First Person Movement Control with Palm Normal and Hand Gesture Interaction in Virtual Reality

Chaowanan Khundam
Multimedia Technology and Animation
School of Informatics, Walailak University
Nakhon Si Thammarat, Thailand
chaowanan.kh@wu.ac.th

***Abstract*—Virtual reality (VR) has become a very popular technology in recent years, which used in the field of multimedia for various purposes. One of the applications that widely used for simulating physical presence in the real world is walk-through VR. In the walk-through VR system will generate simulation character or avatar for user which able to control movement, especially first person movement walk-through in VR with many input devices. In this paper, we present a human - computer interaction with the connection of Oculus Rift and Leap Motion new technological devices for VR. Oculus Rift is VR headset or head - mounted display devices that have a small display optic in front of each eye. Oculus Rift can track head movement and change view point follow it. Leap Motion is in - air controller that can track hand gesture of the user. The combination of them will make users feel like immerse to VR. Users can move avatar any way in VR by their hand interact through the system via these devices. We introduce a new interactive hand gesture system with palm normal for control steering develop by the game engine Unity3D applies synchronization of Oculus Rift and Leap Motion. Our design and development method will allow users to adjust moving speed follows the hand gesture and the range of the user's hand that make a smoothly moving with acceleration.**

***Keywords*—Virtual Reality; Human – Computer Interaction; Head – Mounted Display; In – Air Controller; Game Engine**

I. Introduction

Human – computer interaction (HCI) is an interdisciplinary field in which computer scientists, engineers and design professionals play important roles. Virtual reality (VR) is a computer – simulated environment that can simulate the physical presence in places in the real world or imagined worlds. The goal of VR system is to immerse the participant or others within a computer – generated interact with the virtual environment. Human interaction with VR is making computer – based systems easier to use and more effective for people focus on control movement in VR. Therefore, HCI in VR consists of three parts such as the user, the computer and the way to work together. In this research user mean an individual player in VR while computer means all devices such as Oculus Rift, Leap Motion and computer PC. Interaction is the last part and will be the main part of this research. We studied HCI for design and development depend on interaction between all devices with user [1] [2]. Traditional way to control movement is discrete control by keyboard which using buttons. Users will press and release the key for movement in VR this method appropriate for PC's monitor display. If the system uses others device like HMD its will uncomfortable for controlling movement.

Interaction in VR is very challenging for developers to create new method or applications with respond to the needs of users. Especially, continuous movement control by using hand gesture that able to track hand ranges which combination with various devices. Interaction devices in VR for movement control base on hand gesture in this paper categorized into 2 types. The first one is HMD devices these are tracking and signaling actions are the primary means of input into virtual environments. The HMD device that we use for this project is Oculus Rift DK2. The other one is hand tracking devices or in - controller for track user's hand. We use Leap Motion for tracking and determination of a hand's position and orientation. Then this device track and report the position and orientation of the game engine. The game engine is provided for connecting between users and all devices. By the properties of the game engine Unity3D appropriate for using with already supported with Oculus Rift and Leap Motion [3][4].

Oculus Rift will provide users for first person perspective in VR by stereoscopic display. VR with a first-person perspective are usually avatar-based, wherein the displays what the user's avatar would see with the avatar's own eyes. Thus, users typically cannot see the avatar's body, though they may be able to see the avatar's weapons or hands [5]. Traditional control for first person in VR by use keyboard and mouse not fully support with HMD. For a more immersive VR, users should interact in virtual by their body, such as gesture by hand. VR will feel instantaneous as smooth and seamless as the real world when users are able to control movement of them. This reason motivated us to develop a new method for control user's movement in VR with hand gestures in real-time.

The objective of this research will focus on a combination of Oculus Rift (HMD device) and Leap motion (in – air controller) for first person movement control in VR. We design hand gesture synchronize with movement and develop a method of control in our experiment covering simple path, turning path and complexity path. The result confirms that

Fig. 1. Oculus Rift DK2 and Leap Motion (Leap Motion for virtual reality)

Fig. 2. Leap Motion and the field of view with wide angle beyond Oculus Rift.

users are comfortable to use this input method for movement control in VR. In section II we describe about related technologies and related works. Section III explains the design of hand gesture for first person movement control with acceleration. We show our implementation in section IV and experimentation in section V. Finally, we summarize our research and discuss for future works in section VI.

