Received May 8, 2019, accepted August 22, 2019, date of publication August 28, 2019, date of current version September 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937937

Enhanced Player Interaction Using Motion Controllers for First-Person Shooting Games in Virtual Reality

PRZEMYSŁAW KROMPIEC AND KYOUNGJU PARK, (Member, IEEE)
School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Kyoungju Park (kjpark@cau.ac.kr)

This work was supported in part by the Mid-Career Researcher Program through an NRF Grant Funded by the MEST under Grant NRF-2016R1A2B4016239, and in part by the Chung-Ang University Research Grants, in 2018.

ABSTRACT The main purpose of virtual reality (VR) is to enhance realism and the player experience. To do this, we focus on VR interaction design methods, analyze the existing interaction solutions including both accurate and rough interaction methods, and propose a new method for creating stable and realistic player interactions in a first-person shooter (FPS) game prototype. In this research, we design and modify the existing mapping methods between physical and virtual worlds, and create interfaces such that physical devices correspond to shooting tools in virtual reality. Moreover, we propose and design prototypes of universal interactions that can be implemented in a simple and straightforward way. Proposed interactions allow the player to perform actions similar to those of real shooting, using both hands such as firing, reloading, attaching and grabbing objects. In addition, we develop a gun template with haptic feedback, and a visual collision guide that can optionally be enabled. Then, we evaluate and compare our methods with the existing solutions. We then use these in a VR FPS game prototype and conduct a user study with participants, and the resulting user study proves that the proposed method is more stable, player-friendly and realistic.

INDEX TERMS Virtual reality, player interfaces, human computer interaction, interaction design, first-person shooting game.

I. INTRODUCTION

Recently, virtual reality-related technologies have expanded very rapidly, leading to new trends and many changes in the industry. Many companies have developed and released their own consumer versions of head-mounted display (HMD) devices, thus bringing VR experiences to players for a fairly low cost. Other companies have focused on developing accessories, such as virtual gloves, motion controllers, player-tracking devices, haptic devices, and other gadgets, which allow the realization of a more immersive VR player experience. Such technologies are revolutionizing the entire gaming industry, allowing developers to create high-end content. First person shooting (FPS) VR is the most favored game mode in VR, because of the immersive environment.

Since the release of the first game, there have been many advances in gaming technologies and platforms. However, player interactions and gameplay have not changed significantly, and in most of cases still employ simple interfaces that facilitate player input with devices such as a mouse, keyboard, or game controller. Recently, many companies have presented input devices such as gloves and motion controllers, and VR FPS titles support two types of player interaction in order to control virtual objects in a game: simplistic player interactions, and precise and accurate player motions in the real world. Simplistic player interactions allow for minimal interactions in FPS, and are limited regarding further interactions. Precise player interactions provide a highly realistic and engaging player interface. However, player interactions often yield unstable and incoherent results, because a player's motion is rough and abrupt, whereas the game system requires accurate and precise actions by a player in the physical world in order to interact with a virtual environment. Therefore, it is necessary to design an interaction that is sufficiently stable that rough actions performed by players can yield the desired interactions,

The associate editor coordinating the review of this article and approving it for publication was Chunming Gao.

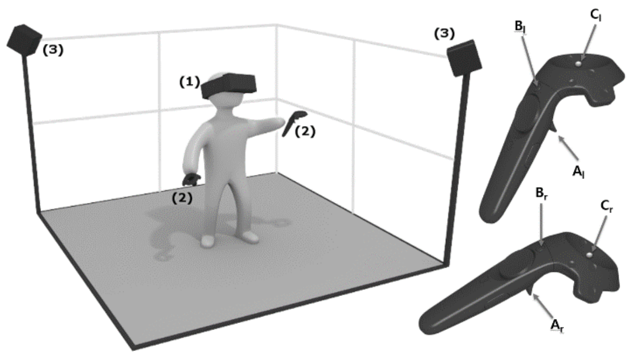

FIGURE 1. Game system environment. A player wears an HMD (1) device and holds motion controllers (2) in each hand. Both the HMD device and motion controllers are tracked by the two sensor stations (3) placed in opposite corners of a rectangle work area (4).

even with discrepancies between the physical and virtual worlds.

However, to the best of our knowledge no methods or solutions considered in previous work have focused on providing a stable two-handed experience, where players can not only feel that they are present in the virtual world, but also feel that they can interact with objects in a more advanced manner. FPS-VR developers tend to copy the standard logic and mechanisms of classical and arcade FPS games directly into VR, and to map rough and abrupt player-motion input data and button events onto specific interactions. Therefore, we propose an improved interaction method that uses both hands, creates stable input data mapping that is used for interaction in VR FPS by exploiting an HMD and controllers, and introduces haptic feedback to create a believable experience and increase player presence and perception.

