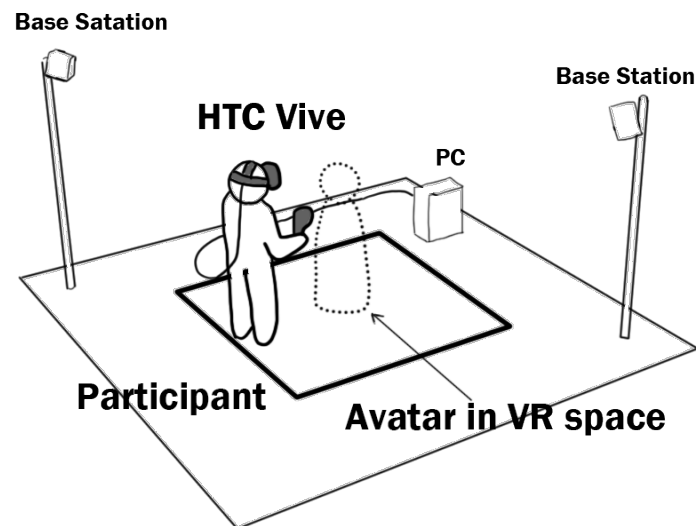


Figure 4.1: An overview of the application. The participant stands in front of the virtual avatar in VR space. The participant wear HTC Vive headset and controllers connected to PC.

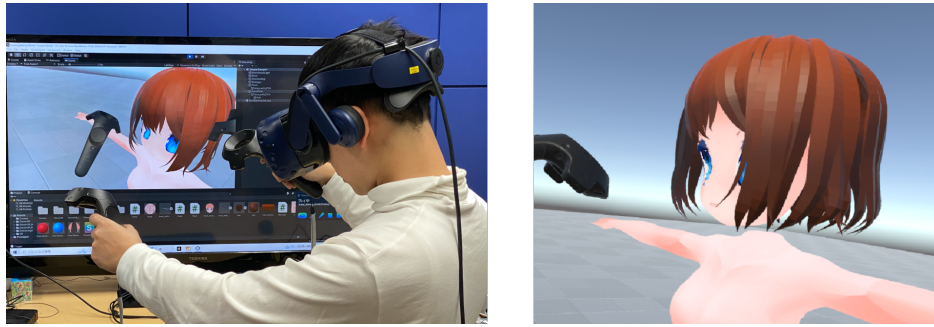


Figure 4.2: (Left) A participant using my system. (Right) The image the user sees via HMD display. The user can feel the avatar as if they are in front of the user.

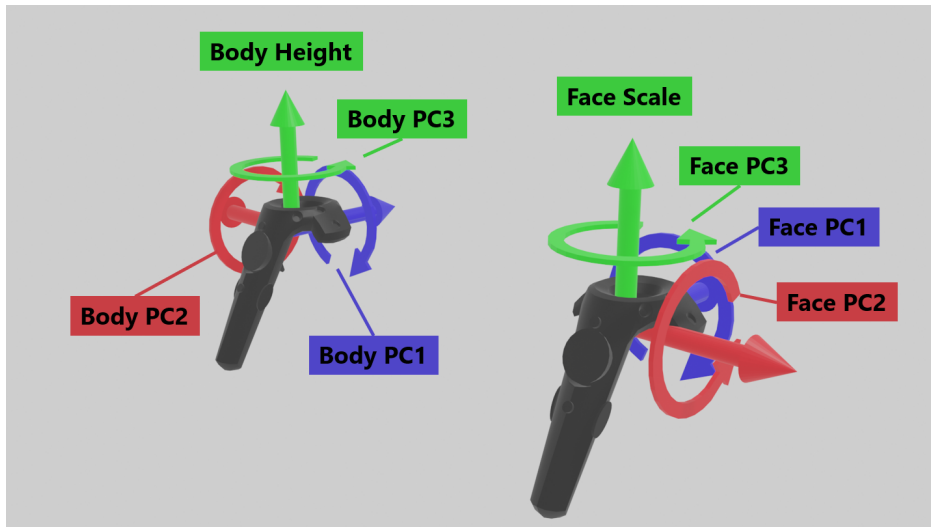


Figure 4.3: The relation between controller and attributes. The left hand relates body deformation, the right hand relates face deformation. The left rotation axes correspond to PC1, PC2, PC3 of body, and the vertical position corresponds to the height of the avatar. The rotation 3 axes correspond to PC1, PC2, PC3 of face, and the vertical position correspond to the scale of the avatar.

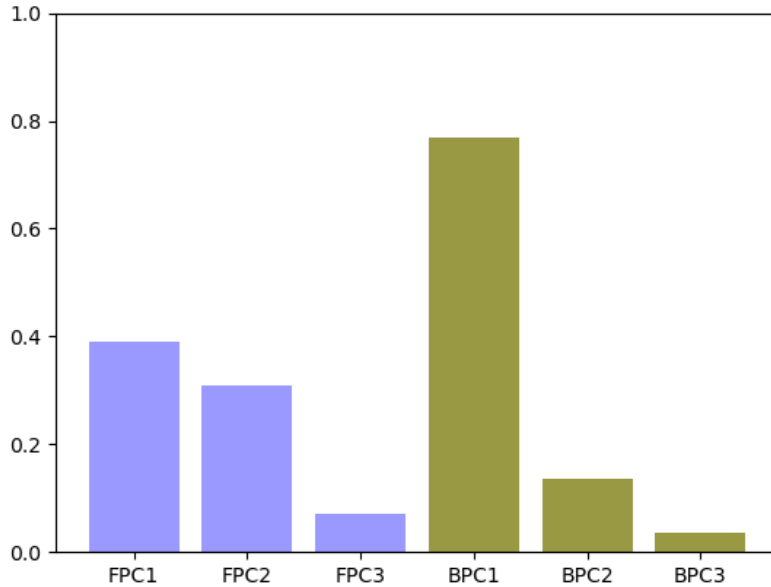


Figure 4.4: cumulative explained variance ratio of PCA. The left 3 are FPC (Face Principal Component), The right 3 are BPC(Body Principal Component)

4.4 User Study

4.4.1 Method

I conducted a user study with 10 participants. The participants are asked to create an avatar, assuming the role of their avatar of a video game using two types of interface, a conventional interface and VR interface that I developed.

As conventional interface, participants operated VRoid Studio (Figure 2.3), a parameter-adjustment type 3D avatar creation software. While in the conventional UI a participant uses 2D-Monitor and dozens of detailed parameters can be adjusted as shown in Figure 2.3, in the proposed UI a participant wears an HMD and manipulates three different parameters for each face and body in the VR space.

Post-operation of both systems, participants were queried on the following aspects to ascertain the effectiveness of the interface:

1. Have you experienced 3D software such as Blender, Maya?

Table 4.1: The changes of shape based on PC. The upper 3 rows show the changes about face, and the lower 3 rows show the changes about body. I extend the range of PC from ± 3 SD to ± 6 SD to exaggerate the changes. The changing of body shapes are shown from the side to reveal its depth change.



















PC	-6SD	0	+6SD
FPC1			
FPC2			
FPC3			
BPC1			
BPC2			
BPC3			

Table 4.2: The result of the questionnaire

Participant	Q1	Q2	Q3	Q4
1	Yes	No	5	5
2	Yes	Yes	4	5
3	No	Yes	4	5
4	No	Yes	4	5
5	No	Yes	4	5
6	No	Yes	4	5
7	Yes	Yes	4	5
8	Yes	Yes	5	5
9	No	Yes	4	5
10	No	No	2	4

Table 4.3: Time to complete work

Participant	2D-UI (sec)	VR-UI (sec)	dif
1	120	80	-40
2	120	90	-30
3	240	110	-130
4	270	50	-220
5	300	60	-240
6	650	110	-540
7	150	50	-100
8	460	100	-360
9	230	80	-150
10	290	100	-190

2. Have you experienced character making system in video game?
3. Is it more effective to operate few attributes than operate many parameters?
4. Is it more effective to operate in 3D VR space than with 2D monitor?

For questions 3 and 4, I used 5-point Likert scale.

4.4.2 Result

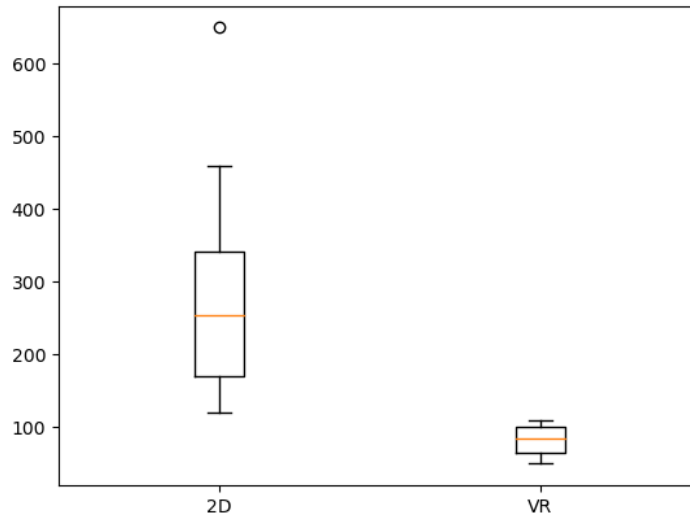


Figure 4.5: Comparison of the time to complete. The left is 2D-UI and the right is VR-UI.

I got a mean score of 4.0 in Q3 and a mean score of 4.9 in Q4 (Table 4.2). Almost all participants gave a rating of 4 or higher for Q3 (more effective to operate few attributes) and Q4 (effective to operate in 3D VR space). The median time to complete in 2D-UI was 255 sec, the median in VR-UI was 85 sec, and the median of time difference was -170 sec (Table 4.3, Figure 4.5). The time to complete was shorter for all participants when the application. The significant difference ($p < 0.01$) was found between two groups.

No relationship was found between the two conditions of Q1 (experienced 3D software) and Q2 (experienced character making system in video game) and no relationship was found between the evaluation and the time.

4.4.3 Discussion

All participants completed the task in shorter time in the application. This suggests that an interface similar to this application is effective in reducing the time required for avatar creation.

The effectiveness of modeling in the VR space was apparently rated highly, with 9 participants giving 5 and 1 participant giving 4. Positive

comments from the participants were that, they could get a better sense of the model by viewing it from various angles with a three-dimensional view, and that they could model with a sense of realism because they could feel the height difference.

The simplification of operations through attribute extraction seemed somewhat effective, as it was rated 4 or higher by 9 participants. The participants generally responded positively, but I got the following negative responses.

- When it came to millimeter-level adjustments, precision of this application was not enough. It was impossible to create perfectly what I wanted to create.
- I could not predict how will the operation change the avatar's shape. Some change was against my intention.

These reactions suggest that the application is suitable for making global level adjustments in a short time, while the conventional system, which uses many parameters, is suitable for making detailed adjustments after such coarse adjustments have been completed. This suggests that it would be ideal if both operations could be completed within VR space. But it is difficult due to the following limitations.

First, it is difficult to point to a precise position or make millimeter-level adjustments via the VR controller. Therefore, if the number of parameters is increased to dozens, it is almost impossible to manipulate them in the VR space. As mentioned in section 2.2.5, for the human body, movements such as repeated stop-and-go within a fine range, or maintaining a certain pose for an extended period, are unlikely to occur in daily life. So those movements are associated with great stress. If based only on work time, it is difficult to determine whether the reduction in work time is due to a sense of mission being completed or to escape this stress. More detailed survey items and measurement data are needed to determine this.

It is also difficult to perform actions such as observing the text or the reference object, which can be easily done with conventional method. In conventional illustration work, a creator could look at its hand to check the tip of the pen, or glance at the reference object immediately to verify the ideal color and shape without stress. However, in a VR space, such feedback is difficult to obtain. The HMD must be removed in order to perform the operation, but frequently putting on and taking off the HMD is not only time-consuming but also psychologically stressful.

The above points are not only limitations of the system, rather limitations of the HMDs and controllers at now.

In the future, as the performance of VR devices improves and the interface of VR applications more sophisticated, it is expected that some of these issues will be resolved. But at present, when manipulating parameters in the VR space, I need do all work in VR space, and the number of parameters that can be handled is limited to a few.

4.4.4 Problems to be Solved

The results of user study suggested that while operations in VR space have the effect of shortening operation time. But depending on the input method, it may cause confusion for the participants.

Additionally, this user study does not identify whether the factor that led to the reduction in work time was the VR work environment or the VR interface. In order to clarify this point, other conditions need to be unified. In other words, it is necessary to implement a non-VR interface in VR environment.

Therefore, I aimed to compare an interface emulating classic interfaces with a VR-specific interface, and conduct another user study that would clarify the effect of VR interface. In the next chapter, I will describe their specific implementation and discussion after its user study.

Chapter 5

Study on VR Interfaces

In this chapter, based on the results of the user study in Chapter 4, a detailed study of the interface is conducted. 3 different methods of manipulating parameters in the VR space, each with different characteristics, were devised and implemented as an application. Then, user studies were conducted to study the influence of each operation method on user behavior and the merits of each operation method in creating avatars.

5.1 Implementation

5.1.1 Plan of Interface

In Chapter 4, I proposed using the controller angle as an input method for VR (Application 1). However, as mentioned Chapter 2, there is still no definitive answer as to what interface is optimal for manipulating multiple parameters using VR. Therefore, I developed a new application, referred to as Application 2, for VR comparison in order to explore the optimal interface in a VR environment. In Application 2, I implemented not only the controller-based input method devised in Application 1, but also a pseudo-mechanical device in the VR space that is more similar to conventional interfaces such as an elevator's up/down button. The biggest difference between these methods is whether or not the user needs to look at a different device (e.g., button or switch) when selecting parameters or increasing or decreasing parameters. Since the UI emulating pseudo-mechanical devices is closer to that of the character-making system, users might be likely to be more familiar with it. It is also possible that a UI that combines elements of both would have the advantages of both, thus providing good experience. Then, I developed an intermediate type of input method that has properties intermediate between

those two. By having users evaluate and study with these three input methods, I analyzed what kind of interface is desirable when handling a large number of parameters in a VR space.

In other words, the question here is as follows, **What kind of UI is efficient when handling multiple parameters in VR space?** was also devised and verified. I also clarified through user studies whether the operations that are efficient and increase user satisfaction in VR space will be physical ones that take advantage of the characteristics of VR controllers or nonphysical ones that imitate conventional physical devices.

5.1.2 Overview of Application

The user stands in the VR space and generates an avatar. The items that can be manipulated are as follows

- Principal component 1 of the avatar's facial mesh
- Principal component 2 of the avatar's face mesh
- Principal component 3 of the avatar's face mesh
- Principal component 1 of the avatar's face texture
- Principal component 2 of the avatar's face texture
- Principal component 3 of the avatar's face texture
- Avatar height
- Avatar's hairstyle

Application 1 allowed manipulation of the principal components of the body mesh, but no participants mentioned the body manipulation on the questionnaire, that means what is important is the face for almost user. Therefore, in Application 2, instead of the mesh of the body, the texture of the face was made manipulatable to enhance the expressiveness of the avatar's face. For the principal components of the avatar's face mesh and texture, the following three methods were implemented for manipulation via controller.

First, I define the controller (C) method as the method that takes controller inputs as parameters as they are, as used in Application 1. In Section 4.2, I discuss how the player's own position strongly influences the horizontal position. Therefore, I used the controller angle as an independent input

parameter. This method takes full advantage of the unique strength of the VR space, which is that physical operations can be used as input devices as they are.

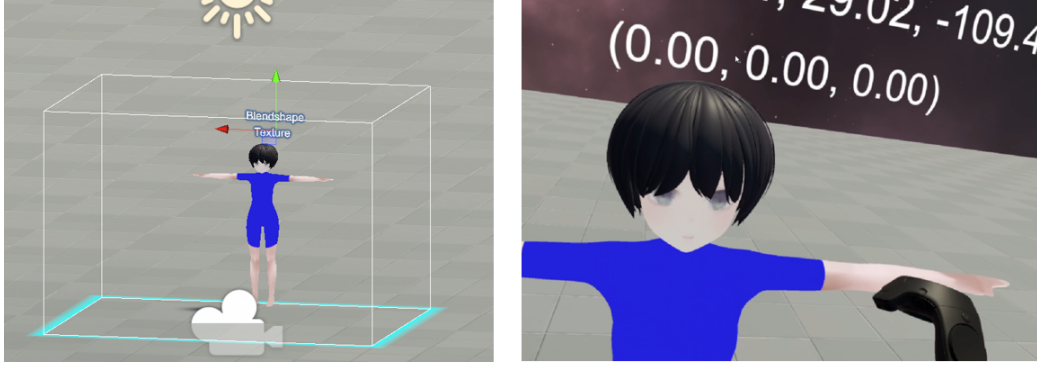


Figure 5.1: An overview of operation C method. The left is an orthogonal perspective of the system. The right is point of view of a user. A user operate parameters by only controller's Euler angle. There is no additional pseudo-button in VR space and the user can operate all parameters with controller.