II. RELATED TECHNOLOGIES AND WORKS

A. Related Technologies

There are 2 devices Oculus Rift and Leap Motion in this research. Furthermore, we use the technology of game engine Unity3D in developing this project to control first person movement in VR. All details of these technologies and how to work it together would be described in this part.

The Oculus Rift is a head-mounted virtual-reality display developed by Oculus VR company [6]. An immersive headset that gives the wearer a full 360-degree view of the virtual world they inhabit. The Oculus Rift pairs with headphones to make games, virtual worlds and live events feel 'real'. Oculus has produced a software development kit (SDK) to assist developers with integrating the Oculus Rift for their applications.

The consumer-oriented version of the Oculus Rift is in development and will feature a greater than 1080p low-persistence-of-vision OLED display. Oculus Rift DK2 currently has a 5.7 inch display with 24-bit color depth, generating a 1080p display in each eye. As now those displays will use stereoscopic 3D to mimic normal human vision. This means that the left eye sees extra space to the left and the right eye sees more to the right. This gives Oculus Rift a greater field of view, echoing that offered by previous VR headsets.

The headset also has a motion-tracking system that uses an external camera which tracks infrared dots located on the headset. The latest version includes Oculus' new 1000 Hz Adjacent Reality Tracker that will allow for much lower latency tracking than almost any other tracker. It uses a combination of 3-axis gyros, accelerometers, and magnetometers, which make it capable of absolute (relative to earth) head orientation tracking without drift. The VR headset is termed as the next big thing in the world of gaming. It has been designed for completely immersing people in gaming with stereoscopic 3D screen and makes them feel that they are actually in the game. Although Oculus will design for game playing, but it is very useful for VR with the reason of many games created based on virtual environment by game engine.

Leap motion is the motion-sensing technology for human interaction that supports the hand and finger motion as input, analogous to a mouse, but requiring no hand contact or touching [7]. The device is a little sticky gum sized package that sits in front of you and hooks to a PC via USB 2 or USB 3 which using two monochromatic IR cameras and three infrared LEDs observes a roughly hemispherical area to distance about 1 meter. The LEDs generate a 3D pattern of IR dots light and the cameras generate almost 300 frames per second of reflected data, which is then sent through a USB cable to the host computer and synthesizing 3D position data by comparing the 2D frames generated by the two cameras. That's make the user keep up with every move. The built in LED illuminators and sensors have a 150 degree field of view in 3D space and detect your 10 fingers in millimeter accuracy. There is a fraction of a second lag that is imperceptible. It's wide 150 degrees field of view and Z-axis of depth. Users can move their hand in 3D like the real world.

Unity3D is a game development and cross-platform game creation system developed by Unity Technologies with a powerful rendering engine fully integrated with a complete set of intuitive tools and rapid workflows to create interactive 3D or 2D content easy to publish to multi-platform [8][9]. Unity integrates seamlessly with Leap motion control designed a variety of assets for Unity Free and Pro, including rigged hands, demo scenes, and resources for VR. Furthermore, Unity also supports to Oculus Rift it is available for all of developers to create applications for Oculus Rift and give Oculus integration package asset to deploy any sort of VR content imaginable to the Rift. The Oculus add-on will include stereo imaging optimizations, 3D audio support, deeper Unity editor integration, inclusion of the Oculus rift in the Unity development and debugging workflow, integration of Oculus-specific APIs within Unity, and direct publishing to the Oculus platform.

Leap motion for VR [10][11] is a combination of Leap motion and Oculus rift it is 3D output meets 3D input between their devices by setting up Leap motion in front of Oculus rift. This is the next level of VR with true 3D interface made users' vision can see their hands and interact by hand gesture in VR. Leap motion is a good controller for hand, but there are less of applications that develop for Leap motion. However, after Leap motion cooperated with Unity3D developers will have increased and there is more applications including ability to work with other device through Unity3D like Oculus Rift. The future of Leap motion is interesting to create more ability of application that up to developers for this challenge as well.

B. Related Works

There are many features of VR controller input devices for different approach some research created a system that assembles various devices for a purpose. Oculus Rift and Leap Motion have recently been used for many researches especially in VR. Interaction with VR is very interesting for developers to study about user's behavior and understand what the efficient application for them in the future.