In this paper, we present a new approach to creating more stable and coherent interactions in VR first-person shooter games. In our system, a player wears an HMD with a set of two motion controllers, which are tracked by two tracking stations positioned in opposite corners of a rectangular work area (Fig. 1). Using these tracking devices, we track a player's head and hand positions and orientations in the work area. The player input information is processed, and mapped onto the camera and virtual motion controllers in VR. After the correspondence between the physical and virtual worlds is established, we design several player-gun related interactions, which are performed using the motion controllers. In addition, we test and evaluate our methods using a VR FPS shooting game demo and a gun template. The system design proposed in this paper can be applied to other VR games and applications that use motion controllers to track a player's hand motion.

The structure of this paper is described as follows. Section II reviews related work on immersive gaming in VR. Section III describes our system setup. In Section IV, we describe our design of the game interfaces. Section V discusses the VR shooting interaction design. In Sections VI and VII, we present our sample game design and discuss our interaction method results, respectively. In Section VIII, we present a player study to evaluate the performance and usability rating of our method. Finally, Section IX concludes this paper.

II. RELATED WORK

A significant amount of work has been carried out in the field of virtual reality and VR games. Thus, we divide the VR game related work into two sections; the immersive player experience research in VR [1]–[16], and the player interface and interaction design in games [17]–[31].

A. IMMERSIVE PLAYER EXPERIENCES IN VR

Player experience in virtual reality depends on many different factors. Among the research results related to this topic, McMahan *et al.* [1] evaluated and proved that there remains a lot to be learned about the effects of increasing a system's fidelity to the physical world. The results of that study show that the levels of interaction and display fidelity can be significant factors in determining performance, presence, engagement, and usability. Another study Tedjokusumo *et al.* [2] shows that FPS VR is the most favored game mode, because virtual-reality environments are more immersive, allowing players to enter a totally different world, where visual effects can be exaggerated to add more value to an environment. Nordvall [3] proved that player experiences in computer interactions are affected more by haptic modalities less than graphical and audial ones. That study shows that poorly understood haptics could become a third modality, which increases interface innovation and accessibility in the process of developing a more immersive experience in games. Another study, by Jannett *et al.* [4], describes the state of immersion as a player experience that is still not clearly defined, and proposes experimental methods to verify whether this experience can be defined quantitatively. A study by Samur [5] examines and compares different techniques used to achieve a digital presence in VR.

Player interface design is a highly important factor in improving overall player experience. In his research [6], LaViola reviews previous technologies and solutions used in designing player interaction interfaces. Furthermore, the author proved that to create rich gaming experiences and player-friendly environments, it is important to find intuitive mappings between a player's physical actions and the game controls. In their book [7], Bowman *et al.* show the importance of player interfaces, which provide a bridge between natural human expression and gameplay. The importance of player interaction interfaces is also described in the research of Katzourin *et al.* [8]. In that study, the authors achieve a compelling game experience by designing a sword and shield interface that uses motion controllers to incorporate the natural movement of human hands. Zenner and Krüger [9] proposed a weight-shifting dynamic passive haptic actuator prototype, which is adopted as a player interface to enhance object perception in virtual reality.

Recently, much research has focused on combining multiple sensory systems provide more coherent object tracking and a more immersive experience. In a study by Kopecky and Winer [10], the authors propose a software system constructed to simplify data from multiple independent tracking systems. Furthermore, the system focuses on tracking latency, and handling calibration errors that lead to situations where different systems report different tracking coordinates. In a similar manner, Kwon *et al.* [11] proposed an approach in which more a realistic and immersive experience can be achieved with the use of less conventional methods, making use of several infra-red and depth tracking sensors. The captured multi-sensor data is combined and simplified into a single skeleton, which can represent synchronized and dynamic player actions in a 360-degree virtual training environment.

A significant body of research also proves that immersive player experience can be achieved through sound. In his research, Lee [12] proposes a system of location-aware speakers. This system leverages Bluetooth technology to connect independent speakers that are positioned in fixed locations, estimates their distance, and distributes the sound. Constructing an audio system consisting of several speakers can be challenging, and therefore in their research Doerr *et al.* [13] make a conventional stereo headset, and develop a system that makes use of the HMD tracking data to distribute sound inside the virtual environment.

The type of content developed also plays an important role in creating an immersive player experience in VR. To enhance player behavior and motivation, Chen *et al.* [14] developed an FPS-VR edutainment system allowing players to perform various tasks within a specified time limit. Chittaro and Fabio [15] presented a serious immersive VR game concept that focuses on educating and preserving knowledge related to safety measures. In addition, the game allows players to experience a serious aircraft emergency, with the goal of surviving it. Another study, presented by Zyda [16], shows that with the right approach developers can create serious highly immersive games that will engage players' emotions on a new level. Zyda also proved that not only graphics and haptics, but also sound and advanced player interfaces play a highly important role in creating immersive games that can engage the minds of players via sensory stimulation and an increased sense of presence.

