Improvement on Immersive Motion Sensing Interfaces with Multiple Non-Contact Kinects and Contact Hydra Controller for HMD-Enabled VR Games

SangHoon Shin, Kyungeun Park*, Seok-Kyoo Kim, Juno Chang,

Department of Game, College of Software, Sangmyung University, 20, Hongimun 2-Gil, Jongno, Seoul, 110-743, Republic of Korea
Fax: +82-2-2287-0106
E-mail: ssh1038@nate.com
E-mail: skkim@smu.ac.kr
E-mail: jchang@smu.ac.kr

Department of Computer and Information Sciences Towson University, 8000 York Road, Towson, MD 21252-0001, U.S.A.
Fax: +1-410-704-3868
E-mail: kpark@towson.edu

*Corresponding author

Abstract: This research focuses on increasing usability of an augmented reality game with improved immersive motion sensing interfaces for head-mounted display (HMD) enabled virtual reality (VR) game. As the VR game users increasingly demand highly intensive immersion level, it is expected to minimize interference between the game players and the VR game. For this, two types of sensing interfaces are considered including multiple non-contact type Kinects and contact type Hydra controllers. This research presents a user-centric VR environment with the interfaces, effectively providing diverse motion sensing capabilities such as jumping and wielding a blade. Also, the experimental action and sports genre games are implemented in combination with the improved interface schemes. Especially, a non-contact interface with multiple Kinects combined with a contact interface, Hydra, improves the motion recognition rate of the VR games. The effectiveness and usability of the proposed interfaces have been assessed according to the empirical evaluation scheme with modified system usability scale (SUS) statements. The main contribution of this research is the enhancement of the system’s usability in terms of user satisfaction resulting from the improved motion recognition rate of the VR games through the suggested immersive motion sensing interfaces.

Key Words: Virtual Reality (VR); Augmented Reality (AR); Head-Mounted Display (HMD); Contact/Non-Contact Motion Sensing Interfaces; Cyberspace; System Usability Scale (SUS).

Reference to this paper should be made as follows: Shin, S., Kim, S., Chang, J., and Park, K. (2014) ' User-Centric HMD-Enabled VR Interface with Improvement Assessment through the System Usability Scale (SUS) Test ', Int. J. Computational Vision and Robotics.

Biographical notes: SangHoon Shin received his M.S. in Ludology from Sangmyung University, Seoul, Korea in 2013. He has worked as a game programmer since 2012. His major interest field of study is virtual reality game and AR game.
Seok-Kyoo Kim has been working as a professor at Sangmyung University, Seoul, Korea since 2014. He received his M.S. and B.S. in Computer Science from Seoul National University, Seoul, Korea, in 1994 and 1992, and the Ph.D. in Computer Science and Engineering from Seoul National University, in 2010. He worked for NC Soft from 1999 to 2003. His major interest fields of study are digital storytelling, HCI and serious games.

Juno Chang has been working as a professor at Sangmyung University, Seoul, Korea since 2003. He received his M.S. and B.S. in Computer Science from Seoul National University, Seoul, Korea, in 1992 and 1990, and the Ph.D. in Computer Science from Seoul National University, in 1998. He worked for i2 Technologies from 1998 to 2003. His major interest fields of study are digital contents, mobile services, and serious games.

Kyungeun Park received her B.S. and M.S. in Computer Science and Statistics from Seoul National University, Seoul, Korea in 1990 and 1992. She received her D.Sc. in Information Technology from Towson University, Towson, Maryland, U.S.A. in 2012 and has been working as a lecturer at Towson University since 2013. She worked for KT and KT Data, Inc. from 1992 to 2006. Her major interest fields of study are context-aware collaboration middleware and semantic data analysis of a multi-sensor environment.

1. Introduction

The success factor of the virtual reality (VR) game lies in making it easier, more intense, and more immersive. Fortunately, the rapid development of various sensor technologies enables us to take advantage of the sensory data and provide the game users with better immersion effect. Brooks defines a virtual reality experience as any in which the user is effectively immersed in a responsive virtual world (Brooks, 1999). Kim et al. (2004) explains the term, believability, as a measurement of the degree of reality of a virtual space. They describe the believability in terms of three elements of virtual environment: immersion, presentation, and interaction (Kim et al., 2004), (Papagiannakis, 2005).