Previous research about in - air controller and hand tracking is aim to create and used applications for VR. Accuracy of hand tracking is the importance for stable system Yoichi et al. [12] introduce a method for tracking a user's hand in 3D and recognizing the hand gesture in real-time without the use of any invasive devices attached to the hand. Input method uses multiple cameras for determining the position and orientation of a user's hand moving freely in a 3D space identifies predetermined gestures in a fast and robust manner by using a neural network which has been properly trained beforehand. There are several types of applications, including 3D object handling for a desktop system and 3D walk-through for a large immersive display system. Many research design hand gesture and applies for its applications. Chastine et al. [13] present a study on using single hand gestures to control a first person shooting game compared with traditional control by keyboard, mouse and gamepad. The goal of this research is to help inform the design space for game researchers, designers and developers for integrating gesture control into their games. The result has shown that in FPS games user training and practice are critical to the performance of gestural-based game control. Participants became more adept at using the device as they progressed through the games indicates that it is indeed possible to use gesture-based control with no experience players. This feedback supports to use Leap motion as a controller in VR and ensure that there is sufficient reason for developers to create new applications with it in the future.

There are many using of HMD in VR for experiencing with 3D models. Stefan Greuter and David J. Robert [14] introduce a system namely SpaceWalk. This platform is split into two hardware components including a motion tracking station and a wireless VR backpack allow a low encumbrance full-body immersion in VR while walking around the living-room. The majority devices of this system are combined from Oculus Rift (DK1) to be an HMD and use Microsoft Kinect 2 for capturing the movement body and limbs processing by a backpack tablet computer runs commodity VR software (Unity3D) with their extension code that integrate all components together. In this research allows users to walk and interact with objects in virtual space in the living-room, but this platform is not provided for very large scale VR environment more than this room. Webel et al. [15] introduce creating a low-cost, fully immersive and non-stationary virtual reality setup that allows the user to intuitively experience cultural heritage artifacts. They explore and analyze recent devices such as the Oculus Rift HMD, the Microsoft Kinect and the Leap Motion controller. When creating HMD VR scenarios, especially recent devices like the Oculus Rift HMD, the Microsoft Kinect and the Leap Motion deliver a very good quality of output. The combination with natural interaction inputs with the Kinect or the Leap Motion enables the user to directly interact with the virtual world. However, this VR system is usually still restricted to the relatively small space in front of the sensor because of user's movement control. Therefore, using of interactive hand gesture for movement in VR will improve the VR system for the user's control by strict position and use hand gesture replace traditional methods.

Increasingly in the range of input devices that make people can use to interact with computers and applications. However, there are no established interaction standards or best practices when developing software for emerging technologies, and user experience suffers as a result. Jake Araullo and Leigh Ellen Potter [16] present a case study exploring the experiences of a group of individuals when playing games using Oculus Rift and the Leap Motion controller. This research has shown that participant responses and actions during sessions also indicated that application use was negatively affected by the inclusion of mixed traditional and non-traditional input peripherals, and by relying on traditional interface metaphors when using emerging technologies.

III. Hand Gesture Design for Movement Control

The main objective of this research is developing a new input method in VR. Therefore, implementation of using Oculus Rift and Leap Motion are considered to apply for first person movement control in VR such as walk-through VR that allow user to immerse with the virtual environment by moving and observe them. In this section, we design single hand gestures by continuous motion of the hand during the walk-through in VR separated as follows:

A. Forward Movement

For hand gesture if user wants to move the avatar forward in our method just raise a user's hand in front of their face with palm direction straight ahead. Try to open hand for better detection and control palm normal always forward.

B. Backward Movement

To move avatar backward (user still looking straight ahead not turn back) our method will use the hand gesture appearance to turn the direction of the palm or flipping the hand facing the body. In this case palm normal will point to the back of the user.

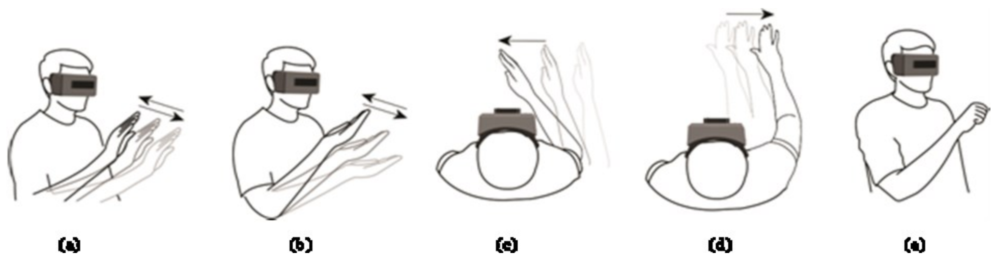

Fig. 3. All hand gesture for movement control (a) gesture for forward movement (b) gesture for backward movement (c) gesture for left step (d) gesture for right step (e) gesture for hold position