B. PLAYER INTERFACE AND INTERACTION DESIGN IN GAMES

When designing player interfaces, developers should pay attention to many different aspects of the game in order to create an enjoyable and fun experience. In the early nineties, Csikszentmihalyi defined a euphoric state of flow in which humans are happy, motivated, and fully concentrated and involved in the action they are performing [17]. The author also proved that to achieve this flow a match between one's skills and existing challenges is required, in order to avoid anxiety or apathy if challenges are too simple or difficult. Following the theory proposed by Csikszentmihalyi, much previous research has focused on analyzing and understanding the means in which video games can produce positive effects on players. Subsequently, many researchers have begun to explore ways to utilize the state of flow in video game development [18]–[20].

A highly important and yet often underestimated element in the game development process is the player interface design. The role that an in-game interface plays is to allow the player to communicate with the game by receiving and sending signals back and forth to the game system. In cases when such an interface fails or evokes player frustration, the overall gaming experience might be degraded, ruining the game [21]. Bergman [22] pointed out that although there much research is being conducted in the fields of human-computer interface (HCI) and game design, both fields show a limited awareness of the other's work. Moreover, he proved that even less research focuses on reducing the gap between the two fields. In a different study by Barr *et al.* [23], the authors examine players' in-game motivations, such as entertainment pleasure, emotions, and challenges, and prove that it is important to minimize interaction disruptions between the player and game. An even more recent study shows that the nature of player interactions in video games has still not been clearly defined, because it is a highly complex and difficult topic, and considerable research is still required [24].

Much research has focused on analyzing player interface and interaction design methods and solutions that are used in different game titles. Johnson and Wiles [25] analyzed affective and effective player interface designs in video games. They also suggest that games generating a positive affect are successful when they facilitate the state of flow proposed previously by Csikszentmihalyi [17]. In research by Caroux *et al.* [26], the authors provide a systematic review of empirical evidence for recent concepts in in-game player interactions. The review divides previous research into six sections, which describe in-game engagement, enjoyment, information input, information output, in-game contents, and multiplayer elements, respectively. In addition, the authors discuss practical implications that can help game developers to optimize their in-game player interactions.

Recently, much research has focused on designing better player interfaces in virtual environments. Clergeaud and Guitton [27] proposed a solution for problems involving increased task durations when trying to locate and select objects in virtual environments. The solution is based on increasing a player's natural field of view while using an HMD device. Zielinski *et al.* [28] attempted to create a bridge between the real world and the virtual environment with the use of an interaction technique that allows players to naturally hold a physical object in both hands while manipulating virtual content inside it. In that paper, the authors also proved their hypothesis that the effect of holding a physical box improves the overall player experience and performance. Oliviera *et al.* [29] proposed a design for a haptic guidance technique for virtual environments. By using a specially designed vibrotactile HMD device, the authors render the

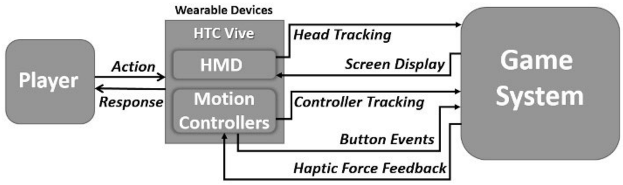

FIGURE 2. A player using wearable devices creates actions, and our system sends responses such as motion controller haptic force feedback or new graphics to the HMD device.

positions of objects in 3D space by varying both the stimulus loci and vibration frequency. A large body of recent research has also focused on topics related to virtual reality walking interfaces. Langbehn *et al.* [30] analyses methods used for redirected walking in VR, and discuss the potential of curved walking. Simeone *et al.* [31] investigates how players react when presented with visual stimuli that compromise their awareness when walking in VR. After analyzing various cases, the authors derived guidelines on how to alter player movement behavior in virtual environments.

III. SYSTEM OVERVIEW

Our system consists of four elements, including wearable devices (HMD, motion controllers), a game system, tracking devices, and a player interacting with the system (Fig. 2). Because we do not focus on developing new hardware, we adopt the recent commercial version of an HMD device that comes with two motion controllers and a tracking system consisting of two base stations. In this system, a player using the HMD and motion controllers provides all required the input data:

- HMD position and rotation tracking,
- Motion controller position and rotation tracking,
- Motion controller trigger, shoulder, and grip button events.

IV. INTERFACE DESIGN

In our system, we map the physical locations and orientations of the HMD and the two motion controllers onto virtual locations in our game system. To achieve this, we assign the sensory center of the HMD and those of the two motion controllers to the model center of the in-game player camera and two virtual mesh objects, respectively. Then, we determine the translation and rotation of each of the three objects in the real world as a player moves and changes their viewing direction, and directly apply these movements to the virtual camera and mesh objects. Here, each motion controller is defined in virtual space as a mesh with its transformation center located at a pivot, with its orientation described by two vectors: a forward vector and an up vector. To enhance the visual coherence and player perception, we model mesh objects that have the same shapes and sizes as the motion controllers in the real world. This allows us to establish correspondences between the physical motion of a player and the virtual motion of a mesh object, and therefore the set of the player's motions can be interpreted as event triggers in the virtual reality.

