Intelligent Automation And Soft Computing, 2020
Copyright © 2020, TSI® Press
Vol. 26, no. 1, 99–105
https://doi.org/10.31209/2019.100000131

# Implementation of Local Area VR Environment using Mobile HMD and Multiple Kinects

**Soo-Kyun Kim[1], Chang-Hee Lee[2], Sun-Jeong Kim[2], Chang-Geun Song[2], Jung Lee[2]**

[1]Department of Computer Engineering, Jeju National University, 102 Jejudaehak-ro, Jeju-si, 63243, South Korea
[2]Hallym University, 1, Hallymdaehak-gil, Chuncheon-si, Gangwon-do, 24252, South Korea

**ABSTRACT**
Recently, the development of HMDs such as Oculus Rift, HTC Vive, and PSVR has led to an increase in the interest of people in virtual reality (VR), and many related studies have been published. This leads to an additional cost increase in configuring the VR system. Also, space problems are caused. When the treadmill is installed, additional space is required, which may adversely affect the popularization of VR. In this paper, we propose a local area VR environment that solves cost and space problems using human tracking using several Kinect and solves the hygiene problems using smartphone-based mobile HMD.

**KEY WORDS**: Virtual Reality, Mobile HMD, Multi-Kinect

## 1 INTRODUCTION

MANY researches on VR have been under way recently, such as IBM's development of the VR attraction system based on the novel 'Sword Art Online' (see Figure 1). However, the problem with these VR attraction studies is that they require a variety of additional systems and equipment such as treadmills, motion-recognition cameras as well as HMDs. This leads to an additional cost increase in configuring the VR system.

**Figure 1.** **'Sword Art Online VR', IBM.**

Also, as shown in Figure 1, when a treadmill is installed, a large space is required, which may adversely affect the popularization of VR. In particular, it is pointed out that hygiene problem such as pollution and viruses that occur when an unspecified number of HMDs are shared with each other is accompanied with inefficiency problem that is relatively small in the number of people to be accommodated.

In this paper, we propose a local area based VR environment that resolves cost and spatial problems by simultaneously tracking several users in a virtual space using several Kinects, and solves the hygiene problem using a smartphone based mobile HMD .

## 2 RELATED WORK

### 2.1 Motion Capture Systems and HMDs

**Figure 2.** **OptiTrack's VR system.**

CONTACT Jung Lee airjung@gmail.com

OPTITRACK uses motion capture system and HMD simultaneously to reproduce user's movement in virtual space (see Figure 2). By tracking the markers attached to the HMD using an infrared camera, the virtual camera can be rotated and moved in a virtual environment. However, this system requires all equipment to be wired, the marker must be attached to the HMD, and an expensive infrared camera is required.

Meanwhile, VicoVR, a crowdfunding project, provides a solution that enables motion capture cameras and mobile HMDs to be connected via Bluetooth to create a VR environment (see Figure 3). The system presented a new VR environment that is out of the existing marker-based VR environment. However, only two people can be traced at a maximum, and all the calculations must be handled by the mobile device because the data is transmitted directly to the mobile using Bluetooth communication without using a PC.


**Figure 3.** Wireless motion capture camera VicoVR.

### 2.2 *Human Body Tracking with Kinect*

Various studies using human tracking using Kinect, a motion recognition controller of Microsoft, have been conducted. Hariharan et al. (2014) built a virtual classroom using Kinect to interact with faculty and students in the E-learning class. This system allows the professor to give equal attention to the student because Kinect recognizes the motion and notifies the professor when the student hands.

Jin et al. (2015), Kumar et al. (2017), and Rane et al. (2018) proposed a personalized virtual trainer using Kinect that provides real-time guidance and evaluation while the user is exercising. This system helps the user to track and check his joint movements and guides him to correct his behavior by displaying a guide to the accuracy of the specific movement.

Heickal et al. (2013) implemented a universal human body motion capture system using human motion data.

Wilson et al. (2014) improved a “Walk in Place” (WIP) technique using two Kinect sensors and compared their method with joystick navigations.

### 2.3 *Studies with Multiple Kinects*

Originally, Kinect was not developed for motion capture, so its motion recognition accuracy is not as high as that of specialized motion capture systems such as Vicon. Therefore, there have been several attempts to improve its accuracy or extend its recognition range using several Kinects.

Kim et al. (2015) have conducted research to track user movements in the real world by arranging multiple Kinects and constructing a new coordinate system based on their installation locations.