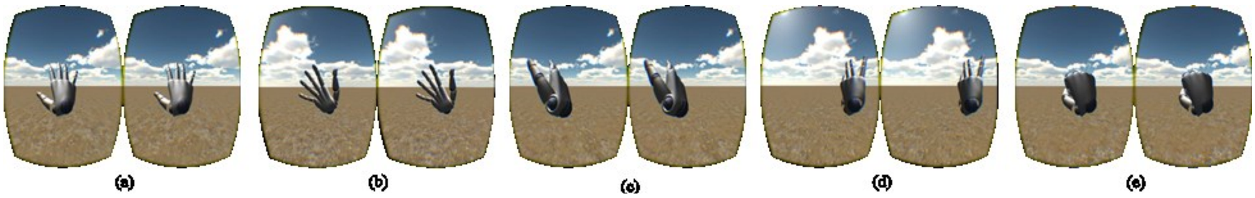

Fig. 4. User's perspective view from first person controller by difference hand gesture (a) when user move forward (b) when user move backward (c) when user step left (d) when user step right (e) when user stop movement

C. Step Left and Right

Side step in this case does not turn left or turn right, but it means walk step to left side or right side while character's face continue looking forward. To move a character step left by hand gesture we will push the palm to the left hand side. As same as step left, step right will push the palm to right hand side by rotating the wrist down out of the body.

D. Stop (Hold Position)

To hold position of the character during moving with hand gesture there are 2 methods to do. The first way is grasping the hand another way is taking the hand out of display area. If no hand enables, character will stop immediately and move again when take the hand into the display.

E. Speed Control

We separated speed controller into 3 cases such as forward speed, backward speed and step side speed. For forward speed, users are able to increase movement speed by pushing the palm more forward. If user pushes less it will affect a little speed up, but if the user pushes more it will increase the more speed too. In case of backward speed, user can adjust moving backward speed by push and pull their hand. However, it has conversed way if user would like to speed up backward they have to pull their hand into their body. For step left and right speed is up to user pushing their hand to side of the screen more or less.

IV. 3D Hand Localization and Implementation

The estimation of 3D hand position in VR will support and provided by Leap Motion's SDK in Unity3D. For more command and apply the SDK function, we can edit the script directly. In the Fig. 5, we set the position of user and user's hand follow the position in VR. Assume that position of the user's avatar denoted by $A(x_A, y_A, z_A)$ and the user's hand denoted by $H(x_H, y_H, z_H)$, which have $\overrightarrow{N}$ as palm normal. We use hand position for detecting hand as follows:

$$isHandEnable = \begin{cases} true & ,H(x_H, y_H, z_H) \neq null \\ false & ,H(x_H, y_H, z_H) = null \end{cases}$$

Before tracking hand for control avatar we always detect hand first after that we will check hand gesture. If *isHandEnable = false* user's avatar will not move and standby else if *isHandEnable = true*. When we detect hand position and palm normal we also have an angle of 2 vectors for check gesture. In this movement we considered only horizontal axis consist of x-axis and z-axis. Thus, the angles we use for movement control are the angle of the hand and the angle of $\overrightarrow{AN}$. We define these angles by horizontal value as follows:

$$\theta_H = tan^{-1}\frac{|\overrightarrow{N}_z|}{|\overrightarrow{N}_x|}, \ \theta_A = tan^{-1}\frac{|\overrightarrow{AN}_z|}{|\overrightarrow{AN}_x|}$$

These angles is use of check moving forward, backward, move left or move right. When we detect the user's hand, palm normal are also detected in local coordination relative with hand position. The direction of this vector will influence to player movement. We can check player move forward by the direction of this vector by θ_H as follows:

$$isForward = \begin{cases} true & ,0 \leq \theta_H < \pi \\ false & ,\pi \leq \theta_H < 2\pi \end{cases}$$

If θ_H value is moreover 0 to π, it mean that users are pushing the hand ahead and avatar will move forward.