As a counterpoint to this, I proposed a method that emulates mechanical devices as they currently exist (Figure 5.2). For example, in conventional research, the overwhelming majority of cases in which physical objects such as buttons are placed in VR space to imitate them are in the form of emitting a laser pointer from the controller and pressing the trigger while the hitboxes are overlapping [105] [106]. There are other examples where pseudo-buttons fixed in mid-air are pressed at close range. However, these require the user to walk around aiming at the pseudo-button, which can cause significant physical fatigue. Therefore, interfaces using laser pointers are most commonly used. In the user study, there are 6 types of parameters for each mesh and texture, and 2 types for each up/down button, so 12 types of pseudo-buttons need to be manipulated. Therefore, since 12 types of pseudo-buttons need to be operated, I have arranged these pseudo-buttons accordingly. As Figure 5.2 shows, the upper green buttons mean the parameter of the face mesh, and the lower blue buttons mean the parameter of the face texture. The upward and downward triangles symbolize increase and decrease, respectively, similar to the function of certain buttons in real-world applications. This is thought to emulate mechanical buttons in the VR space, as is common in the conventional real world. Because of the need to aim at many pseudo-buttons, I made them larger than the real mechanical buttons (a few centimeters apart) and set the distance between each pseudo-button to be about 30 centimeters.

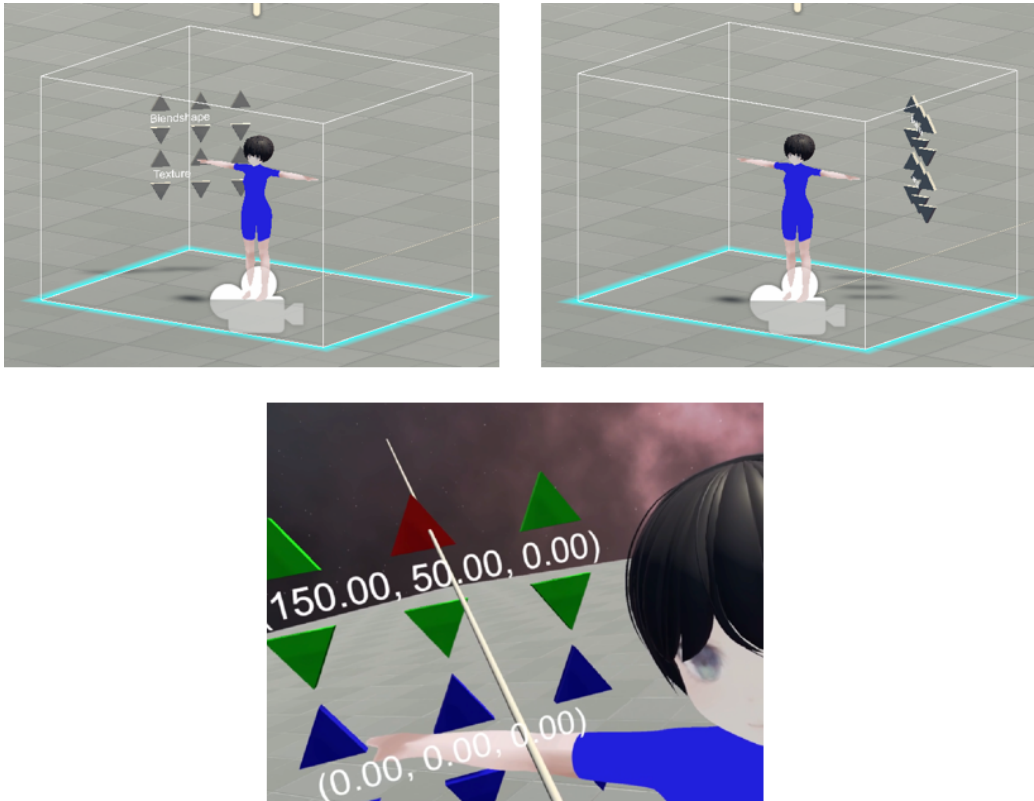


Figure 5.2: An overview of operation B method. The upper row is orthogonal perspectives of the system, the left is with the left sided pseudo-button, whereas the right is with the right sided. The lower is point of view of a user. A user can operate parameter using pseudo-buttons and virtual laser pointer. Green button means the parameters of the face mesh. Blue button means the the parameter of the face texture.

In addition, I proposed an operation method that is intermediate between these two methods, which I called the hybrid type. The advantage of the button method is that it is easy to clearly identify the parameter you are currently trying to operate. On the other hand, compared to the controller method, it's more cumbersome to operate a certain parameter as it requires focusing on and looking at the pseudo-buttons. To solve this problem, I devised a system in which parameters are selected explicitly by the controller, but can be increased or decreased physically. 6 cubes are placed in the VR space to clearly indicate which parameter is currently selected. The type of selected parameter can be switched by pressing a switch on the left-hand controller. Increasing and decreasing the parameters is done with the right hand controller, and two switches are assigned to increase and decrease the

parameters. The switches are positioned in such a way that they naturally move up and down when the player holds the right-hand controller, allowing the player to operate it intuitively without looking at his/her hand, which is a physical operation.

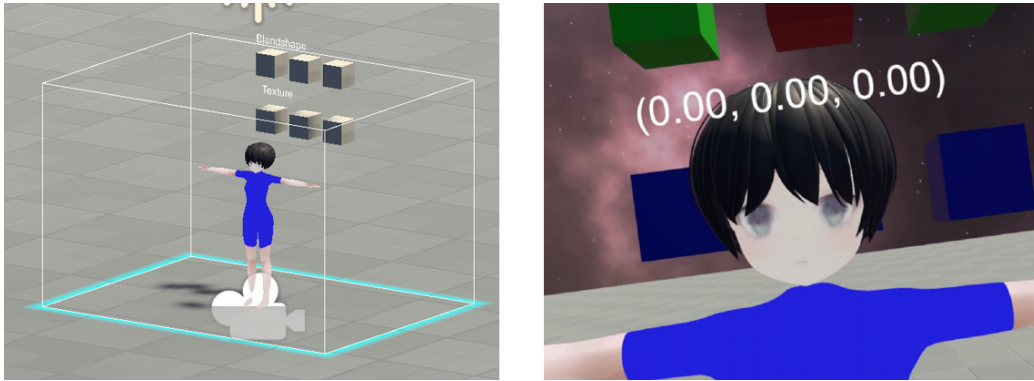


Figure 5.3: An over view of operation H method. The left is an orthogonal perspective of the system. The right is point of view of a user. The cubes means what parameters a user is selecting. The upper 3 cubes means the face mesh, whereas the lower 3 cubes means the face texture. Increase/decrease of the parameter can be operated with controller.

The following is a summary of each operating method. First, there are two types of operation method (Table 5.1). One is the controller-within method, in which the selection of parameters and their increase/decrease are completed by the controller. The other is the interaction with a pseudo-mechanical device. C method is a controller-within method in both selection action and increase/decrease action. So, it is the most VR-dedicated method. On the other hand, B method emulates the conventional method by selecting pseudo-mechanical buttons with a laser pointer. H method has a hybrid nature since the selection is done with pseudo-mechanical buttons and the increase/decrease is done with the controller within.

5.2 User Study

5.2.1 The Flow of the User Study

12 participants (male 10 and female 2, their years olds are between 20 and 30) were recruited to operate Application 2. The participants were asked to experience the three different operation methods in turn. In order to homogenize the habituation by order, the order of the operation methods

Method	Parameter Choice	Parameter Adjust	Feature
C	Controller Within	Controller Within	VR dedicated
H	Pseudo Mechanical	Controller Within	Hybrid
B	Pseudo Mechanical	Pseudo Mechanical	Emulate Mechanical Interface

Table 5.1: Summary of the 3 types of operations method.

was made so that the order of the $3 \times 2 = 6$ methods was evenly distributed. In addition, for method B, both a method with the pseudo-button on the right and a method with the pseudo-button on the left were prepared in order to take into account the influence of the position of the pseudo-button on the player’s behavior. Therefore, the pattern was further differentiated into two ways, for a total of $6 \times 2 = 12$ ways. The differences in order by participant are summarized in Table 5.2.

Participant	1st	2nd	3rd
1	C	H	B
2	C	B	H
3	H	C	B
4	H	B	C
5	B	C	H
6	B	H	C
7	C	H	B*
8	C	B*	H
9	H	C	B*
10	H	B*	C
11	B*	C	H
12	B*	H	C

Table 5.2: The orders by participants. Every single characters stands for methods. C means Controller method. H means Hybrid method. B means Button method. B* means Button method with mirror arrangement.

For each method, participants were asked to experience the application in the following sequence.

When the application starts, an avatar appears in front of the participant. The participant changes the avatar’s face, texture, height, and hairstyle by changing the parameters. When the participant feels that he/she has created an avatar of his/her choice, the application is terminated.

During this time, the following information is recorded at 0.1 second intervals. All movements of the participants during the application experience were agreed upon, recorded, and used for later data analysis.

- Operation time of the application
- Principal component 1 of the avatar’s facial mesh
- Principal component 2 of the avatar’s facial mesh
- Principal component 3 of the avatar’s face mesh
- Principal component 1 of the avatar’s face texture
- Principal component 2 of the avatar’s face texture
- Principal component 3 of the avatar’s face texture
- Height of avatar
- Changing the avatar’s hairstyle
- User’s HMD x-position (left/right)
- User’s HMD y-position (vertical)
- z-position of the user’s HMD (forward/backward)

Based on these records, I calculated indices named movement distance and operation density. The movement distance is the cumulative total of the participant’s movement distance and is calculated by the following formula.

$$D = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (5.1)$$

A boring interval is defined as an interval in which the following conditions persist

The following conditions must be met:

- The distance moved between frames is less than 0.1 m.
- There is no change in the principal components of the avatar’s facial mesh or texture.
- There is no change in the avatar’s height or hairstyle.

Then, for each retraction interval, the effective length is reduced by 0.5 seconds. This is because, even though the operation is performed instantaneously on a frame-by-frame basis, it is unrealistic to assume that the boring interval would suddenly begin at the very next moment. Therefore, all boring intervals are considered to include a small portion of sustained attention from the previous valid operation, and that portion is deducted.

After experiencing all three ways of operation, the user is asked to answer the following questionnaire.

1. When there was a parameter you wanted to manipulate, were you able to choose it immediately?
2. When you wanted to increase or decrease a parameter, were you able to do so immediately?
3. When you increased or decreased a parameter, did the result change as you expected?
4. Did you feel mental fatigue, such as a feeling that things were not going the way you wanted them to?
5. Did you feel physical fatigue, such as pain in your arms or fingers?

5.2.2 Result

For all participants, the avatar parameters and the participant's trajectory were recorded and illustrated. Examples are shown in Figure 5.4 and 5.5. Figure 5.4 shows the avatar parameters and the horizontal axis shows the time flow. Each line graph shows the transition of the principal components 1, 2, and 3 of the face mesh and the principal components 1, 2, and 3 of the texture and the height of the avatar, with the vertical axis normalized with the maximum value of each being 1 and the minimum value being 0. Figure 5.5 shows the movement of the participant, showing the path of movement with the participant looking down from directly above. In other words, the horizontal axis shows the movement in the left-right direction, and the vertical axis shows the movement in the front-back direction.

Table 5.3 summarizes the results of studies, method, and distance traveled for all participants.

Table 5.4 also summarizes the results of the questionnaire for all participants.

An analysis of variance (ANOVA) was performed to examine significant differences between the conditions of the study and the measurement results and questionnaire results. The results are shown in Table 5.5.

5% significant differences were found in "method-duration" and "method-valid". The results are illustrated in a box whisker plot in Figure 5.6 and Figure 5.7.

All trajectory figures and parameter change plots are shown in Appendix A.

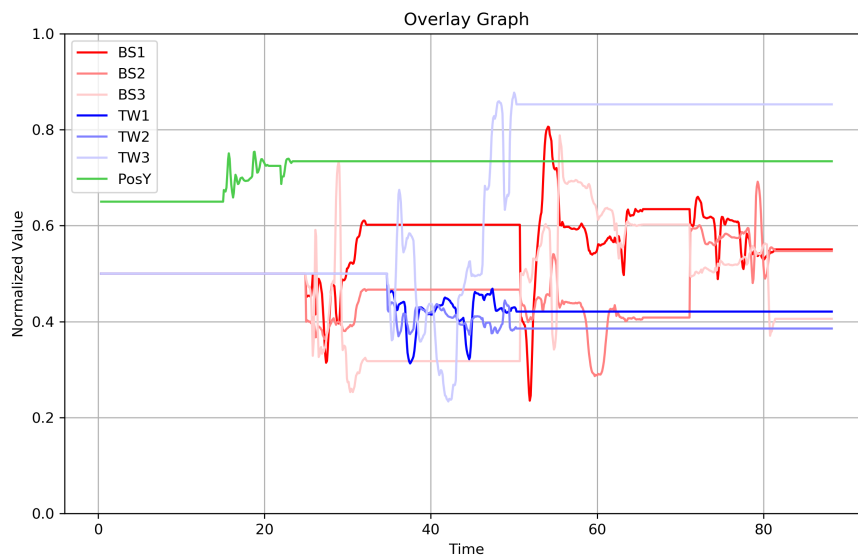


Figure 5.4: A Sample of overlay graph of the user study. Horizontal axis means the flow of time, Vertical axes are 0,1 normalized proportion of parameters. Parameters are principal components of BS(Blend Shape) 1,2,3 and TW(Texture Weight) 1,2,3 and avatar height(PosY).

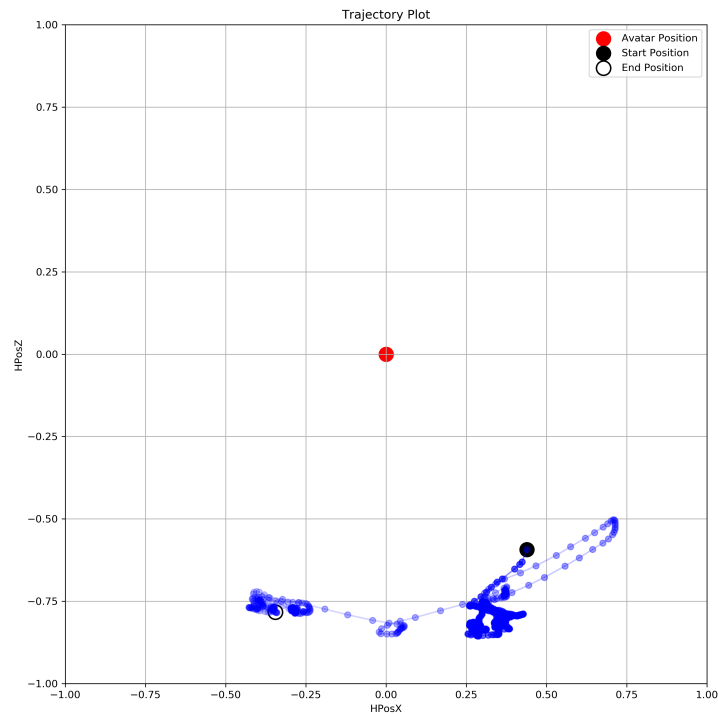


Figure 5.5: A Sample of trajectory graph of the user study. Horizontal axis are the participant's posX (left-right direction) and posZ (front-back direction). The center red circle is the where avatar stands. The black circle and the white circle is the participant's start and end position. The trajectory where the participant moved is the sequence of blue line. The trajectory color gets pale as the time flows.

