

# Low cost haptic and motion based mixed reality peripheral interface

Cite as: AIP Conference Proceedings **2137**, 030005 (2019); <https://doi.org/10.1063/1.5120998>  
Published Online: 07 August 2019

Wei Jun Lou, Aun Shih Teh, and Swee King Phang



View Online



Export Citation

**AIP** | Conference Proceedings

Get **30% off** all  
print proceedings!

Enter Promotion Code **PDF30** at checkout



# Low Cost Haptic and Motion Based Mixed Reality Peripheral Interface

Wei Jun Lou<sup>1, a)</sup>, Aun Shih Teh<sup>1, b)</sup>, Swee King Phang<sup>1, c)</sup>

<sup>1</sup>*School of Engineering, Taylor's University, Taylor's Lakeside Campus,  
No.1 Jalan Taylor's, 47500 Subang Jaya, Selangor DE, Malaysia*

<sup>a)</sup>Corresponding author: weijunlou96@gmail.com

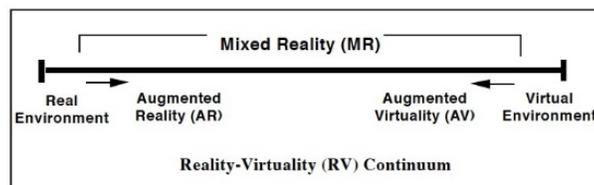
<sup>b)</sup>aunshih.teh@taylors.edu.my

<sup>c)</sup>sweeking.phang@taylors.edu.my

**Abstract.** With the advancement of mixed reality technologies, such as Virtual Reality (VR) and Augmented Reality (AR), greater applications based on these technologies are expected to become more prevalent in society. Mixed reality enables an immersive digital environment experience through visuals and sound and when coupled with Cutaneous haptic feedback, which provide the touch sensation with the virtual environment, allows the creation of an environment that closely emulates the real world. Haptic devices which bridges the physical sensation of these worlds have been continuously enhanced with high-end products being developed such as the Oculus Rift and HTC Vive. This paper investigates the viability and performance of a low cost two-finger wearable haptic device to provide natural and instinctive virtual reality experience. The device is made up of an MPU6050 tracker, disc type vibration motor haptic actuators and flex sensors to record finger joint movement and characterise gesture movement. Arduino UNO microcontroller is used to communicate with the virtual environment rendered by the Unity 3D software. Results indicate that the haptic device is able to communicate and respond with the virtual environment but a lag in the overall hardware reduces the seamlessness perception with the virtual environment.

## INTRODUCTION

Mixed Reality (MR) merges physical reality with the digital world in real time, resulting in a hybrid digital reality world. Paul Milgram and Fumio Kishino coined the term MR back in 1994 and explained the nuance of the reality-virtuality continuum by introducing a mixed reality spectrum as shown in Fig. 1 below [1].



**FIGURE 1.** Mixed Reality Spectrum by Paul Milgram and Fumio Kishino

MR consists of Augmented Reality (AR) which overlays digital objects into the real physical world, and augmented virtuality (AV) which project our physical self in a digital world. Such virtual reality (VR) technologies in today's modern day is able to provide better visual, spatial, and locomotion perceptions through apparatuses such as head mounted display (HMD) and omnidirectional treadmill [2]. These advancements of these apparatuses is critical as immersion is an important factor in order to experience an instinctive and natural application of the digital world. In order to achieve immersive, touch sensation is introduced into VR, complimentary to our visual and sound perceptions.

A haptic device is a device that provides cutaneous force feedback by an actuator that is triggered through the interaction with the virtual environment. It can be differentiated into two main categories, grounded and ungrounded. Grounded haptic device is mounted onto a surface or platform and driven by a mechanical system as its actuator. Examples of existing grounded haptic technologies include the Phantom OMNI by SensAble Technologies, and Virtuoso 6D Desktop from Haption [3] [4]. Ungrounded haptic device, which is the focus of this paper, is typically a handheld or wearable haptic device. There are prior development of ungrounded haptic devices such as DextrES by Hinchet et al., CLAW haptic controller, and Fluidic Elastomer Actuators by Barreiros et al. [5] [6] [7]. Their haptic devices were associated with high end specifications such as high definition and high fps camera, and Oculus Rift tracker, with high performances in term of latent time delay, navigation and location positioning. The development of such devices however comes at a higher costs both in development and for the end user. This paper presents an alternative model of haptic device which uses low cost components, such as MPU6050, disc type vibration motor, flex sensor, and Arduino UNO, and investigates the latent time delay of the hardware when interacting with Unity 3D software which indirectly affects the immersiveness.