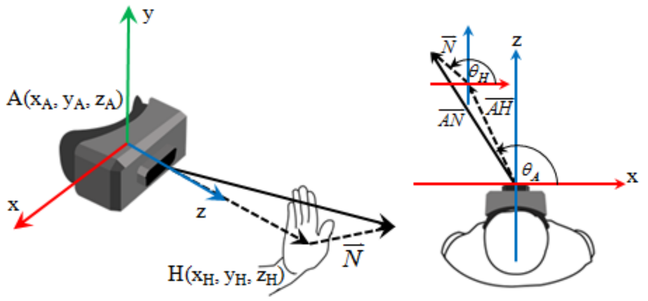

Fig. 5. 3D hand localization and palm normal related with user's avatar

Otherwise, if θ_H value is moreover π to 2π users are flipping their hand and avatar will move backward. Moreover, if θ_A value is in the quadrant 1 or 4 this mean user move the hand to the right and avatar will move a side step to right hand side. Conversely, the user will move the hand to the left and avatar will move a side step to left hand side as follows:

$$isRight = \begin{cases} true & ,0 \le \theta_A < \dfrac{\pi}{2} or \dfrac{3\pi}{2} \le \theta_A < 2\pi \\ false & ,\dfrac{\pi}{2} \le \theta_A < \dfrac{3\pi}{2} \end{cases}$$

Assume that the distance of push and pull their hand that we considered respect to Z axis is front range or distance of Z between hand and the user. Another distance is the lateral range we consider by X axis of the hand relative to user's front view line or Z axis. Front range and lateral range value are defined as follow:

$$front = \left|\overline{AN}\right| sin\,\theta_A = \left|\overline{AN}_z\right|$$

$$lateral = \left|\overline{AN}\right| cos\,\theta_A = \left|\overline{AN}_x\right|$$

These distances are calculated with movement speed. The *front* value will calculate with speed of move forward and backward, while *lateral* value will calculate with speed of move left and right.

We have a function for fist detecting namely *pinchStrench*(). The return value of *pinchStrench*() function is always 0 to 1. If we detect that *pinchStrench*() value more than 0.5 we claim that user are fisting as follows:

$$isFist = \begin{cases} true & , pinchStrench() \ge 0.5 \\ false & , pinchStrecnh() < 0.5 \end{cases}$$

When user shows fisting the avatar will stop moving forward or backward, but still able to move sidestep left or right. If users want to stop they have to hold their fist in the middle of the screen. We will see that when the user move forward, move backward and stop moving are still allow moving left or right at the same time. While moving forward, move backward and stop moving command are separated, unable to work together. These imply to the logical command relate to all parameters as follows:

$$moveForward \equiv isHandEnable \wedge isForward \wedge \neg isFist$$

$$moveBackward \equiv isHandEnable \wedge \neg isForward \wedge \neg isFist$$

$$moveLeft \equiv isHandEnable \wedge \neg isRight$$

$$moveRight \equiv isHandEnable \wedge isRight$$

V. Experimental Evaluation

To test the overall performance of our movement control method by using gesture controlling. We compare movement control performance in Oculus Rift by using a gamepad or joystick and using Leap Motion by our gesture method. The variable that we are considering in this case is the time used during control the avatar from start to finish. We use 11 participants in this experiment most of all have some experience about first person shooting in computer games with keyboard and mouse control, whereas no experience of Oculus Rift and Leap Motion. Before testing, we will explain all missions and show how to control with gamepad and how to control movement by hand gesture. Furthermore, we will let user wear the Oculus Rift and test to rotate their heads with arbitrary angle for familiar with HMD device.

There are 4 missions in our experiment. All missions will use Oculus Rift to be an HMD device and use difference input device to control movement with difference scenes. The first mission, participants will control movement with gamepad in Scene1. The reason that we test participants with gamepad first because of all user have an experience for control with this device they are just known how to complete mission. It will be good to start with this mission. The second mission still using gamepad, but change the scene to the Scene2. After that, the third mission, we will come back to use the Scene1 but change input to Leap Motion and control movement with a hand gesture. Finally, the forth mission, we use the Scene2 with Leap Motion and hand gesture also.

The result after all participants testing has been shown in the Table I. The average time in seconds by using hand gesture control of Leap Motion is less than using of gamepad in both of the scene. Even though there are unstable of Leap Motion detecting hand in any case made fault control of the device not users. To solve this problem we have to update firmware and clear data always before using. However, we will see that gamepad have less deviation (7.95 and 5.78) than hand gesture by Leap Motion (9.54 and 10.41) because of all users used to play some first person shooting game with a keyboard whereas no experience with hand control that made the value like this.

