WIRELESS CONTROLLER FOR INTERACTIVE VIRTUAL REALITY GAMES

S. Kazempourradi, S. O. Ozturk, M. B. Erdemli, B. Gulerce, M. S. Yazici, L. Ozmen, S. I. Tuncer, C. H. Dagidir, E. Ulusoy, H. Urey

Koc University, Electrical Engineering Department, Optical Microsystems Laboratory

ABSTRACT

An array of tiny, low-cost, stand-alone and wireless inertial motion sensor units is designed and fabricated. These sensor units recognise gestures of a user, enabling a comfort control in game applications. Despite the wide usage of motion sensor units in various applications, we utilize an array of detached and low-cost controller units and an Oculus Rift DK2 to develop two VR games, for the first time. In one application, the user can control a spaceship movements by its hand movements. The other game is a first-person shooting game, in which an array of sensors are used for aiming and shooting purpose. This type of control provides the feeling of full immersion inside a VR environment. The developed sensor unit is a promising controller for a broad range of applications in virtual and augmented reality industry.

Index Terms — Motion Sensor, Virtual Reality, VR Games, Oculus Rift, Head-mounted Display

1. INTRODUCTION

Virtual reality (VR) games gain significant interest in the last few years, because they provide the idea of full immersion in the three-dimensional virtual environment. However, most of VR games use known controllers, such as a keyboard or a mouse. The expectation of full immersion cannot be met by such devices. Hence, the development of smart devices is necessary to enable natural control experience in the virtual environment.

In the literature, there are various examples of integrating a smart sensor with a head-mounted display (HMD) to provide a better control in VR games. Lee et al. [1] utilize Oculus Rift, Kinect and a smart phone for developing a VR game. Tregillus et al. [2] use a smart phone to develop a walk-in-place (WIP) sensor. An overview of other virtual reality technologies and controllers is presented in Boas's work [3].

In this work, we developed a tiny, low-cost, and wireless inertial motion sensor unit to sense hand movements and gestures of a user. We utilize this device and an Oculus Rift DK2 to develop a VR game and demonstrate the capabilities of this unit. The developed device can be used in various applications, for instance as a WIP sensor or as a sensor for correcting the head and chest position in VR environment. For the first time, we used an array of detached sensor to develop a first-person shoot virtual reality game [4]. In the following sections, we describe the system and we look into the hardware and software parts in details. Finally, we summarize the work.

2. SYSTEM DESCRIPTION

For the development of the game, we utilize Oculus Rift DK2 as a HMD and our sensor unit as a game controller. The game engine is Unity 3D. In this game, a user controls the direction of a spaceship

Figure 1. The 9-axis sensor data is processed in the micro-controller and goes into BLE chip. The BLE chip in the receiver part, receive the data and send it via USB port to PC. The data is embedded in Unity and the PC runs the game on Oculus Rift.

Figure 2. a) Schematic of the sensor unit. b) Top view of manufactured sensor unit. c) Bottom view of manufactured sensor unit.

by his/her hand movements. The flowchart in Fig. 1 demonstrates the system.

2.1. Hardware

In this part, we describe the hardware part of this work.

2.1.1. Oculus Rift DK2

We use Oculus Rift DK2, shown in Fig. 2(a) as the HMD. The Oculus has a high-resolution (960×1080 per eye) low-persistence OLED display, high refresh rate, head and positional tracker, and it utilizes a lens to provide high field-of-view images. The Rift must be connected by a cable to a PC running Microsoft Windows that will run the game [5].

2.1.2. Sensor Unit

All sensing is performed by a single board, as shown in Fig. 2. The control unit performs the 9-axis motion measurements through three sensors that are embedded in a single unit and transmits acquired data to Unity environment. The receiver unit is responsible

Figure 3. a) Main spaceship model in Unity software revealing the camera position. b) Move to left action, after the user rotates the sensor in the similar manner. c) Move to right action. d) Move upward action.

to get the data from control unit by utilizing a Bluetooth chip and a serial to USB converter module. This unit receives the motion data from sensor unit and provides the instantaneous and synchronous connection between Unity and sensor unit. The main components are:

- Texas Instrument MSP430G2553 micro-controller
- Invensense MPU 9250 sensor
- Microchip RN4020 Bluetooth low energy (BLE)

The sensor has two dies in a single package providing 9-axis motion measurements. The micro-controller interfaces between the sensor chip and the BLE chip. The communication between the device and PC is performed over Bluetooth 4.1 protocol. The schematic of the sensor can be found in Fig. 2(b). We also use another BLE connected to PC via USB port, which acts as a receiver. The sensor unit is equipped with a rechargeable battery that makes it freely usable in any position. The photos of manufactured sensor unit can be found in Fig. 2(c) and (d). The size of the current version of the sensor is 3.5×5 cm. We can shrink the size by half in the next version, by eliminating test pins.