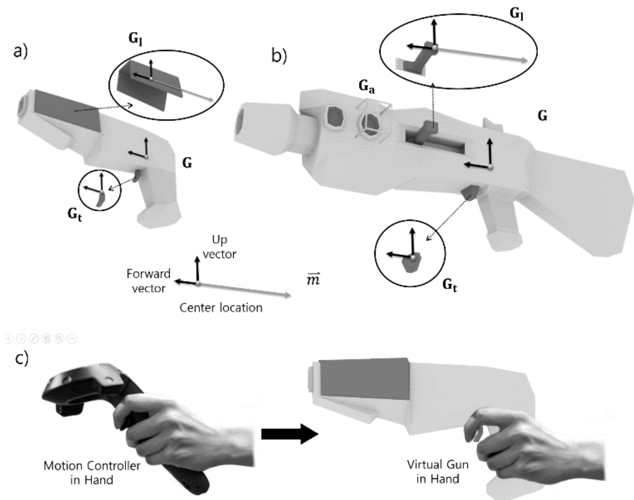

FIGURE 3. Proposed gun interface solutions, consisting of three mesh objects and an additional attachment: (a) pistol interface, (b) machine-gun interface, and (c) correspondence of the motion controller with a virtual gun interface.

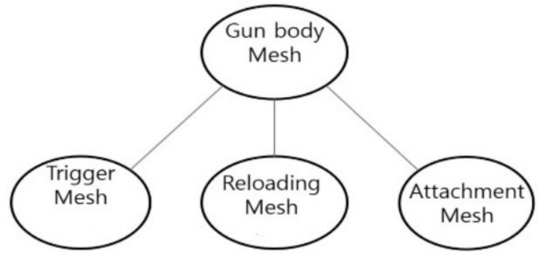

FIGURE 4. Gun interface hierarchical design.

For direct interfaces between the hand motions of the player and the actions of virtual devices performed in FPS (Fig. 3), we design two kinds of guns, a pistol-like model and a submachine gun-like model, and align them with the virtual controllers in such a way that the grip, trigger, and barrel of each model in the virtual reality match with the grip, A button, and sensory head of a controller in the physical world, respectively. In this manner, players can interact with a virtual gun as soon as they grab the motion controller in their hand. Moreover, both of these guns are articulated objects, and their models are composed of four subcomponents: a trigger mesh, a lever mesh, an attachment mesh, and the gun body mesh.

These mesh objects are described by their pivot center coordinates, and undergo transformations according to the model hierarchy as a player performs various actions, as shown in Fig. 4. The main gun body G is the root of the hierarchy, which is located and oriented relative to the corresponding physical motion controller. A trigger G_t connected to the gun body can rotate locally when the player presses the A button on the motion controller in the real world as shown in Fig. 1. A lever G_l, which is also connected to the gun body, is able to translate the lever mesh along the vector in

TABLE 1. Interaction table.

Action Name	Player Interaction	Haptic Feedback
Object Pickup/Drop	Press the A and B buttons while the VR controller is overlapping or colliding with other objects	Single light vibration on pickup and drop
Gun Fire	Press the A button on the controller while holding gun objects	Single strong (small gun) or continuous strong (machine gun) vibration
Gun Reload	While holding the gun with one controller, press the A button on the other controller, and move the lever mesh along the specified reloading direction on the held gun object	six single tap vibrations are mapped onto a short path on the reload axis
Object Attachment	Move the other controller holding ammunition close to the attachment object for the held gun, and press the A button	Single light vibration on attachment

the reloading direction when the player performs a reloading action. Finally, an there is an attachment G_a at the location where players can attach an additional new mesh object, such as ammunition, to the gun body G.

V. INTERACTION DESIGN

We apply our designed interfaces described in Section IV to develop stable player interactions for VR-FPS, so that a player's actions, like actions in the real world, trigger the desired corresponding events. To achieve stable interactions, we provide visual player guidelines, such as a wire frame of a bounding box in the collision area, as well as tactile feedback. Table 1 summarizes our desired actions, player interactions, and the haptic feedback mechanisms signaling successful interactions.

A. PICKING UP

Any object we define can be picked up using either of the motion controllers. Because the gun and other player-defined mesh objects can have complex polygon structures, we use a pre-defined low polygon mesh for each model to detect collisions in a reduced computational time. If the system detects the collision of the motion controller mesh with another mesh object, then the player can pick up or grab that object by simply pressing the A button on the colliding motion controller. The object is then attached to the colliding virtual motion controller C.