## 3 SYSTEM DESIGN AND IMPLEMENTATION

### 3.1 *System Overview*

IN this paper, we propose a virtual reality system in which the real world and the virtual world correspond in 1: 1 in the local region by tracking user movements in the real world using several Kinects. Figure 4 shows a conceptual image of real-world to virtual reality by tracking each user's skeleton in the local domain using multiple Kinects and reflecting it to the virtual reality. In this study, each user's own smartphone was made into a mobile VR HMD using 'Google Card Board ', ' Gear VR ', etc. With such mobile VR HMDs, there is some degradation in graphics performance. But unlike desktop-based HMDs like the Oculus Rift and HTC Vive, you can experience virtual reality in a wireless environment, which improves your freedom in a virtual environment.

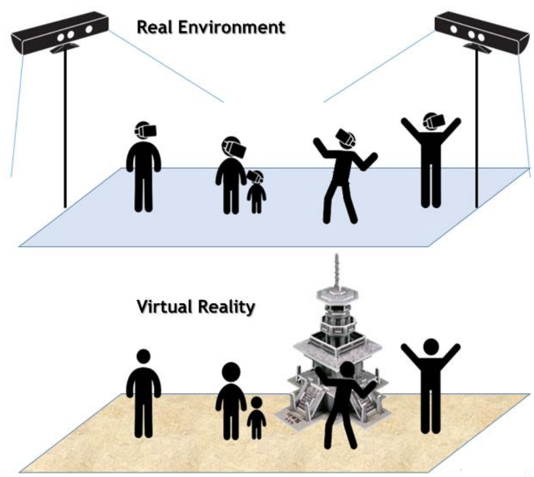



**Figure 4.** Conceptual image of local area virtual reality system using several Kinects.

As mentioned earlier, this study uses Microsoft's Kinect V2 as a motion tracking sensor and it can track up to 6 skeletons simultaneously. Since the interface of Kinect V2 is USB 3.0, only one Kinect V2 connection is allowed per PC. Therefore, to construct multi-Kinect environment, you need as many PCs as the number of Kinect to use. In this study, Kinect-PC set was constructed by using mini PC (ZBOX-CI323NANO, ZOTAC), since motion tracking with Kinect itself does not require a lot of computation and only USB 3.0 port is required. In addition, Kinect V2 works only on Windows 8 or later operating systems, so we installed Windows 10 on every PC that connected to Kinect.

### *3.2 Kinect Client*

This system is implemented in C # to configure multi-Kinect environments and is a client-server network environment consisting of client PCs with Kinect centering on one server PC.

The client constructs a packet of the skeleton data transmitted from the connected Kinect and transmits it to the server. Kinect provides two pieces of data about the skeleton: the center position of the skeleton and the rotation angle of each joint, quaternion. In this study, the position and posture of each user are determined by using the quaternion of each joint and the center position of the skeleton. The size of the position data is 12bytes (4bytes * 3 (x, y, z)) and the size of the quaternion data is 16bytes (4bytes * 4 (w, x, y, z)). Since the number of joints per skeleton is 25 (see Figure 5), the amount of skeleton data per person is 412 bytes (= 12 + 16 * 25).

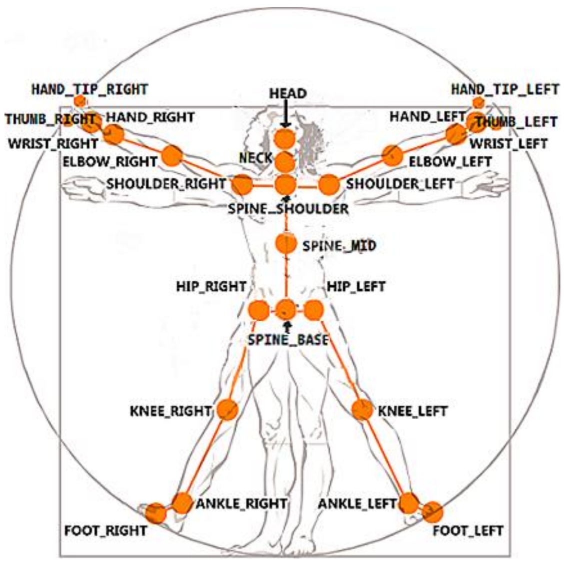



**Figure 5. Joint information provided by Kinect V2.**

The position and quaternion data of each joint is provided in the form of an array. Position data is provided for all joints, but quaternion data is not provided for all joints such as SpineBase, Head, FootLeft, FootRight, SpineShoulder, HandTipLef, ThumbLeft, HandTipRight, and ThumbRight. By excluding joints that do not provide quaternion data from the packet, the throughput can be optimized. In this study, however, position data was stored instead of the quaternion values of joints without rotation data. Therefore, it is used to solve the accumulated error problem due to the floating-point error that occurs when calculating the joint position using only the quaternion value.