Participant	Turn	Method	Duration	Trip	Valid
P1	1	C	138.90	16.72	0.94
	2	H	161.10	11.81	0.82
	3	B	117.90	12.92	0.91
P2	1	C	88.90	5.22	0.82
	2	B	121.40	5.61	0.58
	3	H	149.70	9.82	0.76
P3	1	H	100.80	8.62	0.85
	2	B	91.20	4.43	0.67
	3	C	66.80	7.93	0.98
P4	1	H	97.30	5.68	0.73
	2	C	63.00	3.57	0.89
	3	B	95.80	4.80	0.68
P5	1	B	110.70	3.77	0.79
	2	C	87.90	1.93	0.95
	3	H	113.60	3.23	0.79
P6	1	B	92.20	6.89	0.73
	2	H	103.40	5.69	0.82
	3	C	60.00	2.97	0.95
P7	1	C	141.10	13.43	0.76
	2	H	197.30	9.02	0.69
	3	B*	125.60	8.58	0.76
P8	1	C	72.60	1.74	0.68
	2	B*	81.60	3.73	0.75
	3	H	72.20	2.40	0.72
P9	1	H	172.90	8.49	0.50
	2	B*	76.30	1.90	0.48
	3	C	64.20	6.61	0.86
P10	1	H	114.30	3.67	0.43
	2	C	108.00	1.70	0.72
	3	B*	98.10	2.07	0.38
P11	1	B*	118.00	4.72	0.45
	2	C	156.50	3.50	0.50
	3	H	96.80	2.64	0.43
P12	1	B*	92.60	6.16	0.63
	2	H	113.40	4.69	0.65
	3	C	53.40	2.16	0.84

Table 5.3: All measured data of the user study

Participant	Turn	Method	Q1	Q2	Q3	Q4	Q5
P1	1	C	4	2	4	5	5
	2	H	3	5	4	5	2
	3	B	3	3	5	3	5
P2	1	C	2	2	3	3	4
	2	B	3	3	4	5	4
	3	H	4	4	4	5	4
P3	1	H	4	5	4	5	5
	2	B	4	4	4	5	2
	3	C	3	4	3	2	5
P4	1	H	4	4	5	4	4
	2	C	4	4	3	3	3
	3	B	5	5	5	5	4
P5	1	B	5	5	5	5	5
	2	C	2	2	3	3	5
	3	H	5	5	5	5	5
P6	1	B	2	3	4	2	2
	2	H	5	5	4	2	2
	3	C	3	3	4	2	2
P7	1	C	5	5	3	3	5
	2	H	2	5	4	4	5
	3	B*	2	5	4	4	5
P8	1	C	5	4	5	4	4
	2	B*	3	4	4	4	4
	3	H	3	3	4	4	4
P9	1	H	2	2	2	3	5
	2	B*	5	5	3	5	5
	3	C	5	5	3	5	5
P10	1	H	3	4	4	3	2
	2	C	4	5	4	3	2
	3	B*	2	3	3	3	3
P11	1	B*	1	4	3	4	5
	2	C	2	4	4	5	5
	3	H	4	3	2	5	5
P12	1	B*	4	5	4	4	4
	2	H	2	2	3	3	4
	3	C	3	4	3	5	4

Table 5.4: All questionnaire results of user study

Parameter	Turn	Method	Turn \times Method
Duration	0.135	0.036*	0.192
Trip	0.308	0.840	0.275
Valid	0.326	0.022*	0.877
Q1	0.871	0.883	0.605
Q2	0.717	0.658	0.650
Q3	0.807	0.349	0.658
Q4	0.579	0.493	0.488
Q5	0.868	0.946	0.698

Table 5.5: Table of ANOVA Probabilities. Each column are conditions, Turn, Method, and their interaction. Turn condition means difference by 1st,2nd,3rd studies par a participant. Method condition means difference by C,H,B methods. * star means probabiliy under 5 % so it has significant difference.

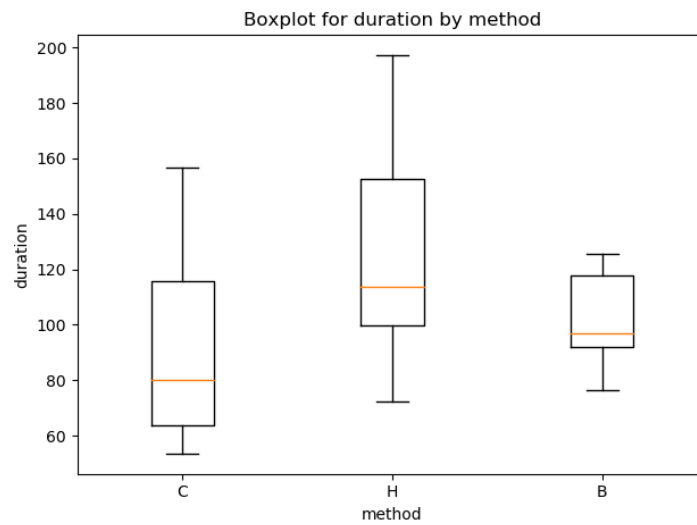


Figure 5.6: Boxplot of method and duration. C method have significantly decreases duration than other methods.

Table 5.6: Correlation Matrix of measured data and questionnaire answer

	dur.	tri.	val.	Q1	Q2	Q3	Q4	Q5
duration	-	0.56	-0.25	-0.26	0.04	0.05	0.16	0.20
trip	0.56	-	0.35	0.00	-0.01	0.08	0.03	0.23
valid	-0.25	0.35	-	0.21	0.02	0.24	-0.18	0.02
Q1	-0.26	0.00	0.21	-	0.54	0.35	0.33	0.00
Q2	0.04	-0.01	0.02	0.54	-	0.37	0.32	-0.02
Q3	0.05	0.08	0.24	0.35	0.37	-	0.20	-0.15
Q4	0.16	0.03	-0.18	0.33	0.32	0.20	-	0.38
Q5	0.20	0.23	0.02	0.00	-0.02	-0.15	0.38	-

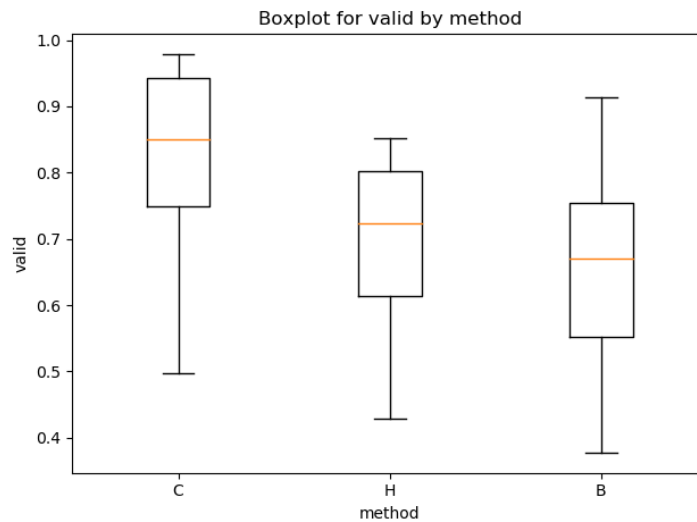


Figure 5.7: Boxplot of method and valid. C method have significantly increase valid value than other methods.

5.2.3 Discussion

5.2.3.1 Measured Data

First, the significant difference between method and duration indicates that the Controller method has a significant effect on reducing operation time. This is in common with the conclusion obtained in user study of Chapter 4, where the VR-dedicated interface was found to reduce the operation time compared to the conventional 2D interface. In addition, the VR-dedicated interface was found to reduce the operation time compared to the conventional interface that emulates a mechanical button interface. In other words,

it suggests that the use of physical operations, which is a characteristic of VR, can reduce the operation time for many users.

In addition, a significant difference was found between method and valid. The time that users spend practically operating the application tended to be significantly higher for the controller method. One possible reason for this is that C method produces high values by necessity due to its operational mechanism while moving hands. However, there were participants whose valid value of C method was not higher than that of the other methods, and there were also participants who, like Participant 8 , produced the lowest valid value. This indicates that there is no bias due to the method, and that C method is purely effective in increasing the density of manipulation (non-boring time/total operation time) by participants.

5.2.3.2. Distance Traveled

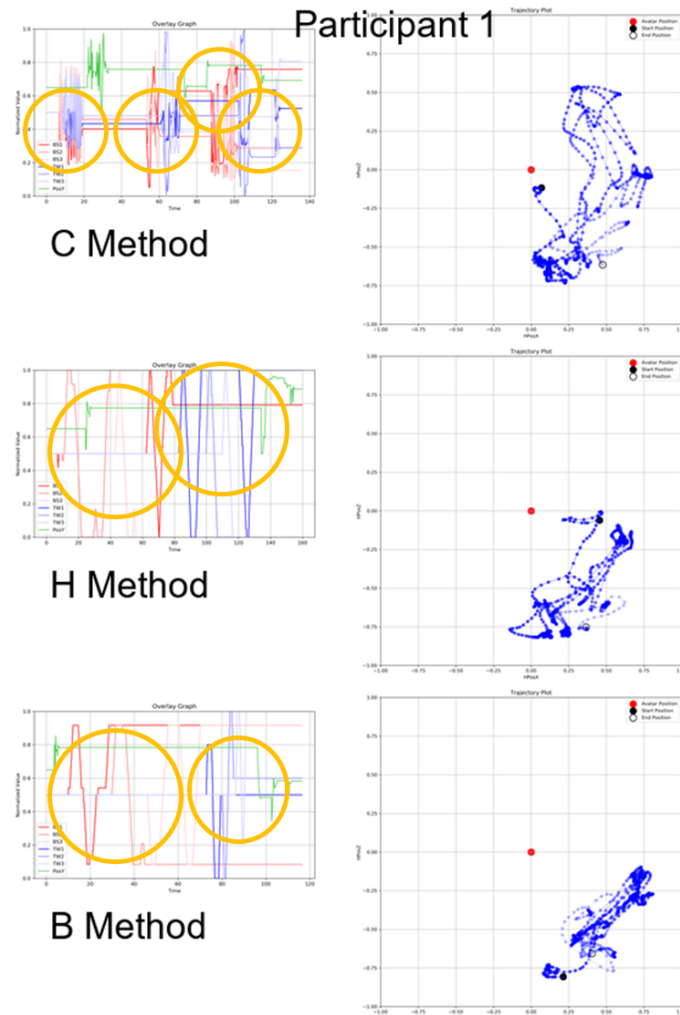


Figure 5.8: Total results of the participant 1. The intensively operation areas are marked by yellow circle.

The major difference among participants was in the distance traveled. For example, participant 1 (Figure 5.8) showed a large movement distance in C method. The reason for this may be that C method does not have any pseudo-buttons on the UI, and therefore, the movement is not restricted. The two user studies with the greatest distances traveled were both C methods. However, there are counterexamples. participant 6 and participant 12 had the greatest distance traveled in Method B.

In addition, for most participants, the participant's movement range was heavily skewed to the right side. This could be due to the following factors.

- The B method buttons was located on the right side, which caused the bias. (Half of the participants experienced the right arrangement in B method)
- Right-handed users psychologically turned their dominant hand outward.

Regarding the bias due to the placement of the buttons, it cannot be said that there is such an effect, since participants of the B* method (mirror arrangement) did not show any cues to be affected. On the other hand, for the dominant arm, almost all participants were right-handed (11 participants) or ambidextrous (1 participant). This indicates that the dominant arm may have a significant effect on the range of movement. Note that the ambidextrous participant (Participant 8) was found to stand slightly left of center, but this is not definitive evidence since this case is $N = 1$.

While there were some participants who turned to the left and right (max observed angle was 150° when the front of the avatar defined 0°), there were no participants who turned completely to the back. This is thought to be because the element operated in this application are mainly faces, which are parts that have little meaning when viewed from behind.

In the real world, due to the increasing miniaturization and densification, the dominance of one hand over the other, for example, on a piano or keyboard, has little impact on the user experience. On the other hand, as discussed in the section 2.2, when a pseudo-mechanical device is provided for a VR UI, its size must be larger than that of the real world. Therefore, a UI whose usability largely depends on the user's standing position may greatly decrease the user experience. In short, if the pseudo-button must be placed on the left side in VR, this might be a bad-UI for the right-handed users, and vice versa.

5.2.3.3 Questionnaire Answer

Next, I discuss the level of satisfaction felt by the participants, based on the questionnaire survey.

Of the five questionnaire items, Q1, Q2, and Q3 are considered to be related to operability. In other words, when a user wants to perform an operation, how quickly and accurately can the user perform the operation? For these three questions, the responses tended to fall into two extremes: either all were answered with high values or all were answered with low values. In

the linear analysis, there was a positive correlation between Q1, Q2, and Q3, all of which had a high correlation coefficient of $r=0.54$, especially between Q1 and Q2.

However, there were some responses in which the values were far apart in Q1 and Q2; in C method on P1, respondents answered that it was easy to select, but difficult to increase or decrease. In addition, the B and H methods on P7 and B method on P11, respondents answered that it was "difficult to select, but easy to increase/decrease". This is thought to be due to the characteristics of each method.

Q4 and Q5 are considered to be related to comfort. In C method, many respondents answered that mental fatigue exceeded physical fatigue (physical fatigue was mentioned only in P10). This may be due to the fact that the input method of twisting the controller is an operation method that is not used in everyday life, and thus caused more mental strain, even though it requires less effort than the conventional operation of pressing the trigger. The other methods did not elicit biased responses as to which was more burdensome.

5.2.3.4 Measured Data and Questionnaire Answer

The relationship between the responses to the questionnaire and the measured data will be analyzed. In this user study, three indices were calculated: duration, which is pure studyal time; trip, which is the actual distance walked; and valid, which is the percentage of substantially active behavior. These and the responses to the questionnaire were considered as quantitative data, and their correlation coefficients were arranged into a matrix.

There was a significant positive correlation between duration and trip, but it is almost self-evident that walking distance increases with studyal time. There was also a significant positive correlation between trip and valid, but in the "non-boring interval". There was a negative, though not significant, correlation between duration and valid. This may be due to the fact that users with long duration include those who spent long periods of time without walking much, and thus the valid rate tends to drop.

Regarding the questions in the survey, the Q1, Q2, and Q3 groups, as well as the Q4 and Q5 groups, show similar trends within their respective groups. On the other hand, Q3 and Q5 have some negative correlation. This can be interpreted as the fact that it is easier to perform the operation as desired, which leads to fewer operations and less physical fatigue. It is noteworthy that valid and Q4 have a negative correlation. This means that the higher the density of operations, the greater the mental fatigue. Considering this point and the difference in the above methods, it is possible that C method

is a method that reduces operation time but increases mental fatigue.

5.2.3.5 Qualitative Discussion of the Questionnaire Comment

The following responses were obtained from the questionnaire.

Overall

- It would be better if guidance was provided not only in the lecture but also in the VR space.
- It was fun to move my body freely and observe the avatar from various angles.

C method

- I felt it was an intuitive operation.
- Compared to other methods, it was difficult to make detailed adjustments.
- It was difficult to grasp what kind of changes were taking place.

H method

- It would be easier to understand H method if the item you are operating on is displayed while you are operating it.
- It was a burden to remember which item indicates which parameter each time.

B method

- The GUI was intuitive and easy to understand.
- It is troublesome because of frequent eye movement (multiple similar responses).
- The laser pointer sometimes overlapped with the characters, making it difficult to see.
- It was easier to operate if the pseudo-button was on the right side (The participant performed B*, which is a mirror arrangement).

It is interesting to note that the term **intuitive** was used to describe different methods. For method C, **intuitive** refers to the increased real-time nature as the avatar continuously changes in response to the controller's movements. For method B, **intuitive** refers to the ease of understanding the function of each pseudo-button and how to operate them. Therefore, the latter concept can be described as **habitation**, but it is difficult to strictly distinguish between **habitation** and **intuition**. This is because human behavioral patterns are significantly influenced by memory and culture as well as physiological characteristics. In other words, an interface that is self-evident to one person can be puzzling to another. The results suggest that it is difficult to implement an interface that many people find **intuitive** in VR devices, where the standards are still not organized and few people use them.

It was also suggested that a detailed guide in the VR space is needed. They are forced to be placed at wider intervals than real mechanical devices due to the limitations of the resolution and input precision of the VR device itself. This causes frequent eye movement and object interference (e.g., overlap between the laser pointer and the avatar), which in turn causes stress to the operator. In this study, the operating procedures were lectured verbally just before the study, but even so, some users said that they felt burdened to recall the operating procedures during the study. In order to reduce the burden on the user, a careful guide that presents information and provides feedback at every timing is desirable.

In this study, B method was used in two arrangements, one with the control panel on the right and the other with the control panel on the left. Although no clear difference in the results was observed in this regard, some participants found it difficult to operate the panel when it was on the opposite side (left) of their dominant arm (right). In this user study, many participants walked in the right half of the space. However, since all participants were either right-handed or ambidextrous, it is impossible to determine whether this behavior was due to their dominant hand or simply a coincidence related to the right placement.