In this process, the position of the gun mesh using G is set to exactly match the coordinates of the pivot of the motion controller C, and we apply the orientation of the motion controller to the gun mesh to align the forward and up vectors. We also hide the virtual motion controller mesh when the player holds a mesh object, to achieve more pleasant visual results (Fig. 5a). On the other hand, if the system detects a collision with a non-gun mesh and the player presses

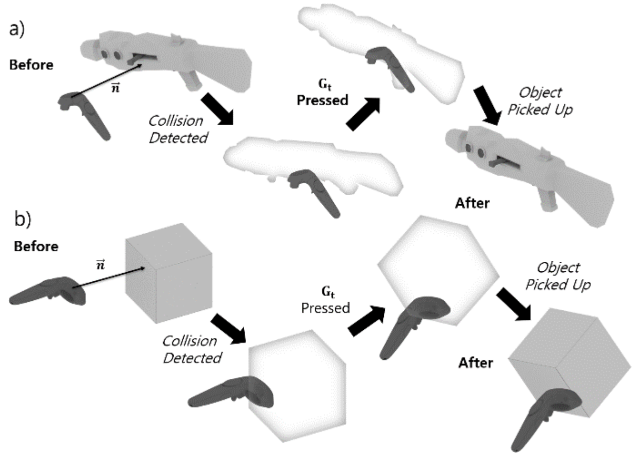

FIGURE 5. Pick-up interaction design for gun objects (a) and non-gun objects (b).

the A button on the motion controller, then that mesh is attached to the colliding virtual motion controller C, but the position of the object's mesh remains unchanged, and only the pivot is transformed to match the transformation of the virtual motion controller (Fig. 5b).

B. FIRING

To fire a gun, the player should press the A button on the motion controller with the gun picked up. During firing, the angle of the A button on the motion controller changes the angle of the trigger button G_t on the in-game mesh. The motion controller Application Program Interface (API) returns a normalized scalar value k, which ranges from 0 to 1. This value corresponds to how far the button is pressed in, where 0 indicates no pressing and 1 means fully pressed in. We use this k value to rotate the gun trigger G_t between 0 and 45 degrees around the y axis, with use of the pitch rotation matrix

$$R = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad (1)$$

where the angle $\alpha = 45°k$ is the size of the pitch rotation to be performed about the y axis (Fig. 6).

C. RELOADING

Reloading is a little complicated, because it requires the player to use both hands. One holds a gun object, and the other performs the reloading action. First, the player should move the motion controller of the free hand sufficiently close to the gun lever mesh that both the motion controller C and the lever mesh G_l are within a certain threshold distance. Secondly, the player should press the A button on the free-hand motion controller, and move it along the expected direction of the reload vector $\vec{m}$ while holding the motion controller (Fig. 7).

Low-precision player movements or other tracking data errors, including occlusion, might create highly abrupt and

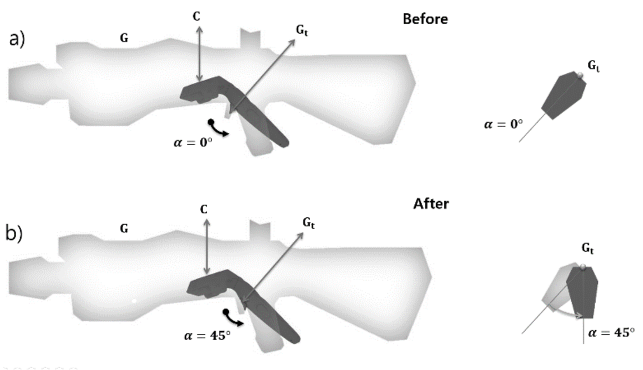

FIGURE 6. Trigger mesh rotation example: (a) mesh in initial position, where $\alpha = 0°$ and (b) maximal rotation caused by full A button press where $\alpha = 45°$.

rough input signals that can lead to incoherent results. We approach this problem in a straightforward manner, by applying the standard vector projection formula

$$\text{proj}_{\vec{n}}\,\vec{m} = \frac{\vec{n} \cdot \vec{m}}{\|\vec{m}\|} \tag{2}$$

where $\vec{n}$ is the actual motion direction, $\vec{m}$ is the expected direction, and the projection of $\vec{n}$ onto $\vec{m}$ is equal to the quotient of the inner product of $\vec{n}$ and $\vec{m}$ by the magnitude of $\vec{m}$. Using this formula, we perform two tasks simultaneously: checking the motion controller movement direction and moving the actual lever mesh along the reload direction. If the projection result is a negative number, then the angle between the two vectors is obtuse, and so the player is moving in the negative direction of the reload axis. On the other hand, if the value is positive then the angle is acute, and so the player's hand is moving along the positive direction of the reload vector. Furthermore, we use the scalar value of the projection result as a measure of how far the lever mesh should be translated along the reload axis defined by the vector $\vec{m}$ from the lever pivot G_l (Fig. 7).

To prove that our method is more stable and produces coherent results in comparison with previously proposed methods, we apply a statistical analysis, such as the simple line regression method. Using the linear regression method, we can prove that the controller movement data is distributed around a straight line. This allows us to prove that the further the motion controller moves from the starting position C_{t0} (Fig. 8a), the more errors accumulate in the data. However, because we use vector projection in our method, this error accumulation does not lead to instability (Fig. 8b).

D. DROPPING

Objects that a player picks up can be dropped onto the ground by releasing the A button on the motion controller. An exception to this rule is gun objects, which require the trigger button for a firing action. For this reason, we use a B button-press event to drop a gun G and disconnect it from the virtual motion controller C.