### *3.3 Kinect Server*

In the Kinect server, the camera matrix of the corresponding Kinect client can be calculated using the position of the skeleton and the quaternion data of its joints transmitted from each Kinect client. Based on this, the position and movement of the user avatar can be reproduced in the world coordinate system of the virtual environment. The data is then delivered to each user to view other avatars in the virtual world, manipulate their avatars, and interact.

First, set the Kinect client first connected to the Kinect server to the origin (0, 0, 0) of the world coordinate system. All the Kinect are located at the same height, and the newly connected Kinect is installed at a position within at least one viewing angle of the existing Kinects. Through this process, you can calculate the relative positions of the other Kinects around the first Kinect. After each Kinect position calculation is completed, it is possible to correct the skeleton positions of the users recognized by each Kinect.

The position of each Kinect and the user's skeleton can be obtained in this way, but there is a problem in determining the frontal direction when the user's skeleton is recognized for the first time. This is an intrinsic problem with Kinect, when the skeleton is first recognized, it cannot be judged whether it is facing the front (see Figure 6).

**Figure 6. A case when the person who sees the back is recognized as facing the front.**

Most previous studies have calculated the rotation value by considering the first direction recognized by Kinect as the front [5][6]. In addition, most of the existing methods to solve this problem have recognized the orientation of the detected Kinect V2 as the front using the facial recognition detection function supported by Kinect V2.

**Figure 7.** **The frontal recognition result obtained by using the rotation value of HMD.**

In this study, however, it is difficult to detect the face because HMD is worn by the user. Therefore, we used the gyro sensor rotation value of the smartphone to detect the frontal direction. By calculating the inner product of the Z-axis vector of the Kinect and the Z-axis vector of the HMD, if this value is negative, it can be judged that the Kinect and the HMD are facing each other (see Figure 7).

And Kinect server finds the average of joint data from each Kinect client for all user skeletons and places it in virtual environment. This data is finally delivered to the smartphone HMD client. The MTU of a network is typically 1500 bytes, and generally the size available for applications is 1460 bytes, except for IP and TCP header data. As mentioned above, since the skeleton data per person is 412 bytes, in this study, when the number of users is more than 3, data is transmitted in two times. Of course, it is possible to change the MTU value through additional setting. However, since it is difficult to change the MTU value in the mobile HMD, and its maximum value in the used network switch may be limited to 1500 bytes, it is assumed to be 1500 bytes in this study.

### *3.4 Smartphone HMD Client*

The smartphone HMD client of this study was developed based on Unity3D engine. It receives the skeleton data from the Kinect server, displays the avatar in the virtual environment, and sends the rotation data of the HMD to the Kinect server. We used Google Cardboard SDK for smartphone-based HMD implementation.

The character model to be used in the HMD client is Unitychan, the mascot character of Unity3D. There is one problem here, the skeletons from Kinect and the Unitychan have a different part in their structure (see Figure 8).

Therefore, the rotation data of the joint should be modified to fit the coordinate system of the model. In this study, we found three joint data of Kinect skeleton nearest to the joints of Unitychan, which is structurally different from that of Kinect's skeleton joint and

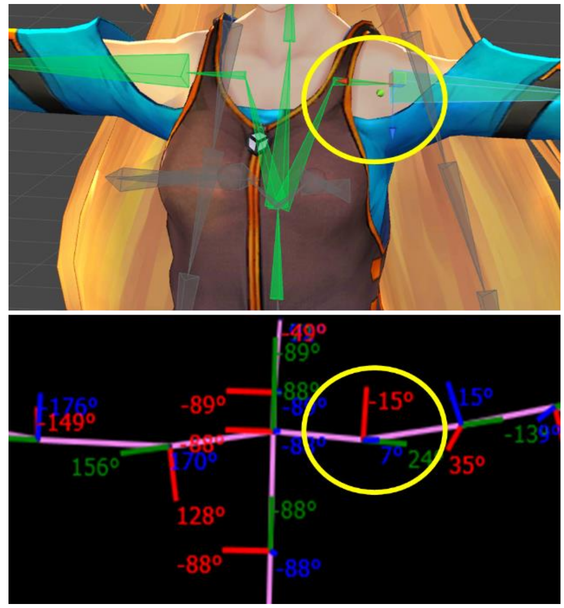



**Figure 8.** **Structure difference between the right shoulder of Unitychan and the right shoulder of the skeleton received from Kinect.**

interpolated their values. As a result, we found that Unitychan's skeleton works normally.