The goal of the VR game is to make users in a cyberspace feel the virtually indirect experiences as real experiences. This can be promoted by using a variety of input/output multimodal devices which increases mutual interaction between users and computers. In addition, the mutual interaction should be made naturally with increased visual detail for better VR effect (Kim et al., 2004). For this, the traditional keyboard and mouse based input schemes need to be replaced by motion sensing interfaces with less interference. Because the traditional interfaces restrict the active spatial space to a desktop environment and limits the source of input for interaction to wired contact interfaces. As a solution to this, the VR game developers started actively introducing wearable, contact type or mobile, non-contact type interface schemes.

In 1960s, Sutherland introduced a paper about the Sketchpad system which presented a new way of man-machine communication method based on line drawing which makes it possible for a man and a computer to converse rapidly through the medium of line drawings (Sutherland, 1963). After then, the first graphics-driven head-mounted display (HMD) was pioneered by Sutherland by presenting a user with a perspective image which changes as he moves in order to create the illusion effect that he is seeing three-dimensional images based on the perspective transformation algorithm and a wide variety of equipment (Sutherland, 1968), (Rolland and Hua, 2005).

Up to the present, a lot of effort has been put into the development of the highly immersive interface to enhance the recognition rate in the motion-controlled video games combined with various motion controllers. Recent technology applied to those motion
controllers makes it possible to identify a variety of different users’ motions like turning head or raising hands.

This research is focusing on increasing usability of an augmented reality game with improved immersive motion sensing interfaces for head-mounted display (HMD) enabled virtual reality (VR) game. For this, two types of sensing interfaces are considered including multiple non-contact type Kinects, developed by Microsoft, and contact type Hydra, developed by Sixense Entertainment. Also, this research presents a user-centric VR environment with the introduced interfaces, effectively providing diverse motion sensing capabilities such as jumping and wielding a blade. The experimental action and sports genre games are implemented in combination with the improved motion sensing interfaces. The action genre game has been developed and interfaced with three multiple Kinects. Whereas, the sports genre game is developed with both contact type interface, Hydra as well as Kinect. The experiment shows that the suggested interfaces improve the motion recognition rate of the VR games. After then, the effectiveness and usability of the proposed interfaces have been assessed according to the empirical evaluation scheme with modified system usability scale (SUS) statements. The SUS test shows positive results that the proposed interface schemes enhance the user satisfaction resulting from the improved motion recognition ratio of the VR games through varied combination of motion sensing interfaces.

This paper is composed of 5 chapters in total. Chapters 1 and 2 present the introduction and the related research. Chapter 3 explains the proposed interfaces including the testbed VR games and user feedback. Chapter 4 introduces the analysis of the SUS test results according to the interface improvement. Chapter 5 concludes with future research direction.

2. Related research

Interfaces for computer games allow users to communicate with the game, help both parties understand each other, and work as an interpreter between them. Bolt proposed a media room environment where voice and gesture inputs at the graphics interface can converge to provide a concerted, natural user modality, as an advanced three-dimensional interface other than traditional two-dimensional interfaces including mouse, keyboard, and touchpad, (Bolt, 1980). Generally, a typical motion sensing interface is divided into contact type and non-contact type interfaces. Contact type interfaces require sensor devices to be attached to a user and work depending on the captured motion information. Whereas, non-contact type interfaces are based on users’ gesture tracked by multiple cameras (Bolt, 1980).