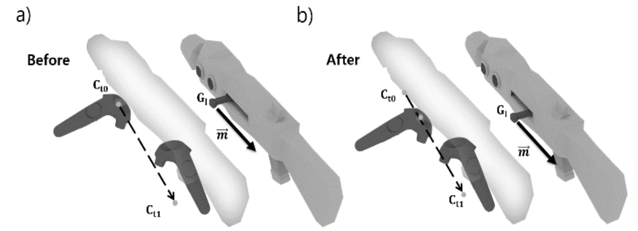

FIGURE 7. Reload interaction design presented for the machine gun example.

E. ATTACHMENT

In our system, we also develop an additional interaction in which the player can use special ammunition with the machine gun. Picking up this ammunition is performed in the same manner as picking up the gun, by pressing the A button on a motion controller while the virtual motion controller mesh C is in collision with or penetrating the attachment object O. To attach such objects to the gun G, the player must first move the held object inside the collision box G_a that is located on the gun. If O is colliding with or penetrating G_a, then the player can attach it to the gun by releasing the A button that was previously pressed. At that moment, the system changes O's parent object to G, and sets its transformation matrix to match G_a. This allows the alignment of O to match the location and rotation of G_a. We exploit this by employing the following algorithm:

① While O is attached to C_1 and G is attached to C_2, then (2). Otherwise, go to (5).
② If O and G_a are colliding so that Collision(O, G_a) equals true, then go to (4), otherwise repeat (1).
③ If Pressed(A_1) equals false then set attach O to G, set O transformation to G_a, and move to (5), otherwise repeat (2).
④ If Pressed(A_1) equals false, then detach O from C_1.
⑤ Stop if O is detached from C_1 or G is detached from G_2.

VI. GAME PROTOTYPE

A. CONCEPT

To test our methods, we developed a first-person shooter game demo. The game is very similar to those of old arcade games: the player has to destroy all of the enemy AI space ships by shooting down and destroying them using a space gun that the player holds in one hand. Players lose their health points at the moment when they are hit by enemy missiles, and the health bar is reduced by a specific amount corresponding to the sustained damage. Therefore, in order to stay alive players must avoid being hit by missiles that move toward them.

B. DESIGN

We created three separate enemy AI space ships, with a design similar to a classic retro arcade game with space invader-like AI enemies. In the arcade game, the player was represented by a small spaceship. Therefore, we created a gun

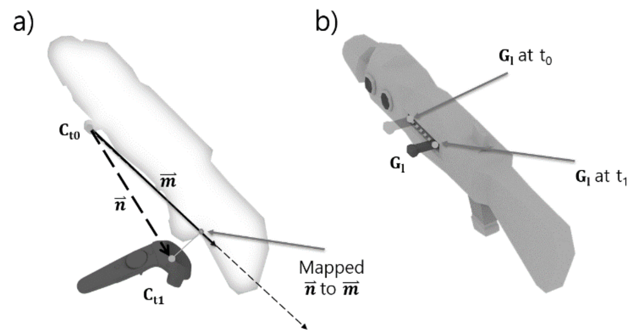

FIGURE 8. Motion controller movement projection in the expected reload vector direction $\vec{m}$ while performing a reloading action (a), and actual lever mesh translation caused by an action performed by the player (b).

FIGURE 9. Game prototype screen capture.

FIGURE 10. Example results for our method, for a reload action performed in the real world using two motion controllers (left image), and the result of this action in virtual reality (right image).

with a design like that of the player's spaceship. In addition, we split the game's User Interface (UI) into two parts: a 3D game menu and 2D player interface. The game menu is visible to the player prior to the gameplay, and the player can choose the menu elements by shooting at them. Among the menu options are Play, Ranking, and Exit buttons, which are selected when the player shoots at them. The 2D player interface is visible in the game mode. This presents the current player status information, including health, ammunition, enemy wave number, and player score (Fig. 9).

C. ENVIRONMENT

This demo game was tested on a PC with an HMD device that comes with two controllers and two base sensor stations. In addition, we projected the game screen onto a white wall, so that it would broadcast the player's gameplay to an audience.

VII. RESULTS

In this paper, we propose a new and more stable approach to designing player interactions in VR FPS games. To evaluate our methods, we developed a gun-template interface with interactions, and we applied our gun-template system in our VR FPS demo game, as explained in the previous section. The gun template is designed such that developers are able to modify and adjust it freely, under the condition that every gun consists of three mesh objects used for shooting, reloading, and grabbing, and a collision box required for attachments. Both the gun template and the VR FPS demo have been developed using a free game engine. Moreover, we have decided to adopt a recent HMD device with the two motion controllers and two tracking stations, which are provided in one set, and can be used with the trending game engine (Fig. 10).

We compared the results of our methods with the two types of VR FPS game: a simple and a highly realistic VR FPS games.

A. RELOADING

Real gun interactions, such as reloading, require many detailed actions that must be performed using both hands. Reloading in recent VR FPS games is achieved in two different ways: simple controller shacking, and complex actions that require precision and player focus. Figure 11 presents a comparison of our methods with others.