## METHODS

### Virtual Environment

Unity 3D software is a real-time game engine and game building application. It had been used in many virtual applications in various fields from mobile game development to a virtual practice platform for chemical engineers [8]. A virtual environment is simulated with a simple square platform with perimeter walls as shown in Fig. 2 below. In addition, a virtual hand and a virtual ball are created for the purpose of interacting and as a platform for testing haptic feedback through the actions of grabbing and tapping.

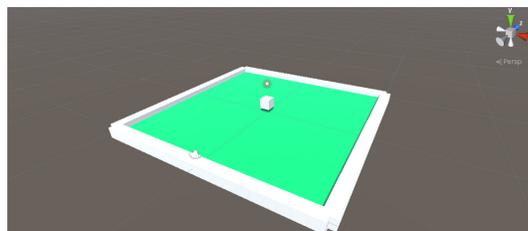


FIGURE 2. Simulated platform in Unity 3D software

### Haptic Glove

The basic components of a haptic glove include a spatial and orientation tracking device, haptic feedback actuator, finger-joint movement detector and a communication medium to connect the haptic device into a virtual environment. In comparison to high-end R&D which utilizes the vive tracker as a tracking device, the MPU6050 micro-electromechanical system (MEMS) is implemented as a cheaper alternative in this study. The MPU6050 comprises of an accelerometer and gyroscope in one single IC whose purpose is to record the location and orientation of the haptic glove and synchronize it with the virtual hand in the simulated virtual environment. It is positioned at the center of back palm as to measure the rotation of the wrist. As for haptic feedback actuator, linear resonant actuator (LRA) disc type vibration motor is used and complemented by a DRV2605L haptic motor controller to cater for different characteristic of vibration for the corresponding action performed in the virtual environment. It is placed only on the pressure-sensitive tips of thumb and index finger. A flex sensor is also attached along the index finger and thumb to record and control the joint movement of the virtual hand. Gestures, such as ‘reset position’ and ‘switch off’, were initially characterized based on the flex sensor and orientation of the haptic device. Combined as a two-finger haptic device that is linked up with the Unity 3D virtual hand through Arduino UNO several gesture motion characteristics were documented. The Arduino UNO served as a data port where all the data information is collected and sent to the

Unity 3D via a USB cable which also receives the virtual object interaction and cues the correct haptic feedback response to the vibration motor.

### **Limitation**

As a compensation for using low cost components, the developed two-finger haptic device came with three limitations. Firstly, the haptic device has a limited degree of freedom in moving the virtual hand. Since the MPU6050, which is the accelerometer and gyroscope, is placed on the center of back palm, it is unable to detect yawing and pitching of the wrist. They are recorded as moving the virtual hand left or right, and up or down respectively. Besides that, the haptic response is localized only at the fingertips and the effective area is determined by the size of the vibration motor. Lastly, the haptic device only renders cutaneous haptic response but not force feedback, which is meant to produce resistive force to help the user have better perception of size and shape of the virtual object.

### **Basic Test Gestures**

The gestures that are coded and implemented in this haptic device are ‘reset position’ and ‘switch off’ gestures. Firstly, ‘reset position’ gesture is used to reset the position and orientation of the virtual hand back to a neutral hand resting pose at origin point. The trigger for this gesture is achieved by holding a fist with palm facing downward for 3 seconds and the thumb and index fingers’ flex sensors have to record an angle value of more than 70° for the whole duration. After resetting, the haptic device will have a 5-second window to recalibrate the offset value of accelerometer and gyroscope. ‘Switch off’ gesture is used to switch off and disconnect the haptic device. This also resets the position and orientation of the virtual hand back to its original position. The trigger for this gesture is achieved by holding a fist with palm facing upward for 7 seconds, and the flex sensors have to record an angle value of more than 70° for the whole duration.

### **Process Flowchart**

The flowchart in Fig. 3 shows operation of the device. When the Arduino UNO is powered up by connecting to a power source, e.g. laptop, via a USB cable, the haptic device is also powered up at the same time. First process to run is the calibration of the MPU6050. The user places his/her hand on a table surface in a resting pose to compute the offset and synchronize with the virtual hand. When the user moves his/her hand, the MPU6050 senses the movement and/or rotation and sends the information to Arduino UNO and to Unity3D which then moves the virtual hand accordingly. Besides that, when the virtual hand grabs the virtual ball in the Unity3D simulated virtual environment, a virtual contact is detected between the index finger and thumb of the virtual hand. The virtual contact prompts the vibration motor to vibrate on the haptic device and the haptic response is characterized by the DRV2605L haptic motor controller. Lastly, when the user clenches his/her fist, where the flex sensor records an angle value of more than 70°, with the palm is facing downwards, and the user hold for 3 seconds; this prompts a ‘reset position’ gesture and resets the position and orientation of the virtual hand. If the palm is facing downwards, and the user holds for 7 seconds, it disconnects the haptic device and reset the position and orientation of the virtual hand.