2.1 Contact type interfaces

Contact type interfaces are supported by various motion controllers including WiiMote, PSMove, Hydra, Data Glove, etc. WiiMote (see Figure 1), a game controller of Wii by Nintendo, recognizes users’ motion based on acceleration sensor and tilt sensor in three-dimensional space. Figure 2 shows PlayStation Move motion controller presented by Sony. It is regarded as superior to WiiMote in accuracy. Another variation of stick style interface called Hydra (see Figure 3) allows users to interact with wired PC and is more accurate than the above wireless interface schemes. In addition, wearable gloves, data gloves, equipped with many sensors are widely used for identifying comprehensive controlling motion with hand and fingers (see Figure 4).
The following table compares previously introduced contact type interfaces in terms of available platform, commercialization status, user preference, advantage, and disadvantage.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Glove</td>
<td>PC</td>
<td>No</td>
<td>Low</td>
<td>Various gestures</td>
<td>Inconveniences of wearing</td>
</tr>
<tr>
<td>Hydra</td>
<td>PC</td>
<td>Yes</td>
<td>High</td>
<td>High recognition rate</td>
<td>Wired</td>
</tr>
<tr>
<td>WiiMote</td>
<td>Wii</td>
<td>Yes</td>
<td>High</td>
<td>High availability</td>
<td>Not available with PC</td>
</tr>
<tr>
<td>PSMove</td>
<td>PS3</td>
<td>Yes</td>
<td>High</td>
<td>Interaction using light</td>
<td>Not available with PC</td>
</tr>
</tbody>
</table>

While choosing the experimental contact type interface for this research, PC platform, higher recognition rate, popularity, and user friendliness are considered as the most important selection factors. According to the analysis, Hydra interface is chosen to be an experimental contact type interface.

2.2 Non-contact type interfaces

Non-contact type motion sensing interfaces interact with game system by collecting motion information from multiple cameras. The interfaces which control computers with users’ bodies were selected as 10 emerging technologies in 2011 at MIT. It includes Kinect, iPoint3D, and LeapMotion. The most popular non-contact motion sensing interface, Kinect, is a webcam-style add-on device through which a user can interact with console/computer, e.g., Xbox 360, without contact-type controllers using their gestures and spoken commands. Fraunhofer developed iPoint3D which is equipped with stereo cameras. It allows people to communicate with a 3-D display through simple gestures – without touching it and without 3-D glasses or a data glove and also supports communications between users and systems by
recognizing their finger gestures as soon as a user moves their hands. LeapMotion controller senses how users move their hands and lets them control their computers.

Figure 5 Kinect Controller

Table 2 compares three of non-contact interfaces in terms of available platform, commercialization, preference, advantage, and disadvantage. The requirements for the experimental PC platform including wide recognition range and various gestures from full-body lead us to select Kinect as the non-contact type interface for this research (Shin et al., 2014).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>iPoint3D</td>
<td>PC</td>
<td>No</td>
<td>Low</td>
<td>Various gestures</td>
<td>Low recognition rate</td>
</tr>
<tr>
<td>LeapMotion</td>
<td>MAC</td>
<td>Yes</td>
<td>High</td>
<td>High recognition rate</td>
<td>Narrow range of recognition</td>
</tr>
<tr>
<td>Kinect</td>
<td>PC, Xbox, Mac</td>
<td>Yes</td>
<td>High</td>
<td>Recognized by using full-body</td>
<td>Unable to recognize mounted device</td>
</tr>
</tbody>
</table>

The following two figures show how the player interacts with the game via Hydra and Kinect interfaces (see Figure 6 and Figure 7).

Figure 6 Playing the Games with Hydra

Figure 7 Playing the Games with Kinect
3. Implementation of interfaces

This research introduces the implementation of the VR games where the proposed interface schemes are integrated and the assessment of the effectiveness and usability of the proposed interfaces according the empirical evaluation scheme with modified system usability scale (SUS) statements (Brooke, 1996), (Bangor et al., 2008), (Shin et al., 2014).

3.1 Interface improvements with multiple Kinects and Hydra

To increase the degree of immersion level, it is important to completely identify various kinds of motions. For this, the Kinect based motion sensing interface has been improved by covering occluded skeleton from a specific direction. Many attempts to use multiple Kinect devices have been tried including Wittenberg (2013). Accordingly, this research adopts three Kinects, each covering a 120 degree sector (see Figure 8). The skeletal data from the individual Kinect sensors are merged to produce a more accurate coordinates. In addition, the HMD repeater has been attached on either player’s body or ceiling. Also, the relevance of the motion through Hydra has been increased by attaching one end of Hydra controller on the player’s body. The other end of the Hydra controller is in charge of getting the movement of the user. This configuration has been devised to obtain more accurate skeletal data and enhance the motion recognition rate.