TABLE I. Evaluation of Time (Second) Used in Experiment

	Scene1		Scene2	
	Gamepad	**Gesture**	**Gamepad**	**Gesture**
Min	51.10	37.11	113.14	87.75
Max	76.39	70.54	131.06	118.65
Average	57.31	47.21	120.49	100.58
STDEV	7.95	9.54	5.78	10.41

VI. CONCLUSION

Hand gesture interaction for first person movement in VR is useful for users to control the movement of their hand. In this research, we apply palm normal with hand gestures and implement for single hand control. Hand controllers with palm normal have more efficient and stable than hand gesture without palm normal. Another advantage of this method is controlling the speed of avatar movement. In particular, by the value of *front* and *lateral* are continuous values that able to control the speed like accelerator meter up to the user requirement. Improvement of HCI, hardware and software of interaction technologies will help VR system designers develop more natural and effective for participants to interact with the VR. The future work in this research, we would like to use both on hand for more control movement of a user's avatar including access of many functions or menu in VR systems. Furthermore, we are looking for add more devices such as Virtuix Omni, Sixense, Depth VR, etc. in our system and create new scripts contact with each device for work together.

ACKNOWLEDGMENT

This research was supported by funding of young researcher scholarship from Walailak University contact No.WU58302. The authors would like to thank our staffs from Multimedia Technology and Animation Lab, School of Informatics for all figure descriptions.

REFERENCES

[1] G. R. Torre, "The Importance of the Sense of Touch in Virtual and Real Environments". IEEE Multimedia 13(3), Special issue on Haptic User Interfaces for Multimedia Systems, 2006, pp. 24–30.

[2] Sutcliffe, A. G., and K. Deol Kaur. "Evaluating the usability of virtual reality user interfaces." Behaviour & Information Technology 19.6 (2000): 415-426.

[3] Rekimoto, Jun. "Gesturewrist and gesturepad: Unobtrusive wearable interaction devices." Wearable Computers, 2001. Proceedings. Fifth International Symposium on. IEEE, 2001.

[4] Patel, Harshada, et al. "Human centred design of 3-D interaction devices to control virtual environments." International Journal of Human-Computer Studies64.3 (2006): 207-220.

[5] Lubos, Paul, Gerd Bruder, and Frank Steinicke. "Analysis of direct selection in head-mounted display environments." 3D User Interfaces (3DUI), 2014 IEEE Symposium on. IEEE, 2014.

[6] D. Plemmons and D. Holz, "Creating next-gen 3D interactive apps with motion control and Unity3D." ACM SIGGRAPH 2014 Studio. ACM, 2014.

[7] A. Rollings and E. Adams, "Fundamentals of game design." New Challenges for Character-Based AI for Games. Chapter 20: Artificial Life and Puzzle Games. Prentice Hall (2006): 573-590.

[8] P. Lubos, G. Bruder and F. Steinicke, "Analysis of direct selection in head-mounted display environments." 3D User Interfaces (3DUI), 2014 IEEE Symposium on. IEEE, 2014.

[9] L. Garber, "Gestural technology: Moving interfaces in a new direction [technology news]." Computer 46.10 (2013): 22-25.

[10] S. Blackman, "Beginning 3D Game Development with Unity: All-in-one, multi-platform game development." Apress, 2011.

[11] I. Goradia, J. Doshi, and L. Kurup, "A Review Paper on Oculus Rift & Project Morpheus." (2014).

[12] Sato, Yoichi, Makiko Saito, and Hideki Koike. "Real-time input of 3D pose and gestures of a user's hand and its applications for HCI." Virtual Reality, 2001. Proceedings. IEEE., 2001.

[13] J. Chastine, N. Kosoris and J. Skelton. "A study of gesture-based first person control." Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games (CGAMES), 2013 18th International Conference on. IEEE, 2013.

[14] Stefan. Greuter, and D. J. Roberts. "SpaceWalk: Movement and Interaction in Virtual Space with Commodity Hardware." Proceedings of the 2014 Conference on Interactive Entertainment. ACM, 2014.

[15] S. Webel, U. Bockholt and J. Keil, "Design criteria for AR-based training of maintenance and assembly tasks." In Proceedings of the 2011 international conference on Virtual and mixed reality: new trends - Volume Part I, Randall Shumaker (Ed.), Vol. Part I. Springer-Verlag, Berlin, Heidelberg, 2011, pp. 123-132.

[16] J. Araullo and L. E. Potter. "Experiences using emerging technology". In Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: the Future of Design (OzCHI '14). ACM, New York, NY, USA, 2014.