5.2.3.6 Problems to be Solved

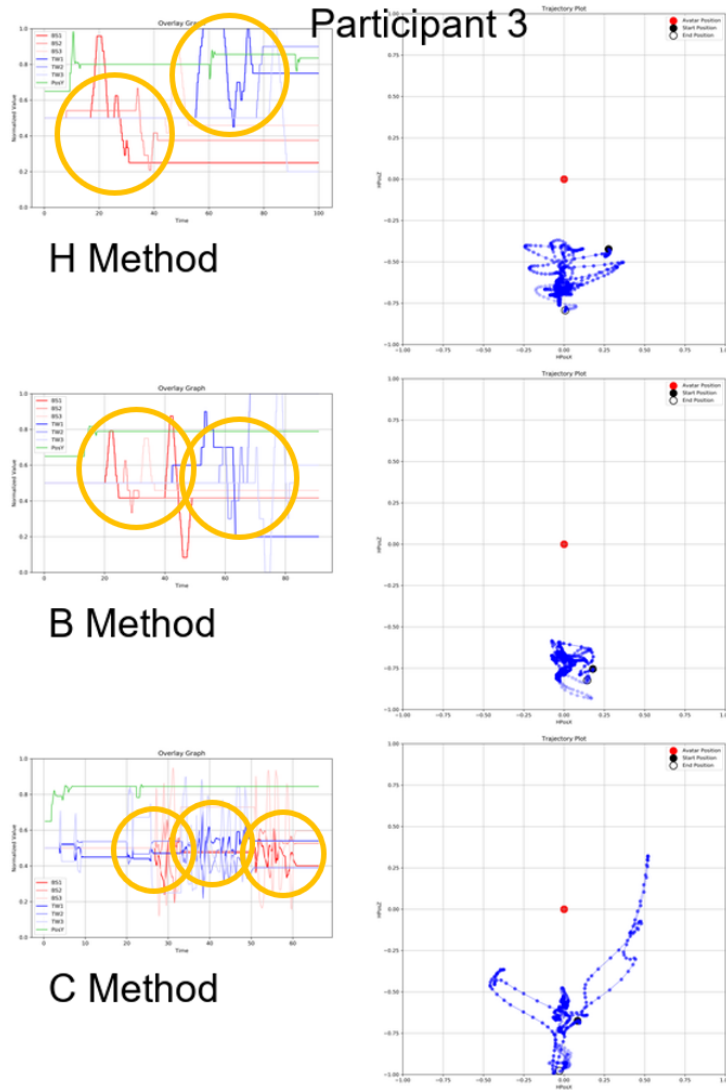


Figure 5.9: Total results of the participant 3. The operation intensively areas are marked by yellow circle.

Next, I will take a closer look at the graphs of the measured data by the participants. Figure 5.9 is the graph for participant 3.

In the H and B methods, the red lines (representing the principal components of facial shape) and the blue lines (representing the principal components of facial texture) exhibit discrete changes in polyline due to the specifications of the UIs. On the other hand, C method shows continu-

ous changes. It is important to emphasize here the intensive and repeated changes in the red and blue lines. In many participants, the manipulation followed the following sequence.

1. The shape of the face was intensively changed only once.
2. The texture of the face was intensively changed only once.

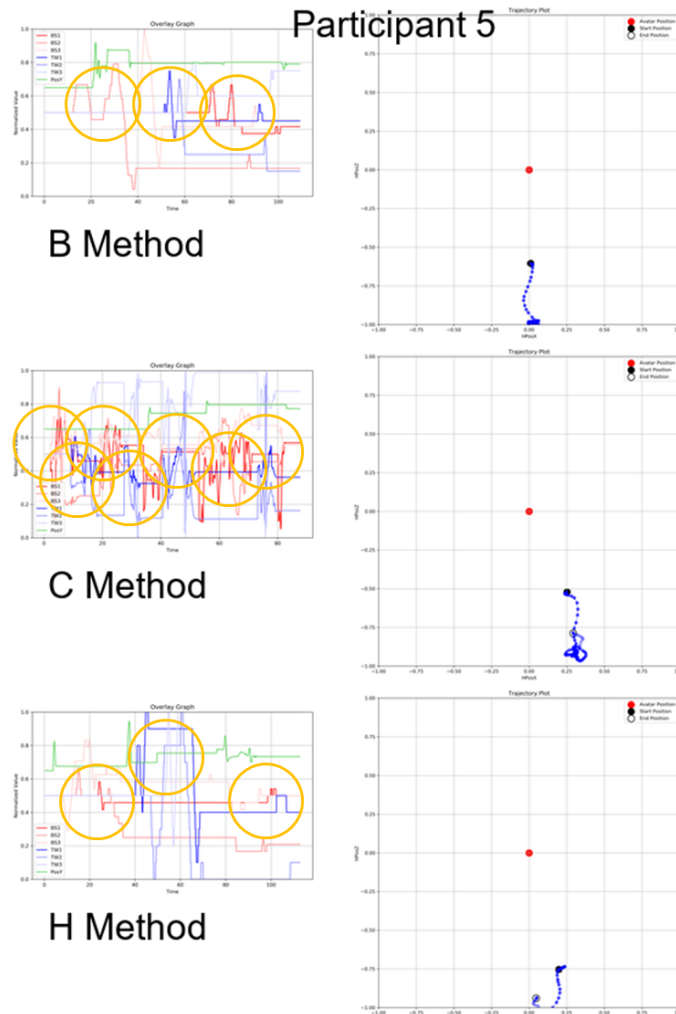


Figure 5.10: Total results of the participant 5. The intensively operation areas are marked by yellow circle.

In the case of C method, however, the behavior was such that these operations were repeated. In Figure 5.9 , such areas are marked, and it

can be seen that such cycles increased in C method. This trend is more pronounced in participant 5 (Figure 5.10), where these cycles are repeated seven times in C method, a larger increase than in the other methods.

I would like to examine the significance of these repeated cycles of operations. The repetition of an operation is defined as **revision** in the study. There are two possible motivations for users

- A user is not satisfied with the operation and wants to revise it. (Negative)
- A user wants to see how the avatar changes due to the manipulation. (Positive)

In the user study, when a certain operation was completed midway through, it was unclear whether it was because the participant was satisfied with the operation or whether they had given up. And it was difficult to judge the quality of the operation. This is because the theme was "your favorite avatar", which had a high degree of freedom.

5.3 Additional Study

5.3.1 Method

I conducted an additional user study to clarify the following two points.

- If the same operation is repeated, is it a positive or negative motive?
- If the theme is clear, how will the operation time change?

I sampled 6 participants randomly from among the participants in the study in Section 5.2 to serve as additional participants. The order of the method and the order of the tasks are equally rotated among the participants alike the study in Section 5.2.

After the study was finished, I conducted detailed interviews were conducted to determine whether participants had negative or positive impressions through the operation. In addition, to evaluate the overall quality of the system, I asked how much the system was able to successfully express the differences between the two themes of characters at Likert 5 scale.

- Style : Open world RPG with toon graphics
- Task 1: A party leader who is reliable

- Task 2: A support member who is modest

The reason for defining the style as above is that it is suitable for anime-like avatars and also fits the trends of the training data. I defined Tasks 1 and 2 so that the avatars' looking tendencies were clearly separated.

In addition, after the user study was over, participants were interviewed in detail to confirm whether they had a negative or positive impression of the operation.

5.3.2 Result

The result is shown as Table 5.7. The number of the participant is the same as the study of 5.2.

Participant	Eval.	1-C (s)	1-H (s)	1-B (s)	2-C (s)	2-H (s)	2-B (s)
P1	5	78.5	84.5	46.3	28.1	80.2	49.1
P2	4	65.2	108.5	114.8	64.6	90.8	72.5
P4	5	64.9	86.0	90.7	47.9	85.5	99.8
P8	4	59.0	140.0	85.2	60.8	87.2	73.6
P10	2	38.3	101.5	37.4	81.1	168.3	66.8
P12	5	61.6	76.8	98.3	103.4	109.2	70.9

Table 5.7: The results of the additional study. Eval means the overall evaluation of the study (questionnaire answer). Each method columns means the duration time (s).

Parameter	Turn	Method	Turn \times Method
Duration	0.294	0.001*	0.298

Table 5.8: Table of ANOVA Probabilities of the additional study. Each column are conditions, Turn, Method, and their interaction. Turn condition means difference by 1st, 2nd, 3rd, 4th, 5th and 6th studies par a participant. Method condition means difference by C,H,B methods. * star means probabily under 1 % so it has significant difference.

The examples of the avatars the participants created are shown in Figure 5.11.

For Task 1 (Reliable Leader), I can see the parts such as dark eyes, high eyes, and a slender face so they have mature feeling and strong will. In Task 2 (modest supporter), the examples gave a childlike feel, from the parts such

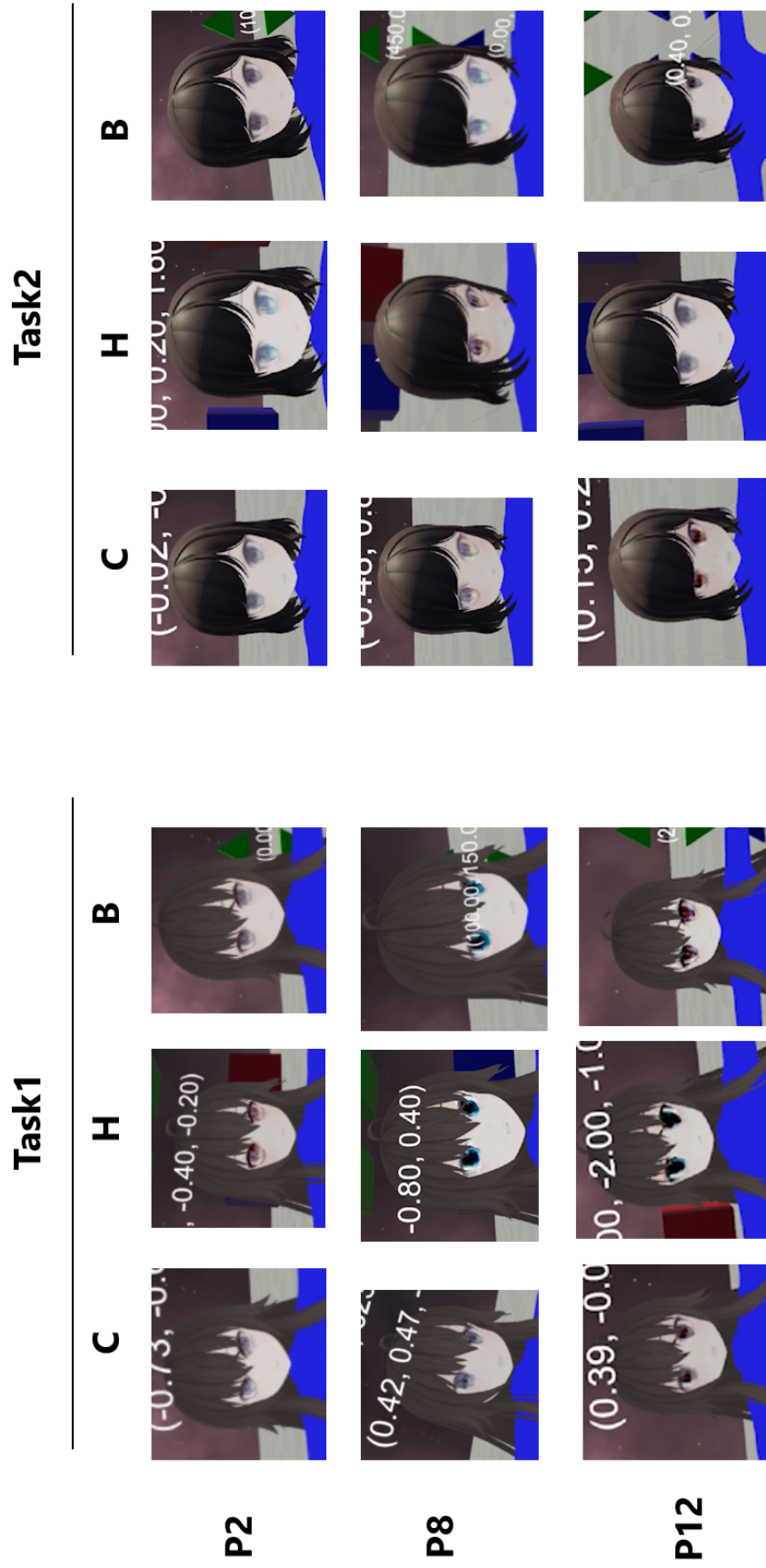


Figure 5.11: The examples of the created avatars by additional study.

as pale eyes, low eye position, and small faces. So, the system can create the two different types of avatar.

Regarding the overall evaluation, a high average rating of 4.2 was obtained. This showed that this system was able to successfully express two characters with different themes. Therefore, this system has shown a certain level of performance for the purpose of allowing users to create the avatar they want.

As with the user study in Section 5.2, the duration (operation time) tended to be shorter in C method. For 4 out of 6 participants, C method resulted the shortest operation time. The Average operation time of C method is the lowest and significant difference is proved by ANOVA with $p < 0.01\%$ (Table 5.8).

In this additional study, the theme has been narrowed down to a certain extent. So it is unlikely that people will give up on the pursuit midway because they are unsure of the final image. Therefore, the short operation time in C method is not an pseudo effect of giving up on a task midway through, but it does indeed have the effect of shortening the time even for tasks of completing the avatar as the participant like.

5.3.3 Detailed Interview

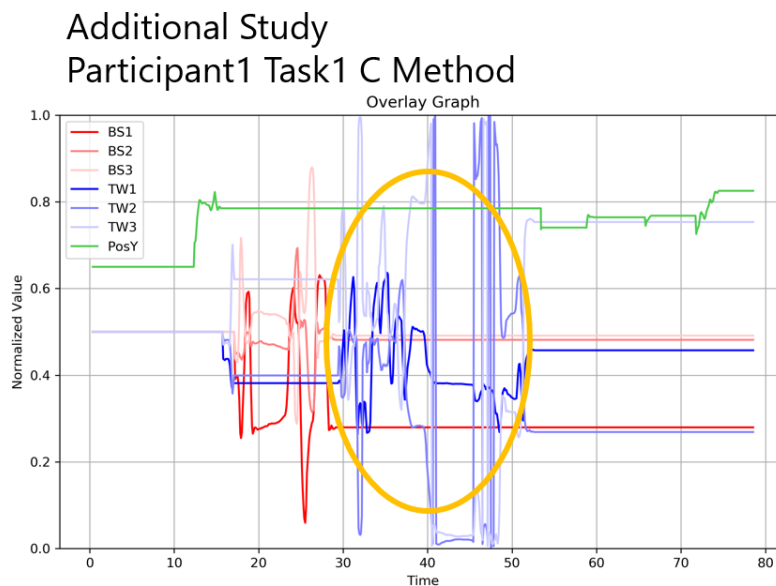


Figure 5.12: The Results of Participant 1 Task 1 C method on the additional study. The intensively operation areas are marked by yellow circle.

Additional Study Participant1 Task2 C Method

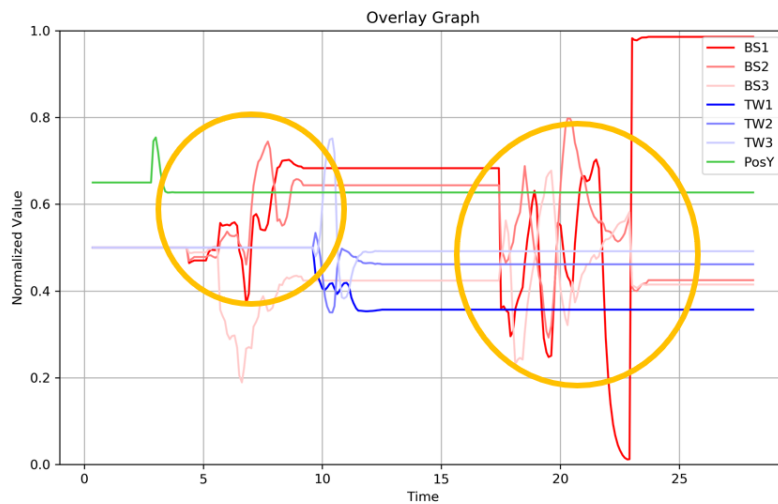


Figure 5.13: The Results of Participant 1 Task 2 C method on the additional study. The intensively operation areas are marked by yellow circle.

