Augmented Reality User Interface

Aurora Cao

Faculty of Engineering Science, The University of Hong Kong

Laidlaw Undergraduate Scholarship Programme research report

Supervised by:

Dr. Anthony Steed

Department of Computer Science, University College London

Dr. Zhongrui Wang

Department of Electrical and Electronic Engineering, The University of Hong Kong

Acknowledgements

I would like to express sincere gratitude for the generous funding and precious opportunity provided by Irvine Laidlaw Foundation. The Laidlaw Undergraduate Scholarship Programme enabled me to have this intriguing and educational summer research experience. Moreover, I would also like to express my most heartfelt thanks to my host supervisor, Dr. Anthony Steed, for his help and guidance as well as my home supervisor, Dr. Zhongrui Wang, for his encouragement and support. Finally, I want to show my full appreciation to Ms. Audrey Chuang for being a truly helpful and responsible programme supervisor and Ms. Helen Lockey for the amazing leadership training sessions.

Content

1. Introduction

With growing popularity, virtual reality (VR), especially immersive VR, is now creating novel ways for a variety of social interactions including entertainment, education, training and so on [3]. Inside virtual platforms, avatars are a most essential part of the user interface since users are represented as avatars to better experience virtual interactions—avatars can be considered as direct projections of users themselves into the virtual environment. Therefore, there could be an indication that the representation forms of users—characteristics of avatars play an invaluable role in user experience and immersion of interactions. There have been previous projects targeted at observing how manipulating avatar features can contribute to the field of virtual multi-user interaction. For instance, L. Izzouzi and A. Steed have integrated the open-source Microsoft Avatar Library into Ubiq—an augmented reality (AR) platform supporting a massive number of interactions, in order to create two contrast avatar systems. Afterwards, several lab experiments for users to experience both the Microsoft Avatars—the more realistic one and Ubiq's original floating Avatar—robot-like, are launched with the purpose of collecting useful behavioral data about how people are influenced when represented differently during non-verbal social platform interactions [4]. Despite the successful result the above experiment has gained, there are some ways to improve the experiment by making the Microsoft Avatars more flexible and real. Subsequently, it's assumed that on top of this experiment, the inverse kinematics (IK) method being implemented onto hands and fingers can be one way to improve the realness and flexibility of the avatars [4]. Under the better contrasting conditions, it's hoped that people's preference for virtual images can be more precisely analyzed, dedicating to the development of AR user interfaces. This essay will present the background technologies, process of analysis, methodologies, and result achieved.

2. Related work and background technologies

2.1. The MoveBox open-source toolbox and motion capture

MoveBox is an open-source toolkit published by Microsoft for avatars to better animate the captured movements of users. It contained open-source code for detecting and imitating upper and lower body movements and facial movements. For avatars to achieve real-time imitation, the motions are tracked by a depth camera and processed through the codes [2]. For example, the codes contained inside IK hand-tracking file are able to detect and mimic users' hand movements like finger bending with low latency and high accuracy.

2.2. The inverse kinematics (IK) methodology

Kinematics is a method to control robots to perform certain movements. To determine whether robots have reached their target states, two more concepts need to be introduced—the joint coordinate and the Cartesian coordinate. Since statistics of the Cartesian coordinate are more direct to be measured, Cartesian vectors are widely used to determine the robots' position from the observers' views [5]. However, the joint coordinate can better present the inner forces and rotations of robots' body parts, which is certainly essential for robots to move and act in a human-like normal way. Conducted from the above two conditions, the transform of coordinates is crucial when controlling a robot, and matrix transform is its solution. On the contrary, the inverse kinematics (IK) got its name because instead of transforming the joint matrix into Cartesian matrix like the kinematics process [5], it does the opposite. With the procedure opposite, the functions of IK became the inverse as well. Instead of telling the robot to change their joint angles and positions to achieve certain positions, IK examine the positions first and then uses them to determine the angles and positions of joints to make the movement more natural. Therefore, IK can make great contributions to helping avatars imitate human movements.

2.3. Relating facts about Unity and Ubiq

In general, Unity is a popular cross-platform engine that provides many toolkits for developers to publish 2D/3D and AR/VR applications, while Ubiq is an AR virtual platform under Unity that links z single user into a broader network to achieve interactions among a great population. They both have high-level user interfaces to restore what's in reality and help users get immersed. For instance, Ubiq can enable users to communicate online just like they are meeting face-to-face in the real world. Same as in daily lives, Ubiq users can interact with objects in their own space or choose to share them and interact with other people via virtual common rooms created by themselves or others. Like objects in reality—except they are generated by C# code, Ubiq's procedure of creating an object is called 'defining', sharing and maintaining an object is called 'spawning', and both Unity and Ubiq call a group of closely related objects 'prefab'.

3. Analyzations and findings

On top of the previous works, this summer mainly tried to implement Microsoft's open-source code MoveBox into the existing Ubiq Avatar system. After doing related research on the internet, analyzing the current Microsoft Library and Ubiq's avatar functioning structure and procedure were of great importance. I found out that the avatar-spawning scripts and avatar-defining scripts were independent, and most importantly, the original avatar-defining scripts came as one whole prefab. This showed that the new Microsoft Avatar prefab can be directly inserted into the Ubiq system without making changes to the spawning process. What's more, stated that "different motion capture and avatar systems may represent the skeletal structure in different ways, with variations in the number of joints and the topology of bones" [1]. Therefore, to find out how to code for the new way of representation, I laid my next focus onto important code scripts' reading. I found out the original joints were all similarly coded, indicating that one could add new joints by simply duplicating and making minor changes in the existing code and this went the same with adding anchors. As a short summary, things were narrowed down into environmental setup and simple code changing.