## **RESULT AND DISCUSSION**

### **Two-finger Haptic Glove**

Fig. 4 shows the developed two-finger haptic glove, with two flex sensors and on thumb and index finger each, two vibration disc motors on the fingertips, MPU6050 situated at the center of back palm, and connected to an Arduino strapped at the wrist.

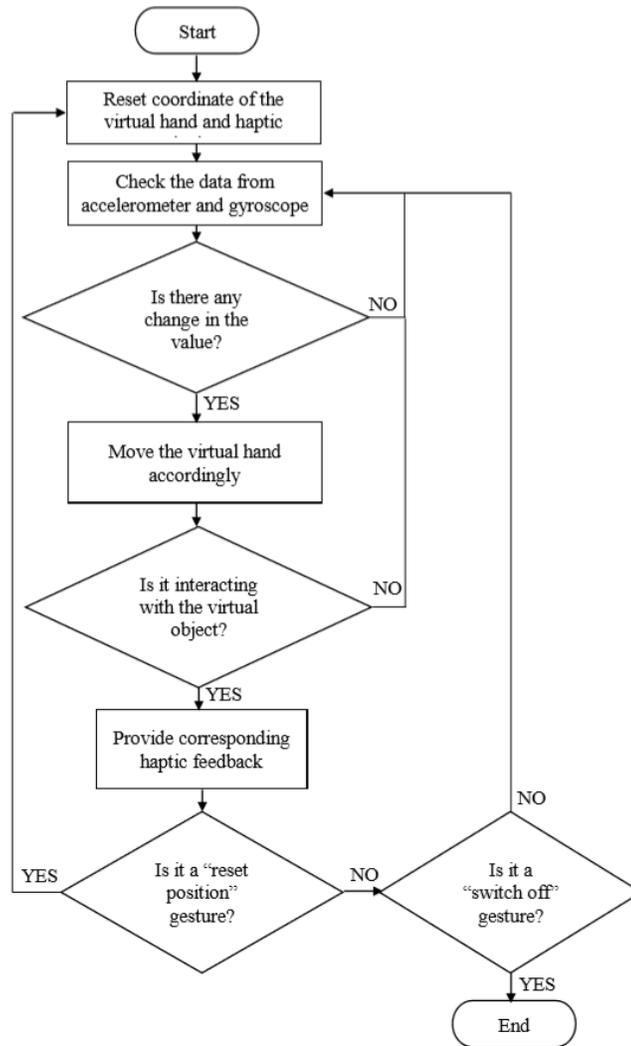


FIGURE 3. Process flowchart of the haptic device

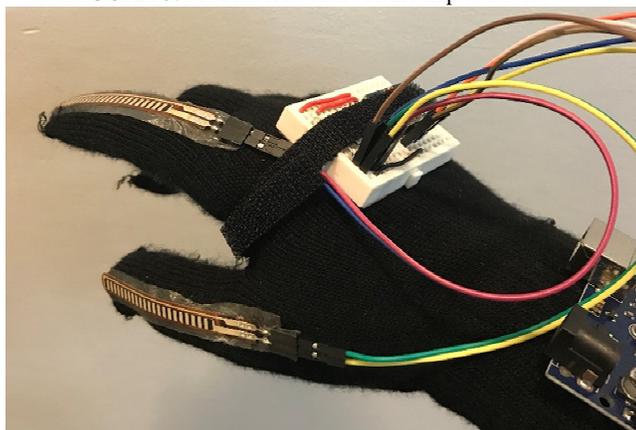


FIGURE 4. The developed two-finger haptic glove.

## Haptic Feedback

The haptic responses are provided according to the interaction with the virtual ball in the virtual environment. A light intensity tapping vibration on both the vibration motors is felt when the virtual hand presses or taps the virtual ball for a short time period. When grabbing the virtual ball, light intensity vibration is felt and the duration of the vibration depends on how long the grabbing action is. Lastly, a strong intensity is felt when the user swipes or pushes the virtual ball along the platform. The duration of vibration is based on how long the swiping or pushing motion is.

## Gesture

‘Switch off’ and ‘Reset position’ gestures were carried out successfully which powered off and disconnected the haptic device, and the position and orientation was reset to its origin respectively.

## Latent Time Delay

Tests were carried out to measure the latent time delay when moving the haptic device. Its procedures are as below:

1. The haptic device was powered up.
2. The position and orientation of the haptic device was reset.
3. The haptic device is moved towards x-direction (extending arms), y-direction (moving arms to the left), and z-direction (moving arms down).
4. A stopwatch was used to measure the time taken for the virtual hand to moved.
5. Steps 2 to 4 were repeated for 3 more times to obtain an average value.