For example, for Participant 1, in Task 1 (C method), a fairly long (20 s) short-term revision was observed in the texture manipulation (Figure 5.12). The reason for this was that he wanted to see what kind of texture they would get if they moved it. There was no particular impression as to whether the operation was painful or not. In Task 2 (Method C), long-term revision was observed (Figure 5.13), like after changing the face shape, changing the texture, and then changing the face shape again. The reason for this was that after changing the texture, they wanted to change the shape of the face and see what it would look like.

Additional Study Participant8 Task2 C Method

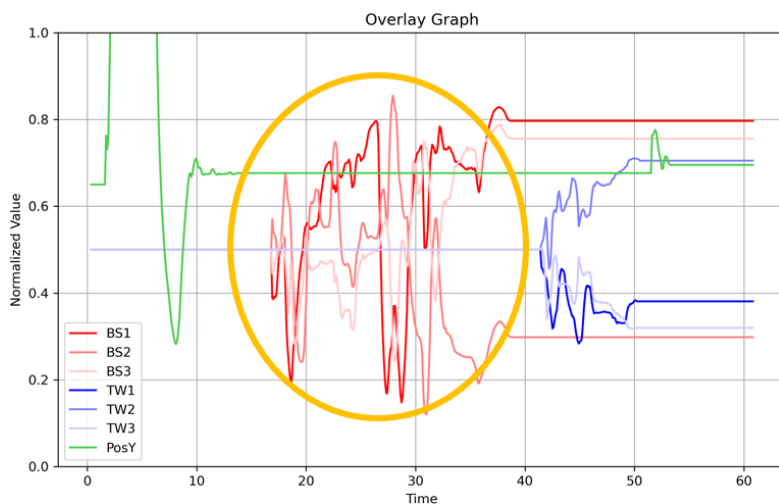


Figure 5.14: The Results of Participant 8 Task 2 C method on the additional study. The intensively operation areas are marked by yellow circle.

Additional Study Participant8 Task2 H Method

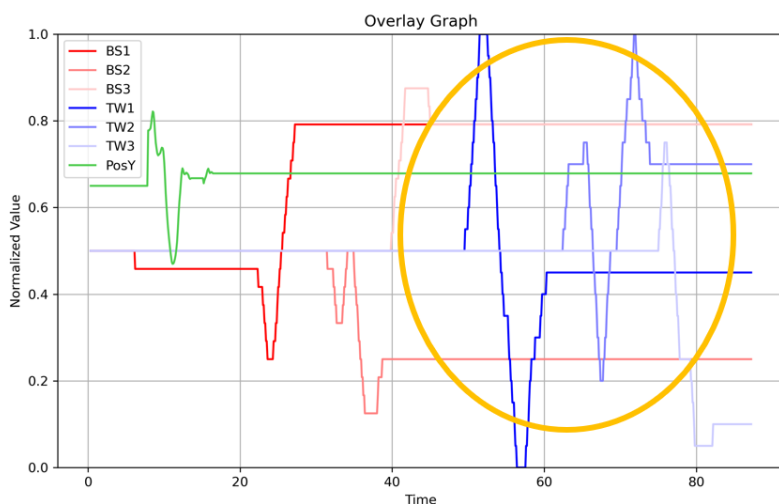


Figure 5.15: The Results of Participant 8 Task 2 H method on the additional study. The intensively operation areas are marked by yellow circle.

As for Participant 8, short-term revisions were observed in Task 2 (H method) and Task 2 (C method) (Figure 5.14 and 5.15). The reason for

this is that he was in the process of trial and error learning how to operate each device, and there was no intention to check the behavior under different conditions. Regarding how they felt about the operation, the respondents said that while it was difficult to understand the correspondence between parameters using H method, it was fun using C method because changing parameters was completed in an instant.

An interesting point is that C method, whose relationship with parameters is more like a black box and is difficult to adjust separately, received a more positive evaluation than H method, in which individual parameters are explicitly displayed. It is thought that this participant did not feel stressed because the enjoyment of receiving a reaction immediately outweighed the need to think about the correspondence with the parameters. This suggests that the instant nature of input and immediate response is more important than the obviousness of the correspondence in determining whether users find the operation enjoyable or not. In short, the quickness may come first.

Regarding long-term revisions, I obtained the following other feedback through in-depth interviews:

- When I tried changing the texture, I came across something that I wanted to fine-tune (one other person gave a similar answer)
- It was just an operational error, and there was no particular intention.

Regarding short-term revisions, I received the following feedback:

- I thought it was like a tutorial, so I didn't feel anything.
- Parameter input depends on wrist rotation, making it difficult to operate near the limits of the range of motion.
- It was difficult and time-consuming to get the desired color for the facial texture.

Long-term revisions were positive, as most users expressed curiosity to see different behavior under different conditions. From this, it can be said that long-term revision is a creative act and encourages users' creativity. But in short-term revisions, there were positive and negative evaluations. This tends to give a negative impression when users are forced to perform physically taxing input or when real-time feedback cannot be obtained. On the contrary, it was found that when comfortable input is guaranteed, it tends to give a positive impression.

This additional study confirmed the following facts.

- This system is effective for users to create the avatars they like.
- Even if the task theme becomes clear, the time-shortening effect of C Method remains the same.
- Long-term revision is a creative act.
- Whether users can enjoy short-term revisions strongly depends on the comfort of the interface, especially its immediacy.

Chapter 6

Conclusion

In this chapter, I discuss the conclusions of the methods I have developed, the user studies I have conducted, and the contributions I have made. I will also summarize their novelty, superiority, and advanced nature, and their contribution to knowledge science.

6.1 Data Set Generation

As shown in Chapter 3, this research proposes a method for generating a data set of 3D data for avatars, which will serve as alter egos in a metaverse that is expected to develop in the future. Unlike realistic avatars, anime-like avatars have a wide range of exaggerated features. It has been difficult to learn their details from data with different topology, but in this study, I established an independent data set generation method. For meshes, I was able to learn their features while keeping a common topology by repeating Subdivision Shrink after template matching. For textures, template matching and UV inversion based on rendering results and UV maps were used to extract features while keeping the same topology. This demonstrated that the mesh and texture features of the training data could be extracted while maintaining a high level of detail. I also demonstrated that it is possible to freely distribute them to generate new models and to use PCA to generate a wider range of models.

6.2 Application and User Study

In this study, I used the data set generated based on the proposed method to create an application that generates avatars within VR space.

As shown in Chapter 4, Application 1 enabled the participants to freely manipulate facial and body features. The user study was conducted with 10 participants, who were asked to create avatars using our system and compare the experience with that of a traditional 2D interface. The results of the user study revealed that the VR-based UI has the advantage of requiring less time for modeling. I also revealed that the appropriate level of manipulation for VR device is global manipulation, while fine-tuning at a local scale still gives a significant advantage to the 2D-UI[107] [108].

As shown in Chapter 5, In Application 2, the participants can manipulate the facial mesh and facial texture. The user study was conducted with 12 participants. 2 synthesis methods and 3 operation methods were proposed to examine the appropriate interface within VR space. The results of the user study showed that most of the participants highly valued the method using only the controller as the operation method. This indicated that designing a physical UI using only the controller within the VR space would result in higher user satisfaction.

From these results, I can infer that when using VR devices to operate a large number of parameters, global parameter adjustment is appropriate. Furthermore, the manipulation should be executed physically through the controller. However, this is strongly dependent on the performance of the VR device, and this detail will be discussed in the next section.

6.3 Limitations and Further Study

6.3.1 Training Data

This study used $N = 40$ 3D models as data. This is a small number for training purposes, but budgetary considerations forced me to settle on this number. A study using a larger number of data sets could be considered as a further study. "However, this would require a substantial budget, which might not be at a realistic level. Therefore, it may be important to develop some means of data augmentation to improve the accuracy of the proposed method.

Copyright considerations are also a factor that should not be overlooked in scaling up the analysis. The current success of machine learning in 2D images is largely due to the ability to use large amounts of out-of-copyright data, public domain data, and data from illustration SNS. Access to 3D training data is limited and the copyright restrictions accompany the selling data. When it becomes common to customize learning models in the future, consideration must be given to whether they can be released and sold. In

addition, as mentioned in the section 2.1, 3D models have a wide range of data domains, and it is natural to assume that there will be a division of work. It is possible to have different authors for mesh data, texture data, rigging data, and animation data, and so on. In the metaverse era future, it will be essential to develop copyright laws and practices for the various elements of the 3D model.

6.3.2 Landmark Auto Estimation

In addition, the proposed method requires manual adjustments in template matching. In this study, the adjustment of 90 landmark points took less than one minute per landmark, which is a realistic amount of time. However, it is expected to become a non-negligible effort as the data set will expanded in the future, so automation of template matching by automatic landmark estimation would be effective in enhancing the proposed method.

6.3.3 Facial Expression

As mentioned in the subsection 2.2, not only body shape and gestures, also facial expressions play a major role in communication. So, most 3D avatar models have the ability to change their facial expressions. They are implemented in the form of shape keys, which are also topology dependent. Furthermore, labeling which expression to assign to which shape key is not unified. Typical emotions such as anger and joy are often created as pre-set templates, but these interpretations vary widely, making it difficult to perform a unified analysis.

Another possible method would be to recognize user facial expressions by camera and deform bones to match the facial expressions. However, this requires a process of rigging the mouth and eyes and making them work together according to the facial expression. In the case of fixed models, it is possible to make them work together naturally through precise adjustment by an expert, but when the meshes are expanded or contracted as in this study, the meshes may intersect with each other, so it is difficult to completely prevent natural behavior. Synthesizing facial expressions and applying them to various avatars is another challenging issue.

6.3.4 Analysis Method

This study used PCA for feature extraction. This is a powerful method in extracting a large number of parameters into a small number of features, but it has the weakness that it can only perform linear transformations. The

complex features of anime-like avatars often have nonlinear transformations, and the features obtained using PCA are expected to be smoothly averaged. However, there is a trade-off with the increase in computational complexity. The use of deep learning involves huge tensor computations, and even when the computation is accelerated by GPGPU, the conversion from CPU to GPU involves overhead. Instant reaction times are essential for interactive interfaces, and time lags, even at the 0.1 second level, can fatally damage the user experience. Therefore, in order to apply deep learning to interactive UI, it is necessary to radically accelerate the computation, including the overhead.

6.3.5 Better VR Devices

In this research application, HTC Vive was used as the VR device. As mentioned in the discussion of the user study, it is impossible to receive detailed information such as finger movements and grip force with this device. In addition, other devices that are compatible with them cannot provide such a high level of accuracy. Therefore, in the user study of this research, the method that reproduces conventional mechanical devices on VR received a low evaluation. However, as device performance develops in the future, the optimal solution will be changed.

As a transitional manner, it is also possible to combine hardware other than VR controllers to enable physical input. For example, a device such as a foot pedal used by a musician during a performance is an example of a device that allows reliable input even when the user's sight is blocked while the user's movement is intense. Such combination of VR devices and other hardware can also be an interesting topic for human computer interaction.

6.3.6 Appropriate Guide and Environment

When building an interface on a VR space, whether in the form of a controller-only method or one that emulates a conventional mechanical device, it is impossible to create an efficient interface with buttons densely spaced a few centimeters apart, such as a keyboard or piano. Therefore, it is necessary to create pseudo-buttons that are large in scale or convert physical actions into input. However, without proper guidance, these devices may force unnatural actions or cause interference between UIs, and thus become unintuitive. Therefore, careful guides are needed that consider the spatial arrangement and make it well known to the user. With such a guide in place, it will be possible to better isolate what users enjoy and what they find stressful.

Additionally, as mentioned in Section 2.2, VR can realize a variety of work environments than traditional software. So, in a VR environment, the difference of environments may change the avatar users want to make, for example, as users create an avatar like a life-sized sculpture while kneading clay or as user holds it in the palm of your hand. In this study, I fixed the work environment, but simulating various production environments may affect users' creativity. When designing an appropriate VR interface, it would be useful to investigate how the environment created by VR interacts with the user's experience.

6.3.7 Hair Style

In this reserach, hairstyles were excluded from data set generation, analysis, and generation. In user study of Chapter 4, hairstyles were fixed, and in user study of Chapter 5, three different hairstyles were toggled by user selection. This is due to the fact that hairstyles have various variations that are not confined to the bone hierarchy. With this proposed method, which assumes a common template in different 3D training data, it is difficult to properly represent hairstyles with such variations. However, hairstyle is a major factor that determines the avatar's personality. If avatars are used more frequently in daily life, it is conceivable that users will want to change their avatars' hairstyles according to their moods, just as they do with their hair in the real world. In the photo-real field, there are limitations to the expressiveness of the hairstyles that can be created, such as those based on generative models [109] or automatically generated hairstyles based on analysis of portraits [110]. Also, in the anime-like field, there are limitations. When a method is developed that can generate a wide range of hair meshes of anime-like avatars, including super deformed and overly decorative elements, the automatic generation of avatars will take a step to the higher level.

6.4 Contribution for Knowledge Science

I would like to discuss how this research resulted contribute to knowledge science.

As mentioned in the introduction, in a metaverse society, it is common for users to belong to multiple communities during their social lives. Furthermore, even within the same community, people begin to strategically choose different avatars to adjust their impressions according to the situation. This is the same as people use appropriate clothing and accessories depending on

the TPO in the real world.

In such a society, not only realistic avatars based on actual bodies but also more fictional anime-like avatars are often used. Unlike photo-realistic avatars, anime-like avatars have a wide variety of expressions. There, one can idealize yourself, ignore its actual gender, or become something other than human, such as a semi-beast. In this way, transforming into a different shape not only has strategic significance in terms of human relationships, but also serves to improve one's self consciousness and human relationships in a more desirable direction.

For example, someone who feels that their body is ugly may be able to use a beautiful avatar. Someone who feels uncomfortable about their sexuality can swap it. In a metaverse society, obstacles faced by one's own body can be overcome by having an avatar. It is also possible that the awareness of weaknesses and trauma will be improved, leading to personal growth. This is human augmentation, and as a metaverse society becomes widespread, the expansion of human experience will have a major impact on knowledge science.

However, as mentioned in Section 2, efficiently generating anime-like avatars is an extremely technically difficult task at present. In this research, I aimed to solve these points from the aspects of data analysis and interface implementation. I succeeded in establishing a generic data set analysis method. In addition, I conducted quantitative analysis and quantitative detailed examination of the advantages and disadvantages of VR devices through user studies. As a result, I created an effective prototype system for an avatar creation system for anime-like avatars, and indicated the direction towards a more user-friendly VR interface. This makes it easier to create anime-like avatars, which are one of the main means of communication in a metaverse society, so promotes communication in the future. This will be the technical basis of a society that improves human cognitive ability and realizes human augmentation, as described above. Creating such a technical basis is the contribution of my research to knowledge science.

Bibliography

- [1] Stylianos Mystakidis. Metaverse. *Encyclopedia*, 2(1):486–497, 2022.
- [2] Neal Stephenson. *Snow Crash: A Novel (English Edition)*. Spectra, 8 2003.
- [3] Susumu Tachi, Makoto Sato, and Michitaka Hirose. バーチャルリアリティ学. コロナ社, 12 2010.
- [4] Rebecca Hetrick, Nicholas Amerson, Boyoung Kim, Eric Rosen, Ewart J de Visser, and Elizabeth Phillips. Comparing virtual reality interfaces for the teleoperation of robots. In *2020 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–7. IEEE, 2020.
- [5] Roy A Ruddle, Stephen J Payne, and Dylan M Jones. Navigating large-scale virtual environments: what differences occur between helmet-mounted and desk-top displays? *Presence*, 8(2):157–168, 1999.
- [6] Beatriz Sousa Santos, Paulo Dias, Angela Pimentel, Jan-Willem Baggerman, Carlos Ferreira, Samuel Silva, and Joaquim Madeira. Head-mounted display versus desktop for 3d navigation in virtual reality: a user study. *Multimedia tools and applications*, 41:161–181, 2009.
- [7] Barbara Olasov Rothbaum, Larry Hodges, Samantha Smith, Jeong Hwan Lee, and Larry Price. A controlled study of virtual reality exposure therapy for the fear of flying. *Journal of consulting and Clinical Psychology*, 68(6):1020, 2000.
- [8] Maryrose Gerardi, Barbara Olasov Rothbaum, Kerry Ressler, Mary Heekin, and Albert Rizzo. Virtual reality exposure therapy using a virtual iraq: case report. *Journal of Traumatic stress: official Publication of The International society for Traumatic stress studies*, 21(2):209–213, 2008.