The first example presents the simplest approach used in games (Fig.11a). This approach requires the player to shake or rapidly move the controller holding a gun towards the bottom. The vector $\vec{m}$ indicates the expected direction of the motion controller. The movement threshold for the player in this example is very high and therefore allows a higher error of movement. Although this method is very simple it causes fatigue and does not present engaging and realistic interaction. The second example of gun reloading (Fig.11b) presents a very detailed and realistic approach. The player must move the motion controller which reloads the gun within a very small threshold range with the actual motion vector $\vec{n}$ almost identical to the expected movement direction $\vec{m}$. In this case the approach is simulates the gun behavior in a very realistic way but due to the required movement precision it is very difficult to achieve by the player. Lastly the figure 11c presents our method which combines realistic gun behavior with the approach taken in the first example where the threshold is very high and allows the player abrupt and clumsy movements. As mentioned in the previous sections our method requires from the player to move the motion controller towards an expected direction described by the $\vec{m}$ vector and at the same time it allows a higher error in

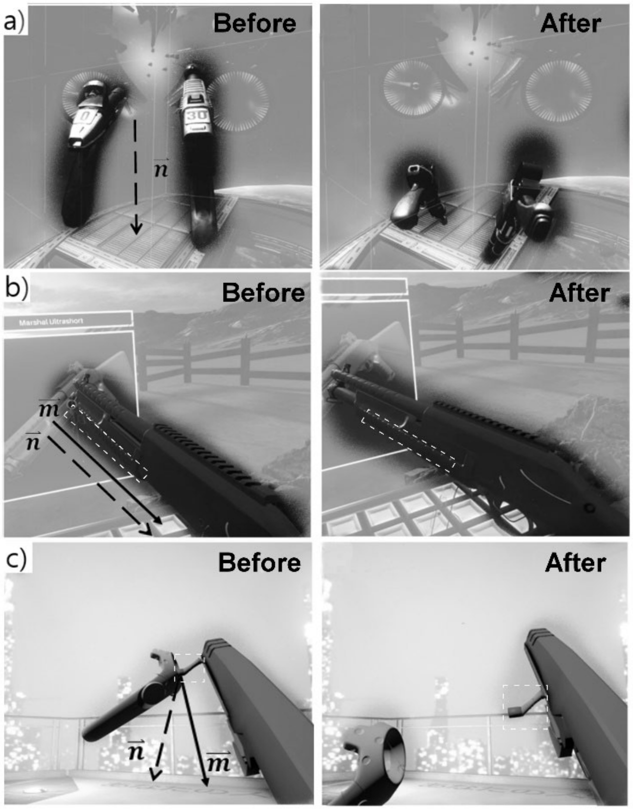

FIGURE 11. Gun reload interaction comparison of three different games: shooting game with a simple solution approach (a), shooting game with interactions designed in a highly complex and realistic manner (b), and our demo game that applies our proposed methods with a stable interaction solution (c).

player movement with use of a higher threshold in compare to the second method. This approach does not cause high fatigue, it mimics realistic gun behavior to a certain level that allows to create an engaging and appealing VR gun interaction.

B. ATTACHING

In games in which reloading is performed using simple actions such as a single button press or shaking of the motion controller, there is no interface available for attaching objects, such as ammunition, to a gun. Therefore, we only compare our demo with the second group of games, which define realistic gun interactions (Fig. 12). Interaction designed in these games does not allow abrupt motion and the acceptable thresholds are very low, therefore performing specific actions requires a lot of patience and players movement precision (Fig.12b). In the third stage of the action the player should match the rotation angle of the object that should be attached to the gun and then has to slide it in very precisely. Such an approach is not appropriate for low precision hardware devices since there is no haptic feedback and the player perception is limited and depends only on the visual feed from the HMD device. Hence, players must repeat this step until they can perfectly match the rotation and movement. In order to solve the problem and reduce the amount of repetitions needed to successfully execute this action we propose a more simplified approach, where the object when in range is automatically guided or attached by the system without the need of manually trying to match the rotation and angle attaching object (Fig. 12a). The method we propose differs from the previous method at the third stage in which it only requires from the user to release the object held in the hand with a simple button release event.