![Multiple Kinects Interface](image)

**Figure 8** Multiple Kinects Interface

Various motions such as attack and invasion are distinguished from each other by analyzing joints’ relative positions of a tracked player with three Kinects. Meanwhile, buttons of Hydra interface are disabled because the button based input functions of Hydra interface are replaced with players’ gestures in this configuration.

3.2 Implementation of VR games

With the advent of the Wii and Kinect, many sports genre games were released and preferred by people but soon after action genre games were rolled out. Accordingly, both types of testbed games are implemented and combined with the proposed interfaces.

As it is important to increase immersion degree for the games, any constraints in getting a user input as well as in giving an output to users should be reduced. The input schemes of the traditional games are keyboard and mouse based. This environment not only causes users spatial constraints by restricting their active space to a desktop environment but also limits the source of input for interaction to wired contact interface. As a solution to this situation, a cyberspace environment has been presented to allow users’ motion such as jumping and raising hands using previously discussed Hydra and Kinect interfaces to work as means like a command to control the system (see Figure 9).
In order to connect Hydra controller, the device is initialized by calling `ControllerSet()` and updated by calling `Update()` as seen in Figure 10. Joystick indicates the transition value of the direction key of the controller. Position converts the location of the controller into 3-dimensional coordinates: x, y, and z. Rotation holds the slope of the controller.

```c
void ControllerSet()
{
    m_Enable = false;
    m_Disabled = false;
    m_Hand = SixenseHands.UNKNOWN;
    m_HandRight = SixenseHands.UNKNOWN;
    m_HandLeft = SixenseHands.UNKNOWN;
    m_Thumbs = 0;
    m_ThumbsPrevious = 0;
    m_Trigger = 0.0f;
    m_JoystickX = 0.0f;
    m_JoystickY = 0.0f;
    m_Position = 0.0f, 0.0f, 0.0f;
    m_Rotation = 0.0f, 0.0f, 0.0f, 1.0f;
}
```

```c
void Update(ref SixensePlugin.sixenseControllerData cd )
{
    m_Docked = (cd.is_docked = 0 );
    m_Hand = ( cd.sixenseHands )cd.which_hand;
    m_ButtonsPrevious = m_Buttons;
    m_Buttons = ( SixenseButtons )cd.buttons;
    m_Trigger = cd.trigger;
    m_JoystickX = cd.joystick_x;
    m_JoystickY = cd.joystick_y;
    m_Position = cd.pos[0], cd.pos[1], cd.pos[2];
    m_Rotation = cd.rot_quat[0], cd.rot_quat[1], cd.rot_quat[2], cd.rot_quat[3];
    if (m_Trigger > TriggerButtonThreshold )
    {
        m_Buttons = SixenseButton.TRIGGER;
    }
}
```

Figure 10 Initialization of Hydra Interface

Kinect interface requires the skeleton of a player to be constructed at the initialization phase by capturing all the coordinates of the constituent bones (See Figure 11).

Figure 11 Skeleton Synchronization based on Kinect Skeleton Structure
Figure 12 shows the initialization code for the Kinect interface.

```csharp
for (int player = 0; player < 1; player++)
{
    if (trackedPlayers[player] > 0)
    {
        for (int bone = 0; bone < int(Kinect.NuiSkeletonPositionIndex.Count; bone++)
        {
            Vector3 bonePos = bonePos[player, bone];
            bonePos[player, bone] = kinectToWorldMultiplyPoint3x4f(,
                kinect.getSkeleton()[SkeletonData(trackedPlayer[player])[SkeletonPositions(bone)];
            bone = kinect.getSkeleton()[SkeletonData(trackedPlayer[player])[SkeletonPositions(bone]);
            Kinect.NuiSkeletonBoneOrientation[] or = -kinect.getBoneOrientation();
                bone = kinect.getSkeleton()[SkeletonData(trackedPlayer[player])[BoneOrientation];
            base.LocalOrientation(bone) = or;
                bone = base.LocalOrientation();
            base.LocalOrientation(bone) = or;
                bone = base.LocalOrientation();
            boneVel[player, bone] = (boneVel[player, bone] - 1) / deltaT;
            boneState[player, bone] = kinect.getSkeleton()[SkeletonData(trackedPlayer[player])[BoneState[bone]);
        }
    }
}
```

**Figure 12** Initialization of Kinect Interface

3.2.1 Sports genre game

As a sports genre game, a race game is implemented by using Hydra and Kinect interfaces. While playing, users are provided with three dimensional live images via HMD of which the following two stream images are shown on both displays. Based on principle of Binocular overlap, this scheme allows both displays to generate different images and to create 3D effect (see Figure 13). In this program, motions like jumping and hand gesture like wielding a blade are recognized through Kinect and Hydra interfaces.