- [9] Wolfgang Kruger, C-A Bohn, Bernd Frohlich, Heinrich Schuth, Wolfgang Strauss, and Gerold Wesche. The responsive workbench: A virtual work environment. *Computer*, 28(7):42–48, 1995.
- [10] David M. Krum, Sin-Hwa Kang, Thai Phan, Lauren Cairco Dukes, and Mark Bolas. Social impact of enhanced gaze presentation using head mounted projection. In *Distributed, Ambient and Pervasive Interactions: 5th International Conference, DAPI 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9–14, 2017, Proceedings*, page 61–76, Berlin, Heidelberg, 2017. Springer-Verlag.
- [11] Christian R Larsen, Jette L Soerensen, Teodor P Grantcharov, Torur Dalsgaard, Lars Schouenborg, Christian Ottosen, Torben V Schroeder, and Bent S Ottesen. Effect of virtual reality training on laparoscopic surgery: randomised controlled trial. *Bmj*, 338, 2009.
- [12] Miguel Barreda-Ángeles and Tilo Hartmann. Psychological benefits of using social virtual reality platforms during the covid-19 pandemic: The role of social and spatial presence. *Computers in Human Behavior*, 127:107047, 2022.
- [13] Sanni Siltanen, Hanna Heinonen, Alisa Burova, Paulina Becerril Palma, Phong Truong, Viveka Opas, and Markku Turunen. There is always a way: Organizing vr user tests with remote and hybrid setups during a pandemic—learnings from five case studies. *Multimodal Technologies and Interaction*, 5(10):62, 2021.
- [14] Organisation for Economic Co-operation and Development. *Teleworking in the COVID-19 pandemic: trends and prospects*. OECD Publishing Paris, 2021.
- [15] 経済産業省. 【報告書】令和2年度コンテンツ海外展開促進事業（仮想空間の今後の可能性と諸課題に関する調査分析事業）, 2021. https://www.meti.go.jp/policy/mono_info_service/contents/downloadfiles/report/kasou-houkoku.pdf.
- [16] Mel Slater, Daniel Pérez Marcos, Henrik Ehrsson, and Maria V Sanchez-Vives. Inducing illusory ownership of a virtual body. *Frontiers in neuroscience*, page 29, 2009.
- [17] Konstantina Kilteni, Raphaela Groten, and Mel Slater. The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments*, 21(4):373–387, 2012.

- [18] Domna Banakou, Raphaela Groten, and Mel Slater. Illusory ownership of a virtual child body causes overestimation of object sizes and implicit attitude changes. *Proceedings of the National Academy of Sciences*, 110(31):12846–12851, 2013.
- [19] Brian Ries, Victoria Interrante, Michael Kaeding, and Lee Anderson. The effect of self-embodiment on distance perception in immersive virtual environments. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 167–170, 2008.
- [20] Young June Sah, Rabindra Ratan, Hsin-Yi Sandy Tsai, Wei Peng, and Issidoros Sarinopoulos. Are you what your avatar eats? health-behavior effects of avatar-manifested self-concept. *Media Psychology*, 20(4):632–657, 2017.
- [21] Kim Szolin, Daria J Kuss, Filip M Nuyens, and Mark D Griffiths. Exploring the user-avatar relationship in videogames: A systematic review of the proteus effect. *Human-Computer Interaction*, 38(5-6):374–399, 2023.
- [22] 陽光 小柳, 拓志 鳴海, and 廉 大村. ソーシャル VR コンテンツにおける普段使いのアバタによる身体所有感と体験の質の向上. *日本バーチャルリアリティ学会論文誌*, 25(1):50–59, 2020.
- [23] Zhicong Lu, Chenxinran Shen, Jiannan Li, Hong Shen, and Daniel Wigdor. More kawaii than a real-person live streamer: understanding how the otaku community engages with and perceives virtual youtubers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2021.
- [24] Hsin Lin and Hua Wang. Avatar creation in virtual worlds: Behaviors and motivations. *Computers in Human Behavior*, 34:213–218, 2014.
- [25] Nicolas Ducheneaut, Ming-Hui Wen, Nicholas Yee, and Greg Wadley. Body and mind: a study of avatar personalization in three virtual worlds. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1151–1160, 2009.
- [26] Carman Neustaedter and Elena A Fedorovskaya. Presenting identity in a virtual world through avatar appearances. In *Graphics Interface*, pages 183–190, 2009.

- [27] Tetsuya Matsui. Power of gijinka: Designing virtual teachers for ecosystem conservation education. In *Proceedings of the 9th International Conference on Human-Agent Interaction*, pages 328–331, 2021.
- [28] 遠藤雅伸 寺澤弘騎. アバター作成においてジェンダースワップする理由に関する研究. In *日本デジタルゲーム学会 2015年度年次大会*, 2016.
- [29] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.
- [30] Hans-Gerd Maas and Uwe Hampel. Photogrammetric techniques in civil engineering material testing and structure monitoring. *Photogrammetric Engineering & Remote Sensing*, 72(1):39–45, 2006.
- [31] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4):1–14, 2015.
- [32] Zhe Li, Tao Yu, Chuanyu Pan, Zerong Zheng, and Yebin Liu. Robust 3d self-portraits in seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1344–1353, 2020.
- [33] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2252–2261, 2019.
- [34] Tianyang Shi, Yi Yuan, Changjie Fan, Zhengxia Zou, Zhenwei Shi, and Yong Liu. Face-to-parameter translation for game character auto-creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 161–170, 2019.
- [35] Nancy Kanwisher, Josh McDermott, and Marvin M Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of neuroscience*, 17(11):4302–4311, 1997.
- [36] Sepp Hochreiter and Jürgen Schmidhuber. Lstm can solve hard long time lag problems. *Advances in neural information processing systems*, 9, 1996.
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

- [38] G Matrone, C Cipriani, EL Secco, MC Carrozza, and G Magenes. Bio-inspired controller for a dexterous prosthetic hand based on principal components analysis. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5022–5025. IEEE, 2009.
- [39] Giulia C Matrone, Christian Cipriani, Maria Chiara Carrozza, and Giovanni Magenes. Real-time myoelectric control of a multi-fingered hand prosthesis using principal components analysis. *Journal of neuroengineering and rehabilitation*, 9(1):1–13, 2012.
- [40] VP Kshirsagar, MR Baviskar, and ME Gaikwad. Face recognition using eigenfaces. In *2011 3rd International Conference on Computer Research and Development*, volume 2, pages 302–306. IEEE, 2011.
- [41] 友博 日比野, 浩然 謝, and 一乘 宮田. 機械学習による特徴量を用いた顔画像選択支援システム. In **第 177 回 CG 研究発表会**, 2020.
- [42] Romain Lopez, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Information constraints on auto-encoding variational bayes. *Advances in neural information processing systems*, 31, 2018.
- [43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [44] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pages 2745–2754, 2017.
- [45] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.
- [46] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11304–11314, 2022.

- [47] Wenju Xu, Chengjiang Long, Ruisheng Wang, and Guanghui Wang. Drb-gan: A dynamic resblock generative adversarial network for artistic style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6383–6392, 2021.
- [48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [49] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [50] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [51] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.
- [52] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.
- [53] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models, 2023.
- [54] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023.
- [55] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023.

- [56] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *ACM Transactions on Graphics (TOG)*, 41(4):1–19, 2022.
- [57] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016.
- [58] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)*, 36(4):1–11, 2017.
- [59] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [60] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020.
- [61] Timur Bagautdinov, Chenglei Wu, Jason Saragih, Pascal Fua, and Yaser Sheikh. Modeling facial geometry using compositional vaes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2018.
- [62] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1886–1895, 2018.
- [63] Qixing Huang, Hai Wang, and Vladlen Koltun. Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph.*, 34(4):87–1, 2015.
- [64] Kar Abhishek, Tulsiani Shubham, Carreira Joao, and Malik Jitendra. Category-specific object reconstruction from a single image. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2015.

- [65] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020.
- [66] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [67] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [68] M. Joseph-Rivlin, A. Zvirin, and R. Kimmel. Momenet: Flavor the moments in learning to classify shapes. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4085–4094, Los Alamitos, CA, USA, oct 2019. IEEE Computer Society.
- [69] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3323–3332, 2019.
- [70] Ruisheng Wang, Jiju Peethambaran, and Dong Chen. Lidar point clouds to 3-d urban models : a review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(2):606–627, 2018.
- [71] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018.
- [72] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019.
- [73] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

- [74] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.
- [75] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [76] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021.
- [77] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021.
- [78] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021.
- [79] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022.
- [80] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1274–1283, 2017.
- [81] Aaron Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. 03 2017.
- [82] Takayuki Niki and Takashi Komuro. Semi-automatic creation of an anime-like 3d face model from a single illustration. In *2019 International Conference on Cyberworlds (CW)*, pages 53–56. IEEE, 2019.

- [83] Ruizhe Li, Masanori Nakayama, and Issei Fujishiro. Automatic generation of 3d natural anime-like non-player characters with machine learning. In *2020 International Conference on Cyberworlds (CW)*, pages 110–116. IEEE, 2020.
- [84] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2021.
- [85] Karan Ahuja, Eyal Ofek, Mar Gonzalez-Franco, Christian Holz, and Andrew D Wilson. Coolmoves: User motion accentuation in virtual reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–23, 2021.
- [86] Yu-Ting Cheng, Timothy K Shih, and Chih-Yang Lin. Create a puppet play and interactive digital models with leap motion. In *2017 10th International Conference on Ubi-media Computing and Workshops (Ubi-Media)*, pages 1–6. IEEE, 2017.
- [87] Ching-Wen Hung, Ruei-Che Chang, Hong-Sheng Chen, Chung Han Liang, Liwei Chan, and Bing-Yu Chen. Puppeteer: Exploring intuitive hand gestures and upper-body postures for manipulating human avatar actions. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11, 2022.
- [88] Grigore C Burdea. Keynote address: haptics feedback for virtual reality. In *Proceedings of international workshop on virtual prototyping. Laval, France*, pages 87–96, 1999.
- [89] Gabriel Robles-De-La-Torre. The importance of the sense of touch in virtual and real environments. *Ieee Multimedia*, 13(3):24–30, 2006.
- [90] Robert J Stone. Haptic feedback: A brief history from telepresence to virtual reality. In *International Workshop on Haptic Human-Computer Interaction*, pages 1–16. Springer, 2000.
- [91] Cecilie Våpenstad, Erlend Fagertun Hofstad, Thomas Langø, Ronald Mårvik, and Magdalena Karolina Chmarra. Perceiving haptic feedback in virtual reality simulators. *Surgical endoscopy*, 27:2391–2397, 2013.
- [92] Kairyu Mori, Masayuki Ando, Kouyou Otsu, and Tomoko Izumi. Effect of repulsive positions on haptic feedback on using a string-based device virtual objects without a real tool. In Jessie Y. C. Chen and Gino

- Fragomeni, editors, *Virtual, Augmented and Mixed Reality*, pages 266–277, Cham, 2023. Springer Nature Switzerland.
- [93] Robert JK Jacob, Hiroshi Ishii, Gian Pangaro, and James Patten. A tangible interface for organizing information using a grid. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 339–346, 2002.
- [94] Hiroshi Ishii. Tangible bits: beyond pixels. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages xv–xxv, 2008.
- [95] Daniel Harley, Aneesh P Tarun, Daniel Germinario, and Ali Mazalek. Tangible vr: Diegetic tangible objects for virtual reality narratives. In *Proceedings of the 2017 conference on designing interactive systems*, pages 1253–1263, 2017.
- [96] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.
- [97] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository, 2015. <https://shapenet.org/>.
- [98] 四條亮太, 櫻井翔, 広田光一, and 野嶋琢也. 獣耳の感情表現能力を用いた感情表現インターフェースの検討. *ヒューマンインタフェース学会論文誌*, 23(4):419–430, 2021.
- [99] Ryota Shijo, Sho Sakurai, Koichi Hirota, and Takuya Nojima. Research on the emotions expressed by the posture of kemo-mimi. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology*, pages 1–5, 2022.
- [100] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, aug 1987.
- [101] Un-Hong Wong, Hon-Cheng Wong, and Zesheng Tang. An interactive system for visualizing 3d human organ models. In *Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05)*, pages 6–pp. IEEE, 2005.

- [102] Pratomo Adhi Nugroho, Dwi Kurnia Basuki, and Riyanto Sigit. 3d heart image reconstruction and visualization with marching cubes algorithm. In *2016 International Conference on Knowledge Creation and Intelligent Computing (KCIC)*, pages 35–41. IEEE, 2016.
- [103] Edwin Catmull and James Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. In *Seminal graphics: pioneering efforts that shaped the field*, pages 183–188. 1998.
- [104] 友博 日比野, 浩然 謝, and 一乘 宮田. 3d アバターを対象としたトポロジー非依存のデータセット生成方法. *2023年度第51回画像電子学会年次大会ジャーナル*, 2023.
- [105] Thomas B Moeslund, Moritz Störring, and Erik Granum. A natural interface to a virtual environment through computer vision-estimated pointing gestures. In *Gesture and Sign Language in Human-Computer Interaction: International Gesture Workshop, GW 2001 London, UK, April 18–20, 2001 Revised Papers*, pages 59–63. Springer, 2002.
- [106] Sebastian Oberdörfer, David Heidrich, and Marc Erich Latoschik. Usability of gamified knowledge learning in vr and desktop-3d. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.
- [107] Tomohiro Hibino, Haoran Xie, and Kazunori Miyata. 機械学習による特徴量を用いたvr空間におけるインタラクティブアバター作成システム. In *第201回ヒューマンコンピュータインタラクション研究発表会*, 2023.
- [108] Tomohiro Hibino, Haoran Xie, and Kazunori Miyata. Interactive avatar creation system from learned attributes for virtual reality. In Jessie Y. C. Chen and Gino Fragomeni, editors, *Virtual, Augmented and Mixed Reality*, pages 168–181, Cham, 2023. Springer Nature Switzerland.
- [109] Lieu-Hen Chen, Santi Saeyor, Hiroshi Dohi, and Mitsuru Ishizuka. A system of 3d hair style synthesis based on the wisp model. *The Visual Computer*, 15:159–170, 1999.
- [110] Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. Autohair: Fully automatic hair modeling from a single image. *ACM Trans. Graph.*, 35(4), jul 2016.

Appendix A

Appendix

All results (trajectory figures and time plot figures) of the user study in the section 5.2 are shown. How to see the figures are noted in the section 5.2.