The result for x-direction, y-direction, and z-direction were tabulated in Table 1, Table 2, and Table 3 below.

**TABLE 1.** Time taken for synchronization in x-direction.

| Tries Number | Time taken for synchronization (s) |
|--------------|------------------------------------|
| 1            | 2.267                              |
| 2            | 2.783                              |
| 3            | 2.568                              |

**TABLE 2.** Time taken for synchronization in y-direction.

| Tries Number | Time taken for synchronization (s) |
|--------------|------------------------------------|
| 1            | 2.357                              |
| 2            | 2.865                              |
| 3            | 2.456                              |

**TABLE 3.** Time taken for synchronization in z-direction.

| Tries Number | Time taken for synchronization (s) |
|--------------|------------------------------------|
| 1            | 2.195                              |
| 2            | 2.884                              |
| 3            | 2.459                              |

The average of the time taken for synchronization was calculated by using *Equation (1)*.

$$t_{ave} = \frac{t_1+t_2+t_3}{3} \quad (1)$$

Table 4 shows the result of calculation for the average time taken for synchronization.

**TABLE 4.** Average time taken for synchronization for x-direction, y-direction, and z-direction.

| Direction | Average Time taken for synchronization (s) |
|-----------|--|
| X         | 2.539                                      |
| Y         | 2.559                                      |
| Z         | 2.513                                      |

As shown in *Table 4*, the average time taken for synchronization was approximately 2.5 seconds. Possible causes of this length of time delay was due to heavy computational load in Arduino UNO. MPU6050 records raw data from the accelerometer and gyroscope. The lag was attributed to the computation and extraction of the required information, such as acceleration and orientation in x, y, and z directions. Furthermore, internal noise within the system may mask and alter the information, causing some data and package loss. Thus, more studies are required to analyse the data received and transmitted by the Arduino to the Unity3D software. This would include attributes such as data error and packet loss which could lead to a better management of the sensor signals.

## CONCLUSION

This haptic device is developed with the objective of decreasing the manufacturing cost while maintaining the performance and providing immersive VR experience to the user. Although the former objective is a success, the latter has not been achieved due to a lag in response attributed to the heavy computation load on Arduino UNO and Unity 3D software, and internal noise of the cable that mask and change the nature of the data. However, this haptic device is able to provide cutaneous feedback force at the pressure-sensitive fingertips through disc type vibration motor, and able to perform ‘reset position’ and ‘switch off’ gesture. Further troubleshooting and improvement will be carried out in order to decrease the latent time delay. Recommended approach is to use a more dedicated microcontroller but maintaining a low cost and to reduce the internal noise in the system.

## REFERENCES

1. P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays," *IEICE Transactions on Information Systems*, Vols. E77-D, no. 12, pp. 1321-1329, 1994.
2. S. H. Pyo, H. S. Lee, B. M. Phu, S. J. Park and J. W. Yoon, "Development of an Fast-Omnidirectional Treadmill (F-ODT) for Immersive Locomotion Interface," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018.
3. A. Jarillo-Silva, O. A. Dominguez-Ramirez, V. Parra-Vega and J. P. Ordaz-Oliver, "PHANToM OMNI Haptic Device: Kinematic and Manipulability," in *Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, 2009.
4. "Virtuose 6D," [Online]. Available: <https://www.haption.com/fr/products-fr/virtuose-6d-fr.html>. [Accessed 23 May 2019].
5. R. Hinchet, V. Vechev, H. Shea and O. Hilliges, "DextrES: Wearable Haptic Feedback for Grasping in VR via a Thin Form-Factor Electrostatic Brake," in *31st Annual ACM Symposium*, New York, 2018.
6. I. Choi, E. Ofek, H. Benko, M. Sinclair and C. Holz, "CLAW: A Multifunctional Handheld Haptic Controller for Grasping, Touching and Triggering in Virtual Reality," in *CHI Conference on Human Factors in Computing Systems*, Montreal, 2018.
7. J. Barreiros, H. Claire, B. Peele, O. Shapira, J. Spjut, D. Luebke, M. Jung and R. Shepherd, "Fluidic Elastomer Actuators for Haptic Interactions in Virtual Reality," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 277-284, 2019.
8. S.-G. Ouyang, G. Wang, J.-Y. Yao, G.-H.-W. Zhu, Z.-Y. Liu and C. Feng, "A Unity3D-based Interactive Three-dimensional Virtual Practice Platform for Chemical Engineering," *Computer Applications in Engineering Education*, pp. 91-100, 2017.