Code Composer Studio (CCS) is an integrated development environment (IDE) that supports TI micro-controller and Embedded Processors portfolio. The CCS software comprises a suite of tools used to develop and debug embedded applications. It includes an optimizing C/C++ compiler, source code editor, project build environment, debugger and profiler. The CCS software is used for programming of TI MSP430G micro-controller. The micro-controller enables the data flow between sensors (MPU-9250) and the Blue-tooth (RN4020 BLE) module. When the connection is established between the receiver unit and the sensor unit, RN4020 BLE modules in the control and receiver units starts transmitting and receiving the sensor data, respectively. Once the sensor stops transmitting the data, the controller recalibrates itself and shuts down.

2.2. Software

The game is implemented using the Unity 3D engine. We use some free spaceship models (Fig. 3(a)), where one of them is the main model and the others act as enemies. The environment of the game is created using a sky box that is wrapped around the scene, as shown in Fig. 3(b).

The sensor data is processed in Unity. We attach the sensor on top of the user's hand. The software maps the hand movements

Figure 4. a) To improve sensor accuracy, we low-pass filter the signals of accelerometer and magnetometer and high-pass filter the gyroscope data. b) x, y, and z component of sensor data before and after applying filter. All data are normalized in the plots.

to the spaceship movements. Four sample moves are shown in Fig. 3(c)-(f). Environment includes planets, a sun and asteroids. The spaceship starts firing when enemies enter in its field of view.

The raw data output of accelerometer, gyroscope and magnetometer are inaccurate. Especially, the output from the magnetic field sensor includes a lot of noise. To avoid gyroscope drifts and noisy orientation, the gyroscope output is used to compute short-time orientation, while the accelerometer and magnetometer data is used as support information over long periods, as shown in Fig. 4 (a). Equivalently, we low-pass filter the signals of accelerometer and magnetometer and high-pass filter the gyroscope data [6]. The offset correction and timer part is done the micro-controller before feeding into BLE chip. The filtering parts implemented via C# and embedded in Unity software. For low-pass filtering of data, we sample accelerometer and magnetometer data at 0.5 sec intervals and combine it with gyroscope data. As illustrated in Fig. 4(b), the filtering and combination of signals will reduce the noise and increase the accuracy.

To test an array of IMU sensors, we designed a first person shooting (FPS) game in which user experiences full immersion sense. A user is equipped with two control units. One of the sensor units is used to control the user's arm for aiming purpose. The other sensor unit is used for shooting.

In the game environment we create a number of boxes (targets), as shown in Fig. 5. These boxes have three destroyable hit-points. Utilizing a skybox and adding the fire audio enhances the full immersion sense. Our game has a two minute time limit and a score counter. Each successful shot counts as 1000 points and the player can see his/her score when time is up. Moreover, we add a ray cast that appears on the margin of the gun and plots a straight line through the end of the gun.

3. SUMMARY AND CONCLUSIONS

Figure 5. A scene from Unity software revealing the camera position, the user virtual arm and target boxes.

We implemented a VR game that utilizes Oculus Rift and a stand-alone motion sensor. The user can smoothly control the movements of a spaceship in the virtual environment by simple hand movements. The design parameters of motion sensor are forced to make it stand-alone, low-cost, and tiny. The communication of the sensor and the PC is satisfied with a pair of low energy Bluetooth modules, which increase the capability of the sensor to be used in various disciplines and situations. On the software part, we filter sensor data and embed the filtered data into the VR game. The developed sensor unit can become a promising user interface for different VR applications.

Acknowledgement

This work is supported by European Research Council under the Seventh Framework Programme (FP7/2007–2013) ERC grant agreement no 340200, with acronym Wear3D.

4. REFERENCES

- [1] J. Lee H. Lim D. Lee, K. Baek, "A development of virtual reality game utilizing kinect, oculus rift and smartphones," *J. App. Eng. Res.*, vol. 11, no. 2, pp. 29–33, 2016.
- [2] E. Folmer S. Tregillus, "Vr-step: Walking-in-place using inertial sensing for hands free navigation in mobile vr environment," *Proc. CHI conference*, 2016.
- [3] Y. Boas, "Overview of virtual reality technologies," *Proc. Interactive Multimedia Conference*, 2013.
- [4] S. Kazempourradi S. O. Ozturk M. S. Yazici L. Ozmen S. I. Tuncer C. H. Dagidir E. Ulusoy, H. Urey, "Development of a tiny, low-cost and wireless motion sensor for interacting with virtual reality games," *Egocentric Perception, Interaction and Computing Workshop (EPIC), European Conference on Computer Vision (ECCV)*, vol. 1, pp. 12–13, Amsterdam, Netherlands 2016.
- [5] Web, , "http://www.oculus.com."
- [6] J. Lawitzki, "Application of dynamic binaural signals in acoustic games," *Master thesis at Hochschule der Medien Stuttgart*, 2012.