Participant 1

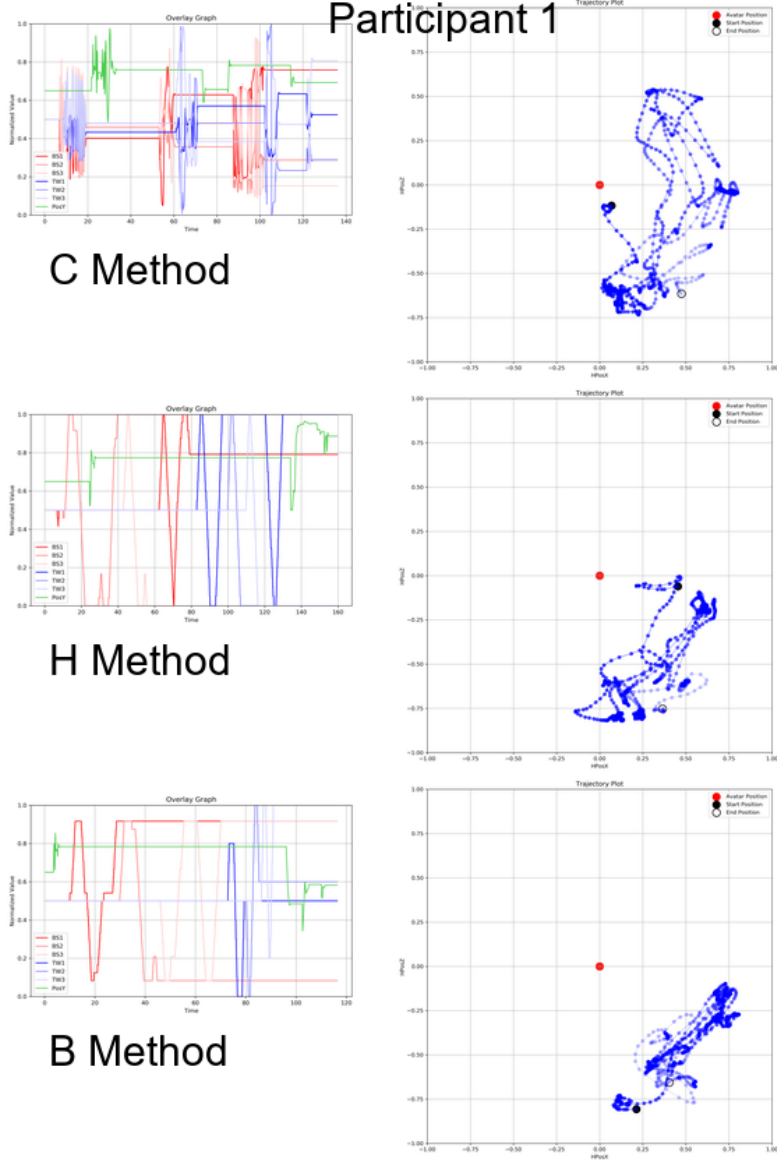


Figure A.1: Total results of participant 1 of the user study in Chapter 5

Participant 2

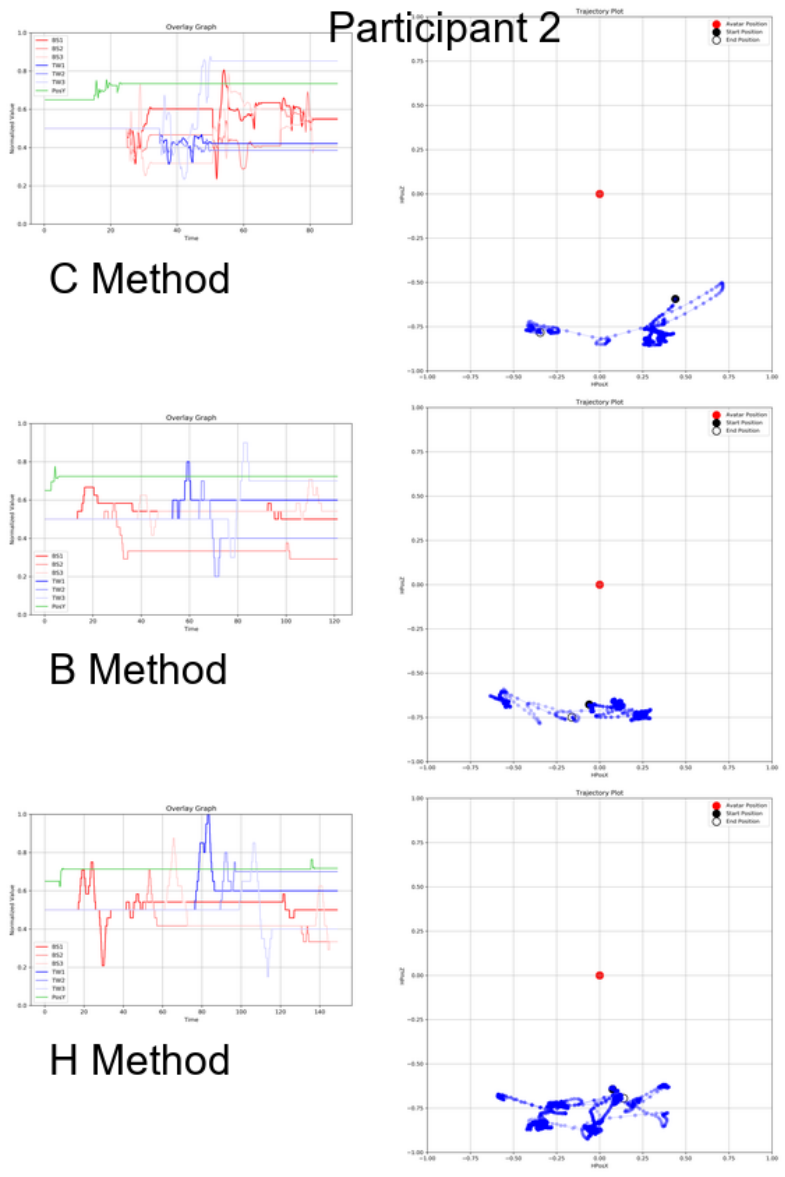


Figure A.2: Total results of participant 2 of the user study in Chapter 5

Participant 3

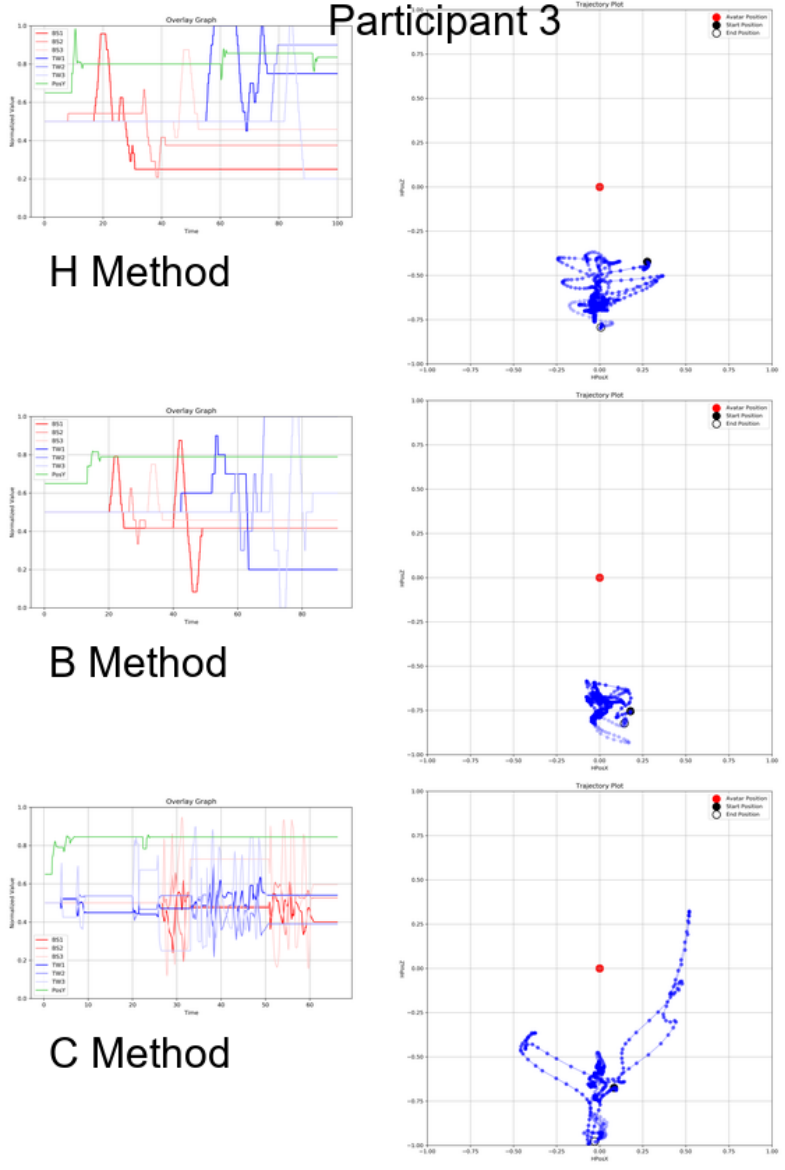


Figure A.3: Total results of participant 3 of the user study in Chapter 5

Participant 4

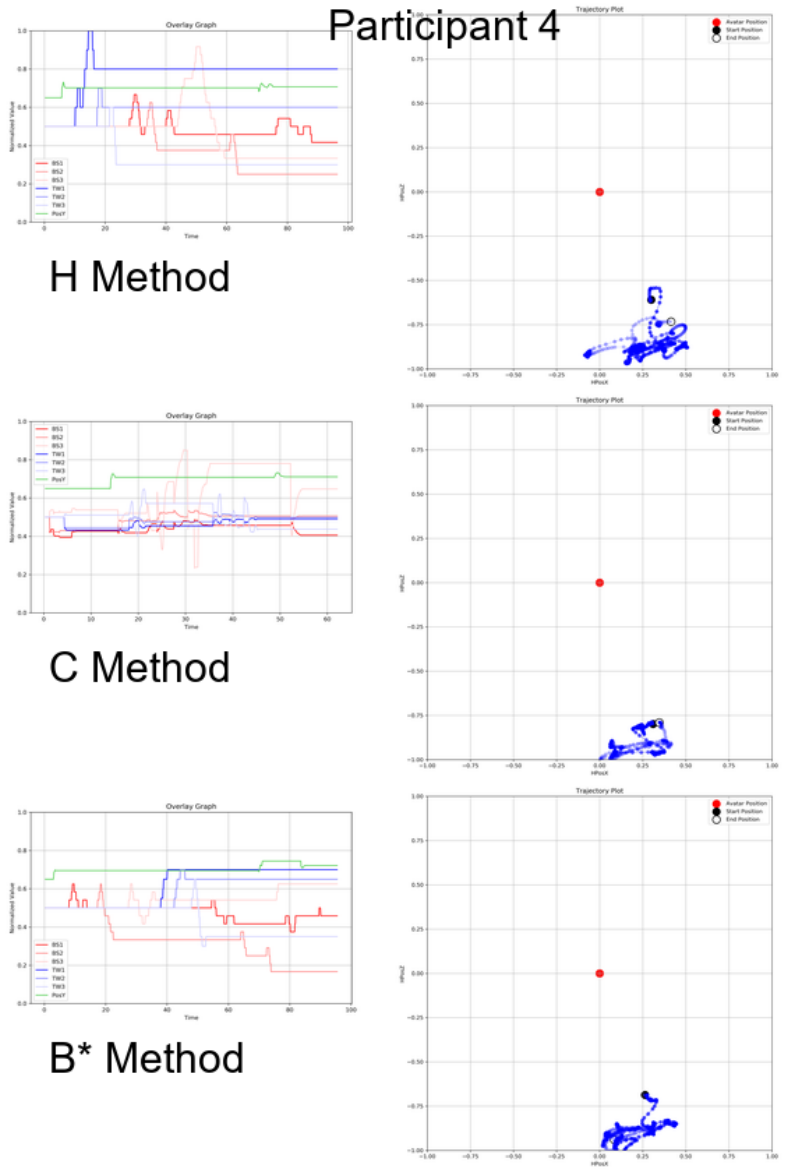


Figure A.4: Total results of participant 4 of the user study in Chapter 5

Participant 5

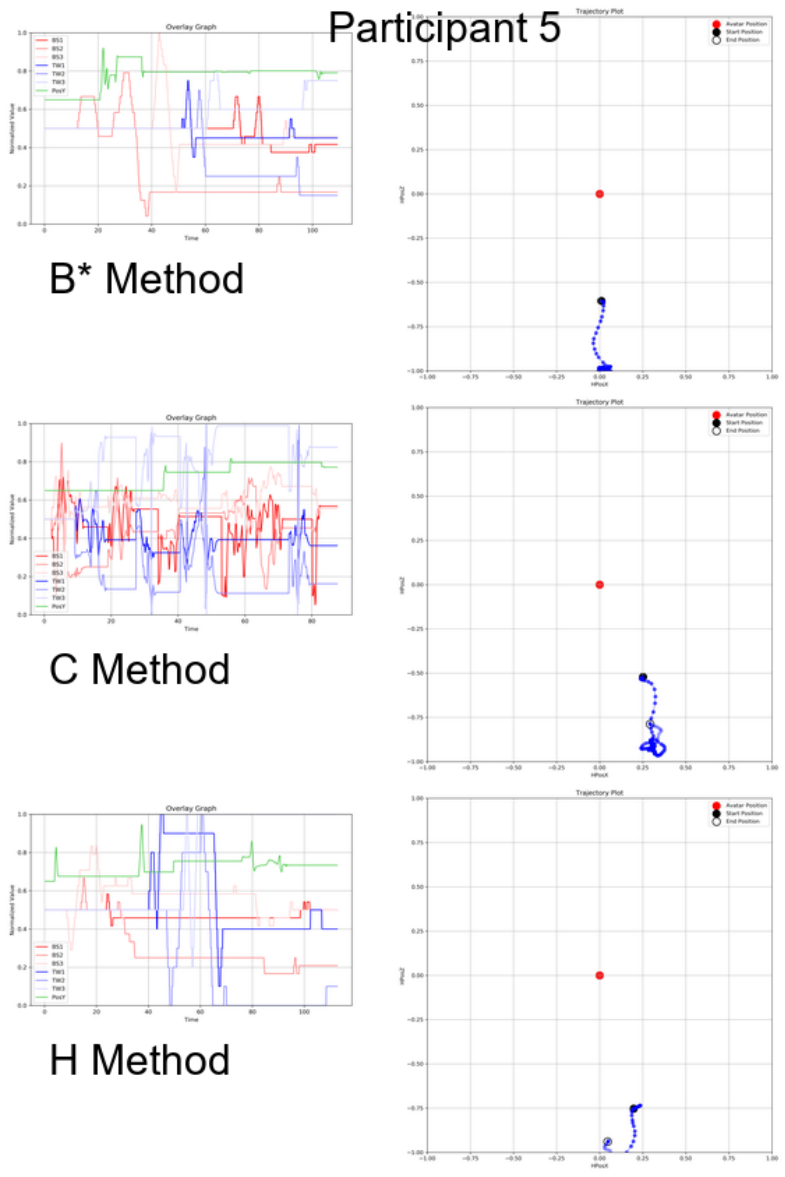
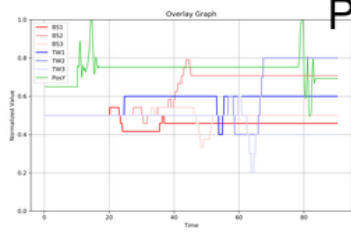
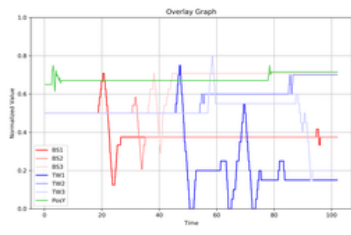
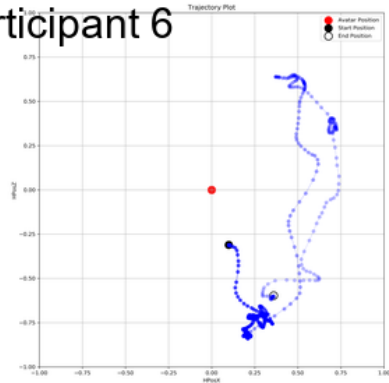


Figure A.5: Total results of participant 5 of the user study in Chapter 5

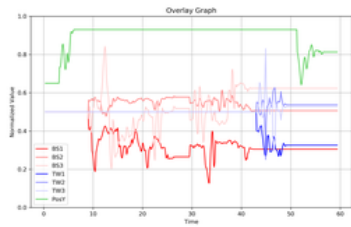
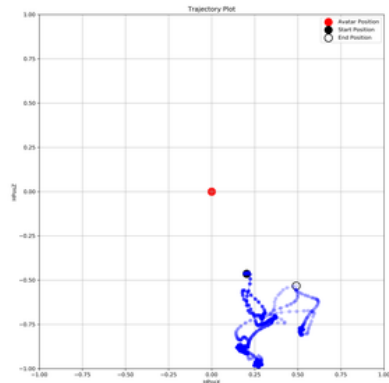
Participant 6