## 4 EXPERIMENTAL RESULTS

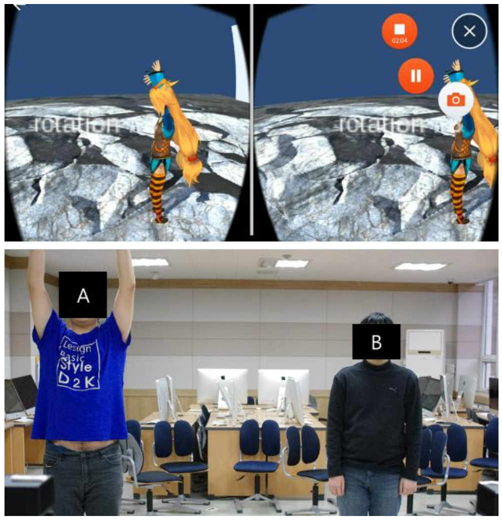
IN this experiment, ipTIME A604 model was used for network system construction and its MTU was set to 1500. The reason for fixing the MTU value is to prevent the router from trying to variably adjust the MTU for optimization.

The experiment shown in the Figure 9 is the view from the viewpoint of B through the smartphone HMD after instructing the experimenter A and B to move in front of Kinect and perform a simple operation. For the convenience of experimentation, we made it possible to move the user's point of view recognized by Kinect freely using the magnetic switch of the Google card board.

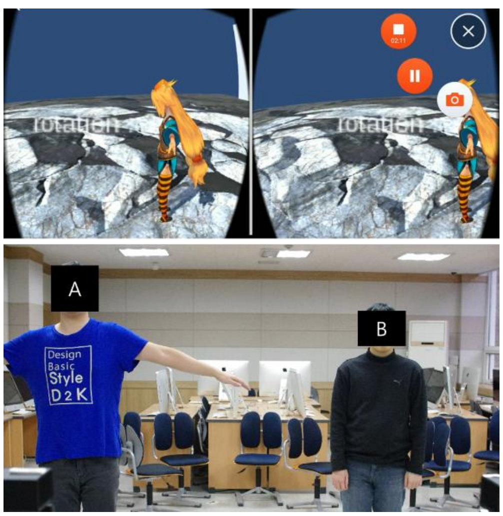
### *4.1 Multiple User Recognition*

With Kinect V2, it is possible to track up to six users simultaneously. The number of participants in this experiment was 5, and it was confirmed that they were recognized as stable as shown in Figure 11. However, as mentioned above, each experimenter is represented by the same avatar in the virtual environment, although the body shape is different. This part will also improve the reality by introducing an algorithm that creates an avatar for each body type of the experimenter.

### 4.2 Test for Simple Posture




(a) Posture #1




(b) Posture #2

**Figure 9.** VR character according to posture change (Point of view from B).

## 5 LIMITATIONS

WE carried out experiments to shake hands while wearing mobile HMDs. As shown in the Figure 10, the handshake gesture itself operated normally in the virtual environment, but the hand was not in contact with each other. This is a problem that occurs because the avatar model is not adapted to the actual body shape of the experimenter. This study focused on the recognition of many users in a virtual environment, so we could not predict this problem in advance. Of course, using the avatar adjusted to the body shape of the experimenter is expected to solve this problem.

**Figure 10.** Shaking hands in a virtual environment and reality.


**Figure 11.** Multiple user recognition result for 5 persons.

## 6 CONCLUSION

IN this paper, we have implemented local space VR environment by tracking the movement of many users wearing smartphone-based mobile HMD by using several Kinects and applying it to the virtual avatar. Conventional fixed VR environment using treadmill requires one device per person, but using the proposed system, many people can enjoy VR contents in a given area without expensive devices other than Kinect. As a result, it is possible to save a lot of space for implementing a virtual environment, and it is expected to bring a new trend of the VR environment. In addition, since the user's smartphone is configured with HMD by using Google cardboard, hygiene problem is solved, and there is a lot of room to be utilized as a virtual reality platform in public places.

