InstantGaming: Playing Somatosensory Games Using Smartwatches and Portable Devices

Meng-Shiuan Pan[1], Wu-Chun Hsu[1], Chun-Hung Liu[1], Ku-Chen Huang[2], and Chien-Fu Cheng[1]

[1]Department of Computer Science and Information Engineering, Tamkang University, New Taipei City, Taiwan
[2]Office of Physical Education, Tamkang University, New Taipei City, Taiwan
e-mail: mspan@mail.tku.edu.tw

Abstract

In recent years, playing somatosensory game is one of popular home entertainments. Many game companies design and develop lots of somatosensory games such as fitness games, sport games, and fighting games. We observe that to play somatosensory games, players have to buy specific devices or equipment (e.g., Kinect/XBOX or Wii). These devices and equipment are expensive, and a vast space is needed when playing. To overcome the above drawbacks, in this paper, we propose to use smartwatches and portable devices to play somatosensory games. In our design, players wear smart- watches, which sense the changes on players' bodies by accelerometers. By the collected acceleration sensory data, the proposed posture detection scheme implemented on the smartwatch can judge the performed postures of players. The decided posture will then deliver to the portable device (e.g., smartphone or tablet), which can show the reaction of the game. In this work, we implement a fighting game to demonstrate our design.

Key words: somatosensory games, smart devices, posture detection

Introduction

In recent years, somatosensory games are popular home entertainments. There are many somatosensory games that are in different types, such as fitness games, sport games, and fighting games. But, to play somatosensory games, players have to buy specific devices or equipment (e.g., Kinect/XBOX [1] or Wii [2]). These devices and equipment are expensive, and a vast space is needed when playing.

According to the development of smart devices, many people have multiple portable devices, e.g., smartphones, smart watches, and tablets. To overcome the above drawbacks on playing somatosensory games, in this paper, we propose to utilize users' portable devices to play somatosensory games. In our design, players wear smart- watches, which sense the changes on players' bodies by accelerometers. The smartphones or tablets are used to display game scenes and screens. By our solution, people can play somatosensory games anytime and anywhere with others, and will not be restricted by locations.

In this paper, we design a somatosensory game by smart watches and handheld devices. The designed game is a fighting game, which can be played by two players. These two players fight with each other by doing the designated gestures or postures. The game is easy to learn and play. When implementation, we use Android Studio [3] to write programs, which are used to handle sensory data from smart watches' inertial sensors. Then, we use the gathered acceleration sensory values to detect player's postures. When a posture is detected, the designed program then compares the detected posture with the movements that are recorded in the movement database. Based on the movement database, if the detected posture is similar to the recorded movement, the designed program takes the performed posture is a trick in our game. Moreover, we combine Android wear API [4] with Unity [5] to achieve data exchanges between smart watches and other Android handheld devices. We use MAYA [6] to build skeletons and animations of the game's characters. If a trick is identified, the corresponding character in the game will be controlled by Unity, and then can perform animations on game screen. Our solution has the following characteristics:

- Portable and popular: Users can play somatosensory games easily by using smart handheld devices. Instead of finding a large space and carrying specific devices to play games, our design allows players to play somatosensory games at anytime and anywhere.
- Interactive: Two players can play fighting games by lightweight smart watches. They can see instant movement of the controlled characters on the screen of their smartphones or tablets.

The remainder of this paper is organized as below. Section 2 describes the system architecture. Section 3 introduces the designed posture detection scheme. Section 4 and Section 5 indicate the detailed implementation and demonstration, respectively. Finally, Section 6 concludes this paper.

System Architecture

Figure 1 shows the system architecture, which can be divided into smart watch side and handheld device side. The smart watch side contains the designed posture detection module. The handheld device side contains a unity module, which indicates game screen, and a Bluetooth data processing module, which handles Bluetooth signals from smart watches. Before playing, smart watches connect to the handheld device through the Bluetooth protocol. In our design, the handheld device is the Bluetooth master. The master device connects to two smart watches at the same time. When the game is started, the posture detection module in the smart watch tries to capture the movement of players and decides if the performed movement is similar to the predefined tricks in our game. After a trick is identified, the trick information will be transmitted to the handheld device. The unity module in the handheld device will control the corresponding character draw by MAYA to perform trick animations on the game screen.

 ISBN 978-1-5090-4897-7

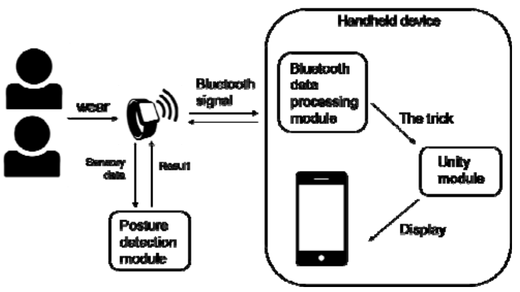

Figure 1: The system architecture

When gathering sensory data, our system further performs the following procedures. Assume that at a time t_i, the perceived linear acceleration values are defined as:

$$A_m(t_i) = [a_{x,ti}\ a_{y,ti}\ a_{z,ti}\ ]$$

To filter noises, in our system, we adopt exponential moving average concept to implement a low pass filter. When filtering, we set the weight of a new sensory value to be α. Let t_{i-1} be the previous sampling time and the sensory values in the corresponding $A_m(t_{i-1})$ been updated by the low pass filter. Then, this module update $A_m(t_i)$ by the following equation:

$$A_m(t_i) = \alpha \times A_m(t_i) + (1-\alpha) \times A_m(t_{i-1}).$$

Note that deciding a suitable α is an empirical task. When setting a small α, the output will adjust slowly and include less noise. On the other hand, when setting a large α, the output will adjust quickly and be sensitive to noise.