B* Method



H Method



C Method

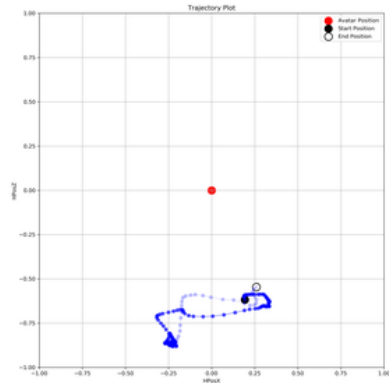


Figure A.6: Total results of participant 6 of the user study in Chapter 5

Participant 7

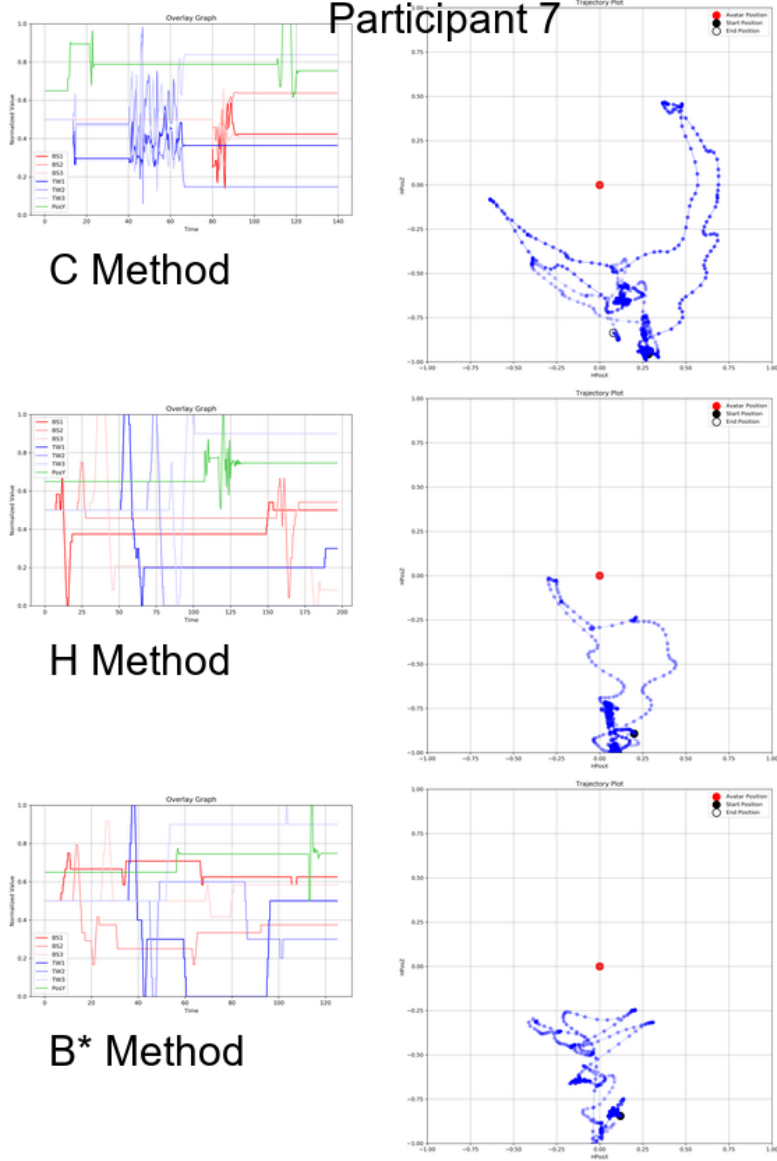
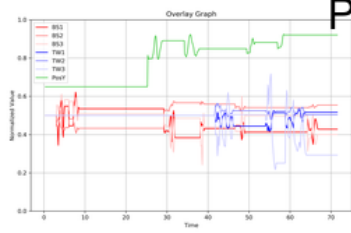
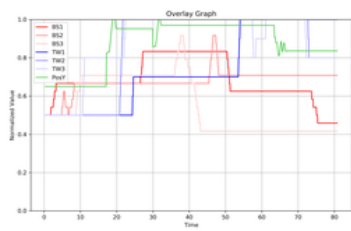
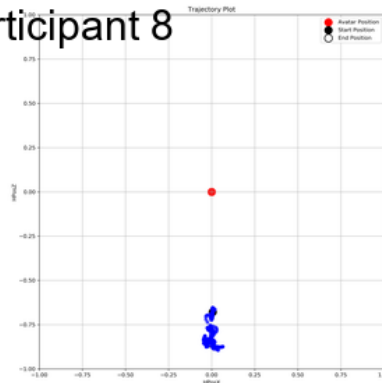


Figure A.7: Total results of participant 7 of the user study in Chapter 5

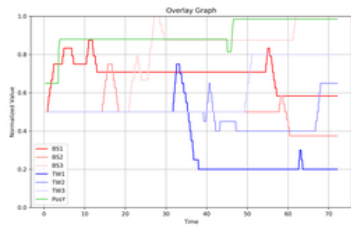
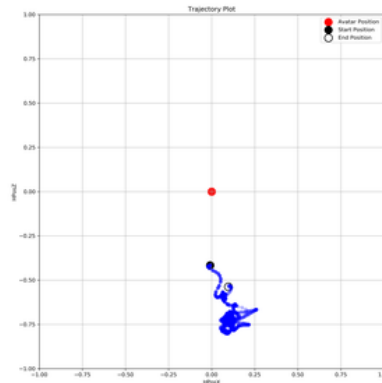
Participant 8



C Method



B* Method



H Method

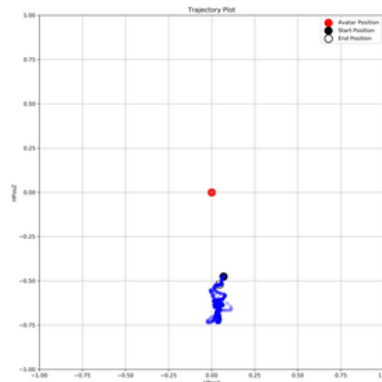


Figure A.8: Total results of participant 8 of the user study in Chapter 5

Participant 9

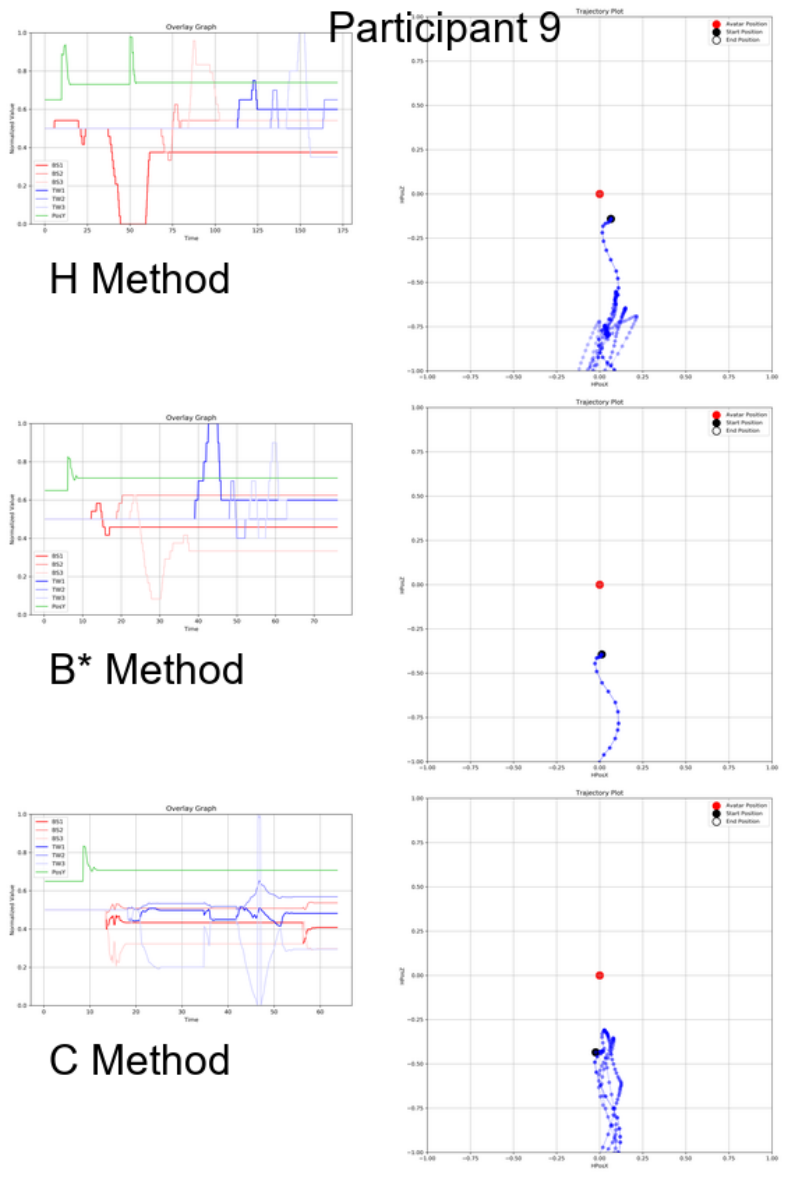


Figure A.9: Total results of participant 9 of the user study in Chapter 5

Participant 10

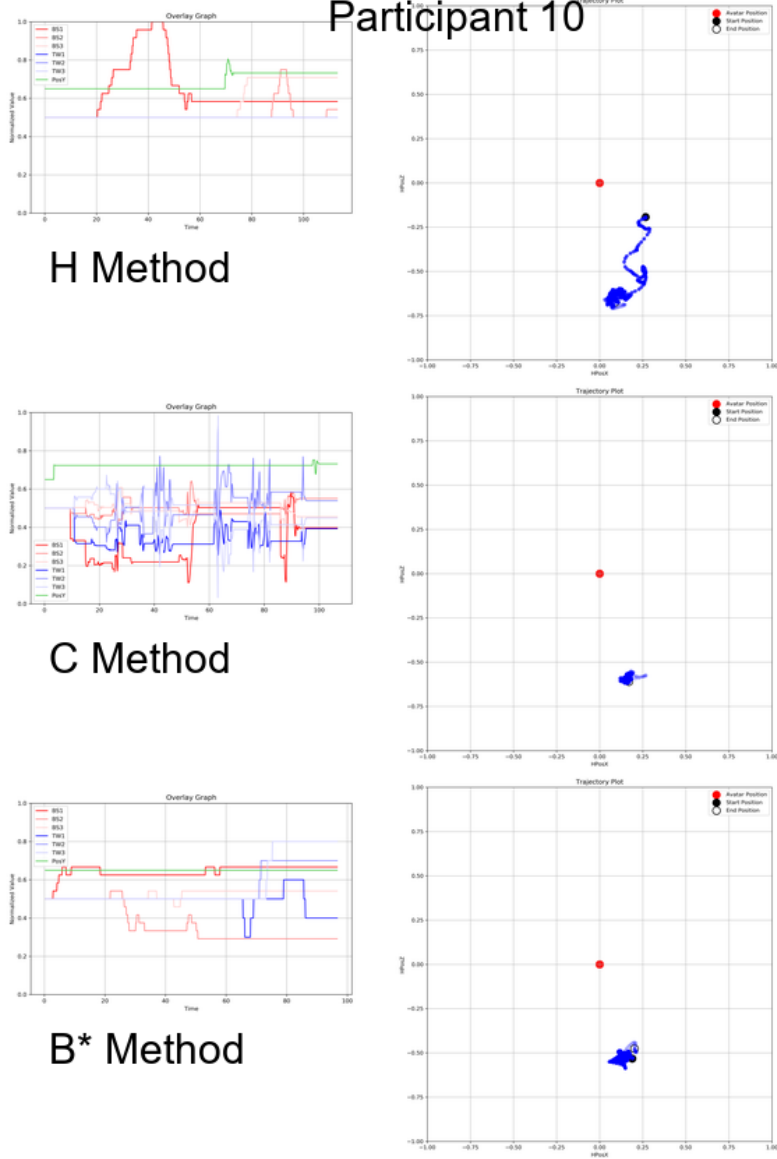


Figure A.10: Total results of participant 10 of the user study in Chapter 5

Participant 11

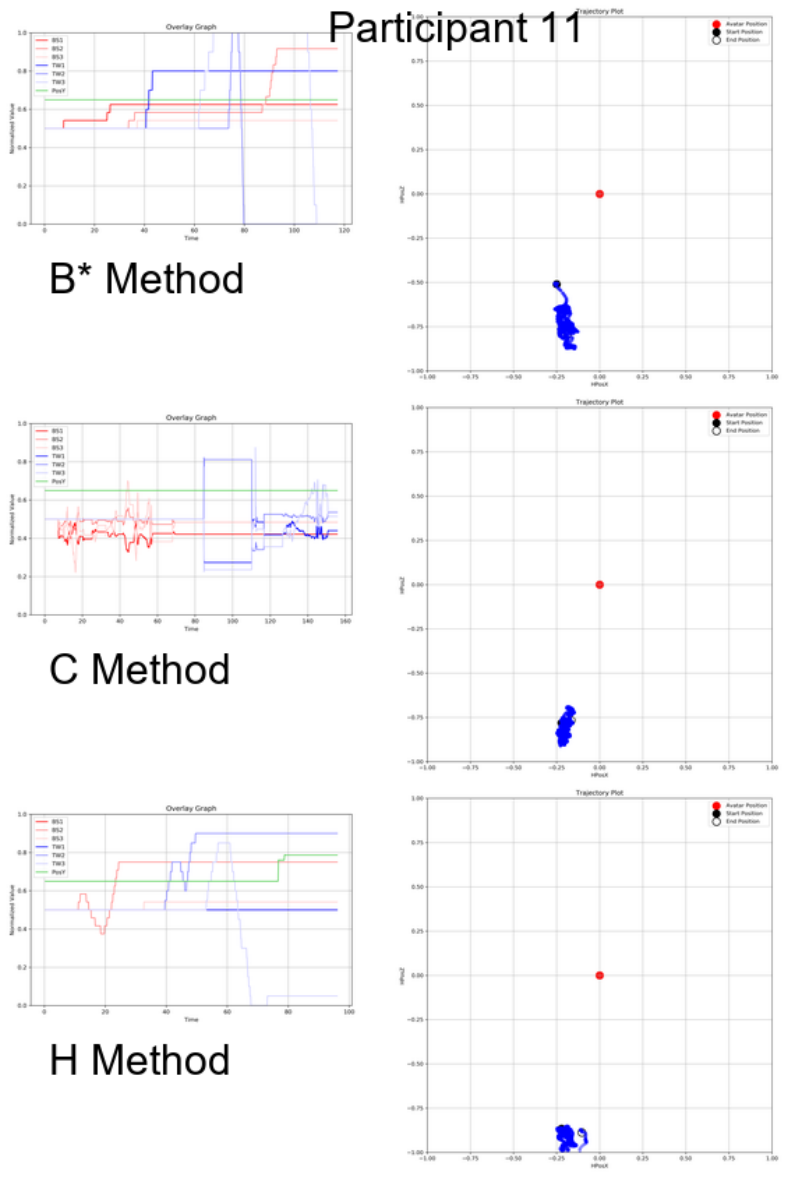


Figure A.11: Total results of participant 11 of the user study in Chapter 5

Participant 12

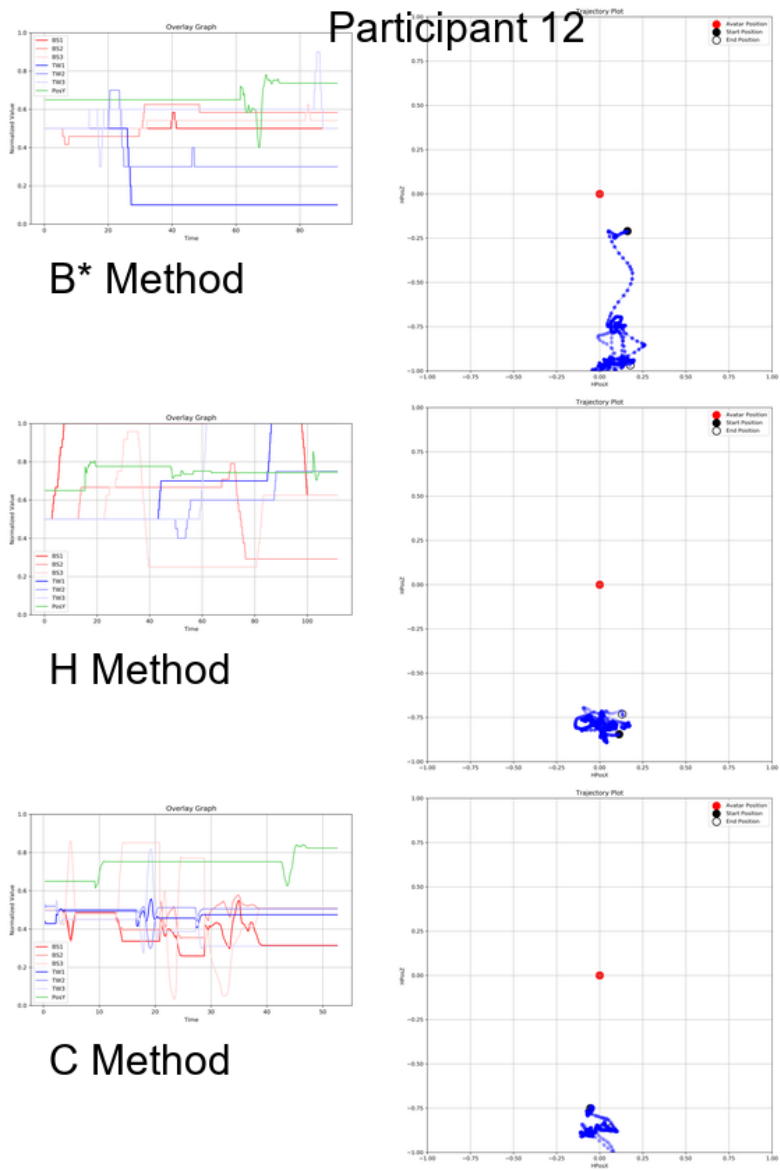


Figure A.12: Total results of participant 12 of the user study in Chapter 5