![Figure 13 Race (Sports) Game](image)

To give interactivity and tension to the game, a tiger is chasing an avatar of the player and the avatar runs continuously to avoid the chaser. The chaser is getting faster and faster whenever the player makes mistakes by reducing HP (heart point) of the avatar which affects the speed of the avatar. When the player obtains an item(s), the HP is increased. The game stops when the HP reaches 0. The following code implements the behavioral rules of the characters (See Figure 14).
### 3.2.2 Action genre game

The following figure introduces action genre game based on multiple Kinects and Hydra interfaces (see Figure 15).

![Action Game](image)

By referring to the registered gestures, movement and attack motions are recognized. In addition, avoidance and ignorance actions are available in this program. As Kinect captures coordinates of individual bones or joints, diverse motions are recognized in the program. Also, the refined motions can be handled with Hydra interface by using its buttons and joystick.

### 3.3 User feedback

Both action and sports genre games are integrated with Kinect and Hydra interfaces and reviewed by fifty users. At first, their feedbacks with the initial version of the games are collected. In case of action game, opinions associated with Kinect interface include low-level immersion degree caused by uni-directional recognition of users’ gesture and difficulties of simultaneous manipulation with gesture. Sports genre game also shows the unsatisfactory immersion level caused by the uni-directional recognition of Kinect and irrelevance of the motion to the game due to insufficiently classified gestures.
4. Improved usability through the SUS test

Brooke said that usability is not a quality that exists in any real or absolute sense in his study on System Usability Scale (SUS) (Brooke, 1996). The SUS was originally developed as a “quick and dirty” survey scale that would allow the usability practitioner to quickly and easily assess the usability of a given service (Bangor et al., 2008). The usability of the proposed interface is tested according to the empirical evaluation scheme with modified SUS statements presented by Bangor et al. based on the original SUS statements (Brooke, 1996) (Shin et al., 2014).

![Figure 16 Interface Improvement (Comparison of the SUS test results)](image)

The overall SUS test result of the initial interface is 47 and 53 before and after improvement, respectively (See Figure 16). Over 50 percent of users answered that the initial program was awkward to use and over 70 percent of users assessed that they needed to learn a lot of things to use the program. For the improved interface, over 50 percent of users responded that they would like to use the program frequently and over 60 percent of users answered that the program was easy to use. Both SUS tests shows 6-point increase in the improved interface compared with the initial interface. According to this test result, we can assess that the system’s usability and interactivity have been enhanced in terms of user satisfaction.

5 Conclusion and future work

This research includes the highly immersive interface program and suggests improved user-centric interactive interface for HMD-enabled VR games. Motions such as jumping and wielding a blade are more accurately recognized with Hydra and multiple Kinects interfaces. In order to enhance the motion recognition rate of the interface scheme, three multiple Kinects are arranged and the revision to data capturing method was applied to Hydra interface. The suggested interfaces have been integrated into both action and sports genre games and the effectiveness and usability of the interface have been assessed according to the SUS test. The main contribution of this research is the enhancement of the system’s usability.
in terms of user satisfaction resulting from the improved motion recognition rate of the VR games through the suggested immersive motion sensing interfaces.

The upcoming HMD interface is expected to be lightweight and wireless enough to feel comfortable and maximize immersion degree of the VR games. In case of interface controllers, there exist finer head tracking and remarkable lightening demands. The suggested interface scheme can be improved to elaborate the VR programs by analyzing various motions with the advanced features of newly introduced interface devices. This experience is expected to be applied to wide range of IT frameworks as the innovative means to enhance usability and availability of many smart applications handling various sensor data.

References


LeapMotion, available from: https://www.leapmotion.com/product