The Posture Detection Scheme

The posture detection scheme can be divided into two phases. The first one is the *training phase*. In this phase, we first collect linear acceleration values of the designated tricks. For each trick, we ask five people to perform the corresponding postures for at least 30 times. By the collected sensory data from these 5 people, we check if the perceived sensory data have similar patterns by the following steps.

1. *Normalize the length of the performed trick*: Recall that the time duration of the performed trick for every player may be different. So, in this step, the gathered sensory data will be normalized to equal length. Assume that a time segment of a trick is (t_i, t_j). This time segment will be enlarged to a predefined length T_d. We first calculate the magnification $l = T_d/(t_j - t_i)$. When enlarging a time segment, we need to modify the sensory values in the segment to preserve its characteristics. Let a_t be a sensory value in the segment (t_i, t_j). After enlarging the segment, the new location of the value a_t will be $t \times l$. By the displacement formula $S = 1/2 \times a \times t^2$, when the time scale is enlarged, it implies that a movement will spend more time to complete. Under the condition that the total movement is unchanged, when the time scale is enlarged by l times, the acceleration values will be degraded by l^2 times. As a result, the sensory value a_t has to change to a_t/l^2.
2. *Find characteristics and thresholds*: In this step, we mutually observe the characteristics of normalized sensory data. The key operation is to find some thresholds, which are representative to indicate the corresponding tricks. For a trick, we will find at least two thresholds, which are corresponding to the time and intensity.

Note that after the above step 2, a trick will be considered as a series of threshold values. For example, in a straight punch trick, the linear acceleration values of X axis should exceed 25 m/s^2 when 200 *ms*, and the ones of Y axis should be below -5 m/s^2 when 500 *ms*.

Then, the second phase is the *detection phase*. In this phase, the real-time gathered sensory data will be stored using a linked list. The proposed scheme traverses the linked list to check if the gathered sensory can satisfy the series of threshold values obtained from the training phase. The linked list will be updated by deleting those sensory values, which are processed.

Implementation Details

In this section, we describe two implementation details in our system. The first one is how to implement many-to-one communication by the Bluetooth protocol. Commercially, a smartwatch only connect to one smartphone. But, in order to play the designed somatosensory game, multiple smartwatches have to connect to a single smartphone or a single portable device. When implementing, the smartphone or the portable device will be the master device of the Bluetooth. Multiple smartwatches will be taken as the slaves, and connect to the master. We modify the Bluetooth protocol to add an ID field to indicate the corresponding smartwatch. By the ID of a smartwatch, the system can identify players.

The second is our system utilizes MAYA to draw the characters in our game. A character in our game is a model. We construct skeletons of the model, and for each posture, we have to mutually set the end points of the character's skeleton. Our animation is produced as the flow of producing a cartoon. Fig. 2 shows a procedure of producing an animation that the character punches by its left hand and then right hand, and performs a swing kick to end the trick. The produced animation will be controlled by Unity module. The unity module further handles the music and game screen.

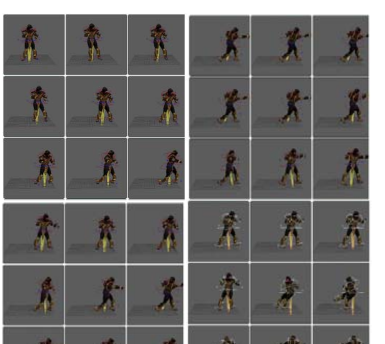

Figure 2: The production of an animation.

Demonstration

Fig. 3 is the welcome screen of the designed somatosensory game. When two players connect to the handheld device, the game can be started. Then, the handheld device will send command to players' smartwatches to enable the inertial sensors and the corresponding posture detection module. Fig. 4 indicates the game screen. When a player does the designated trick, the corresponding character in the game will show the animations. The health point (HP) of another player will decrease. The game finishes if the HP of a player becomes zero. Moreover, Fig. 6 shows the demonstration that a player is now playing our game.

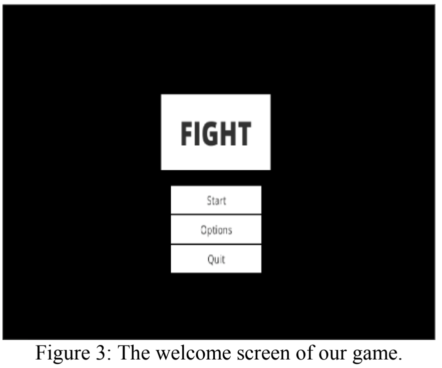

Figure 3: The welcome screen of our game.

Figure 4: The game screen.

Figure 5: A player is now playing our game.

Conclusions

In this paper, we have designed a somatosensory game, which can be played using smartwatches and smart handheld devices. The players do not need to buy specific devices or equipment when playing somatosensory games, and are able to play at any location. In the future, we are going to design more kinds of somatosensory games and enhance the user experience when playing.

.

Acknowledgements

M.-S. Pan was supported by the Ministry of Science and Technology under Grant 105-2221-E-032-046. K.-C Huang was support by the Ministry of Science and Technology under Grant 105-2410-H-032-064.

References

[1] Kinect/Xbox, https://www.microsoftstore.com/store/msusa/en_US/pdp/Kinect-Sensor-for-Xbox-One/productID.2267482500.
[2] Wii, https://en.wikipedia.org/wiki/Wii.
[3] Android Studio, https://developer.android.com/studio/index.html.
[4] Android wear API, https://developer.android.com/wear/index.html.
[5] Unity, https://unity3d.com/.
[6] MAYA, http://www.autodesk.com/products/maya/overview

 ISBN 978-1-5090-4897-7