4. Methodology

After the corresponding analysis, several steps were listed as the methodology to clarify the procedure.

4.1. Setting up the environment

For starters, it's important to ensure that the software versions match with the hardware. Virtual Reality experiences can be supported on several systems like the Meta Quest and HTC Vive and usually require headsets to collect visual information. Connecting the headset to computers needs the support of GPU-matching graphic cards and environmental toolkits. To be specific, the Quest2 headset was used during this summer's hand-IK development, which needs the matching version of the NVIDIA graphic card and corresponding CUDA toolkit. Afterwards, the overall environment needed to be set up for all three packages—the Microsoft avatar library, MoveBox toolkit and Ubiq to function normally. This required the correct version of Unity and MoveBox toolkit package.

4.2. Establishing communication between user and virtual avatars

After setting up the environment, the next step would be to establish connections between the headset and computer by implementing camera packages and manipulating the settings inside the Unity inspector slot to specifically suit the Quest2 headset and hand-tracking model. Then the IK hand-tracking package in MoveBox should be merged into the original Microsoft Avatar prefab, along with the successfully set camera, to create a new package object that has the capability of detecting and mimicking headset users' finger, hand and arms' movements.

4.3. Implementing new features into avatars

After previous preparations, efforts were then made to replace the Ubiq's original avatar with the new object package, thus fitting the newly improved Microsoft Avatars into the Ubiq network to enlarge the contrasting factors for further experiments and studies aimed at finding how differing people's virtual representations can affect their interactions including their moods, tones, communication frequencies and so on. Referring to the analysis in 3., after swapping the avatar defining prefab, the following step would be code adjusting, which required the beforehand analysis of differences between the upper parts of Ubiq's original floating avatars and newly implemented Microsoft avatars. Conclusions were that the Microsoft avatars have arms and movable fingers while Ubiq floating avatars don't, which foreshadowed that more parts of the human body would need tracing and imitating. According to the functioning procedure of the inverse kinematics (IK) methods, more joints and complex topology were required, leading to more variables to define and process in code as well as more anchored sensors to be created inside prefabs. Therefore, this first procedure is to add more functioning variables that stand for new joints into Ubiq character's C# coding scripts. Since joints for both hands are the same, and joints for fingers are repetitive, the hand can be seen as a class—a structure to group variables, which can clarify and simplify the code and fingers can be seen as repetitive members in one list with only serial number differences. After fixing up the codes inside the avatar's scripts, corresponding anchor sensors were added by inserting new game objects into the avatar prefabs. With the two above steps settled, the next step would be to adjust the angle parameters until the avatar's hands and arms look normal as in reality.

4.4. Demonstrating and recruiting participants

Finally, after successfully implementing the IK method into fingers and hands, a demonstration documentation or video would be made. Since the ultimate goal of the project is to study human behaviors, recruitment for active participants would be needed. As a consequence, hand-tracking IK along with other improvements by other peers in the group including implementing facial tracking and lower-body IK would be merged into one whole showcase in order to attract people's attention and interest.

5. Progress and result

This summer I mainly followed the steps listed in the above methodology part. I have managed to finish the first 3 steps except for the last part—fixing angle parameters. To ensure the work can be smoothly handed over, I uploaded my current codes to Git Hub as backup and made documentation on some of the key points and summarized the current process and what's next to be done. Since the ultimate goal of this project is to study non-verbal virtual social interaction and I ran out of time to make contributions inside the current project, I assisted with other projects at that stage to gain experience and make the most of my research time. For example, recording participants' reactions under different modes—using controllers or bare hands, to play a block stacking virtual game. All in all, I believe I have tried my best to learn and contribute.

6. Conclusion

During this six-week opportunity, I explored more hidden facts that I previously had not realized in the fields of virtual reality (VR) and augmented reality (AR). Contradicted to my immature understanding, VR and AR actually have more human and social-based characteristics. They tended to be more public serving and day-to-day than just abstract contents and narrow spectrum software. Immersive AR and VR are becoming increasingly popular nowadays with many similar platforms springing up. Under such circumstances, creating attractive user interfaces is one of the most crucial actions to raise a platform's competitiveness. This six-week opportunity tried to seek solutions in the aspect of upgrading software and launch experiments to conduct results from gained participants' feedback. Thus, to achieve the final conclusion, it needs to connect technology, sociology and ecology together in harmony. With the three fields stimulating and improving one another, the lives of people could be more convenient and greatly benefited.

References

[3] M. Melo, G. Gonçalves, P. Monteiro, H. Coelho, J. Vasconcelos-Raposo, and M. Bessa, ‘‘Do multisensory stimuli benefit the virtual reality experience? A systematic review,’’ *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 2, pp. 1428–1442, Feb. 2022.

[2] M. Gonzalez-Franco et al., "MoveBox: Democratizing MoCap for the Microsoft Rocketbox Avatar Library," *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR),* Utrecht, Netherlands, 2020, pp. 91-98, doi: 10.1109/AIVR50618.2020.00026.

[4] L. Izzouzi and A. Steed, "Integrating Rocketbox Avatars with the Ubiq Social VR platform," *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, Christchurch, New Zealand, 2022, pp. 69-70, doi: 10.1109/VRW55335.2022.00025.

[1] B. Spanlang, X. Navarro, J.-M. Normand, S. Kishore, R. Pizarro, and M. Slater, “Real time whole body motion mapping for avatars and robots,” *in Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, 2013, pp. 175–178.

[5] Poon, J. K.-S., & Lawrence, P. D. (1988). *Multiprocessor - compatible inverse kinematics and path planning for robots.* Thesis (Ph.D.) --The University of British Columbia (Canada), 1988.