However, Kinect was not originally designed for motion tracking, so it has a somewhat larger error compared to professional equipment. In the future, we would like to increase the accuracy by using equipment that specializes in motion tracking at a cheaper price, and to raise the reality by creating an avatar tailored to the individual body shape of the user.

## 7 ACKNOWLEDGMENT

THIS research was supported Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2017R1D1A1B03035576).

## 8 REFERENCES AND FOOTNOTES


S. Charles (2016). Real-time human movement mapping to a virtual environment, *IEEE Region 10 Symposium (TENSYMP)*.

C. Chen, R. Jafari, and N. Kehtarnavaz (2015). Improving Human Action Recognition Using Fusion of Depth Camera and Inertial Sensors, *IEEE Transactions on Human-Machine Systems*. 45(1), 51-61.

C. Chen, R. Jafari, and N. Kehtarnavaz (2016). A Real-Time Human Action Recognition System Using Depth and Inertial Sensor Fusion, *IEEE Sensors Journal*. 16(3), 773-781.

C. Chen, R. Jafari, and N. Kehtarnavaz (2017). A survey of depth and inertial sensor fusion for human action recognition, *Multimedia Tools and Applications*. 76(3), 4405-4425.

J. Han, L. Shao, D. Xu, and J. Shotton (2013). Enhanced Computer Vision With Microsoft Kinect Sensor: A Review, *IEEE Transactions on Cybernetics*. 43(5), 1318-1334.

B. Hariharan, S. Padmini, and U. Gopalakrishnan (2014). Gesture Recognition Using Kinect in a Virtual Classroom Environment, *Fourth International Conference on Digital Information and Communication Technology and Its Applications (DICTAP)*.

H. Heickal, T. Zhang, and M. Hasanuzzaman (2013). Real-time 3D full body motion gesture recognition, *IEEE International Conference on Robotics and Biomimetics (ROBIO)*.

F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao (2015). Multi-layered Gesture Recognition with Kinect, *Journal of Machine Learning Research*. 16.

X. Jin, Y. Yao, Q. Jiang, X. Huang, J. Zhang, X. Zhang, and K. Zhang (2015). Virtual Personal Trainer via the Kinect Sensor, *IEEE 16th International Conference on Communication Technology (ICCT)*.

W. Kim and M. Kim (2015). Visualization of Skeleton Tracking Range for Multiple Kinect Sensors, *IEEE Region 10 Symposium (TENSYMP)*.

P. Kumar, R. Saini, M. Yadava, P. Roy, D.P. Dogra, and R. Balasubramanian (2017). Virtual trainer with real-time feedback using kinect sensor, *International Conference on Communications and Information Sciences*.

R.A. Rane, N. Potnis, S. Sansare, N. Mokashi, and S. Patil (2018). Virtual Personal Trainer using Microsoft Kinect and Machine Learning, *International Journal of Computer Applications*. 179(11), 23-28.

P.T. Wilson, K. Nguyen, A. Harris, and B. Williams (2014). Walking in place using the Microsoft Kinect to explore a large VE, *In Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, 27-33.


## 9 DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the authors.

## 10 NOTES ON CONTRIBUTORS

**S.-K. Kim** received Ph.D in CS, Korea University in 2006. He was an advanced researcher in Samsung Electronics from 2006 to 2008. He is a professor in the department of game engineering in Paichai University. His research interests include geometric modeling.
nicesk@gmail.com

**C.-H. Lee** received MS in CS, Hallym University in 2017. He manages a VR contents production company. His research interest is virtual reality.
41774@hallym.ac.kr

**S.-J. Kim** received Ph.D in CS, Korea University in 2003. She is a professor in the department of convergence software in Hallym University. Her research interests include geometric model, scientific visualization, and virtual reality.
sunkim@hallym.ac.kr,

**C-.G. Song** received Ph.D in CS, University of Oklahoma in 1992. He is a professor in the department of convergence software in Hallym University. His research interests include virtual reality and HCI.
cgsong@hallym.ac.kr,

**J. Lee** received Ph.D in CS, Korea University in 2006. He is a professor in the department of convergence software in Hallym University. His research interests include virtual and augmented reality.
airjung@gmail.com