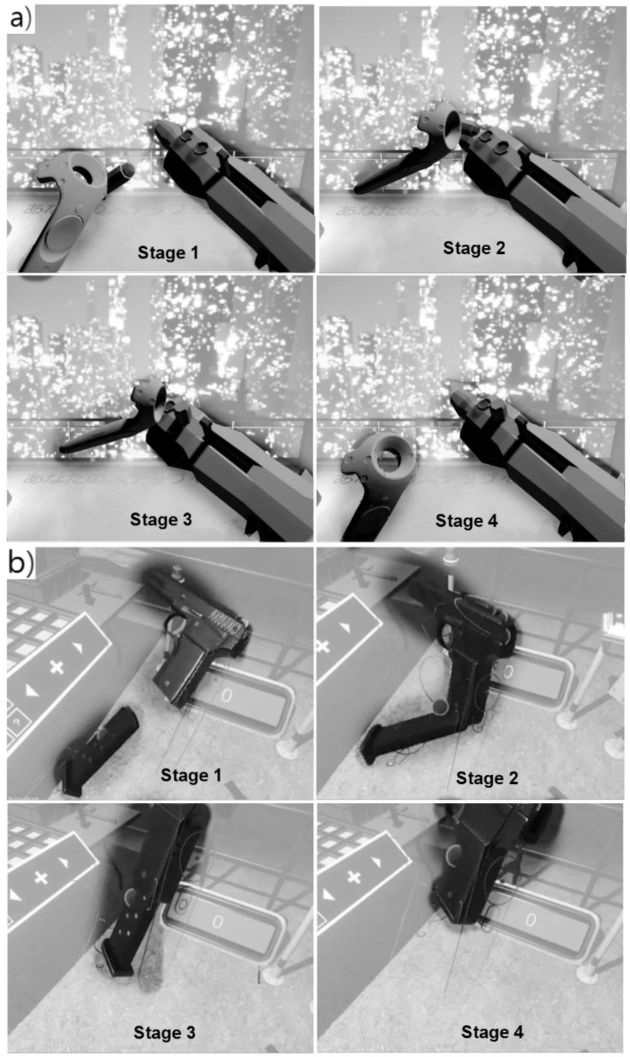

FIGURE 12. Gun attachment interaction comparison: our proposed method (a) and a complex and realistic approach for player interaction (b).

Other interaction solutions, such as firing, picking up, and dropping, create similar results to those of previous game designs, and only their implementations may differ in some way, because of the different platforms and hardware they run on.

VIII. PLAYER STUDY

We conducted a player study to evaluate our proposed methods and provide a usability rating for all three types of player interaction designs in VR: the simple interaction method G1, complex interaction method G2, and our method G3.

TABLE 2. Results of questionnaires showing participant ratings on a five-point likert scale.

Question		Frequencies										Median	Standard deviation
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
Q1: It felt natural to use	G1	3	2	4	5	5	3	5	3	3	4	3.7	1.05935
	G2	3	3	3	4	3	1	3	3	4	4	3.1	0.875595
	G3	4	5	4	5	4	4	5	4	4	4	4.3	0.483046
Q2: I could interact precisely	G1	2	2	4	5	3	3	5	4	4	4	3.6	1.074968
	G2	2	2	3	4	4	3	3	2	3	3	2.9	0.737865
	G3	5	5	4	5	4	4	5	5	5	4	4.6	0.516398
Q3: It was easy to use	G1	2	3	3	5	4	2	5	3	4	5	3.6	1.173788
	G2	3	2	2	5	3	2	4	2	2	3	2.8	1.032796
	G3	4	5	3	5	4	5	5	5	5	5	4.6	0.699206
Q4: It was stable when using	G1	2	4	4	5	2	4	4	5	3	1	3.8	1.135292
	G2	2	2	4	3	2	3	3	2	3	2	2.6	0.699206
	G3	5	5	4	5	3	5	5	5	5	4	4.6	0.699206
Q5: It was fun to use	G1	3	3	4	5	5		5	3	3	4	3.8	0.918937
	G2	3	4	3	5	5	4	5	4	3	3	3.9	0.875595
	G3	5	3	4	5	5	4	5	4	4	5	4.4	0.699206
Q6: I felt tired using it	G1	4	5	2	3	4	5	3	5	4	4	3.9	0.994429
	G2	2	4	2	2	3	4	1	2	2	1	2.3	1.05935
	G3	2	1	1	1	2	1	1	1	1	2	1.3	0.483046
Q7: I felt frustrated using it	G1	3	4	2	2	1	3	1	4	4	2	2.6	1.173788
	G2	3	5	2	1	3	4	3	4	4	1	3.2	1.135292
	G3	1	1	2	1	2	1	1	1	1	1	1.2	0.421637
Q8: I felt haptic feedback	G1	2	2	3	3	5	1	5	2	3	2	2.8	1.316561
	G2	3	3	3	4	4	2	5	4	3	3	3.4	0.843274
	G3	5	5	4	5	5	5	5	5	5	5	4.9	0.316228

Ten volunteers participated in the study (P1-P10): five males and five females. The average age of our participants is 27 years, and their average knowledge and experience in the field of virtual reality is on a moderate level. The study is divided into two parts: a performance evaluation and usability rating. For the performance evaluation, we evaluated the times that the participants required to perform actions such as firing, reloading, attaching a component, and grabbing the gun mesh using their virtual hands. For the usability rating study, we prepared a questionnaire for the participants to rate several factors of each game interaction.

Before the study, each participant completed a demographic questionnaire and wore the HMD. A researcher verified that the HMD was mounted correctly on the participant's head, and that there were no problems related to the eye lens focus and sharpness. Then, the researcher adjusted the HMD correctly to ensure the best experience and avoid unnecessary distractions and discomfort. Next, each participant received instructions from a researcher on how each game interaction should be performed. Later, each participant trained their skills to become accustomed to the interaction methods G1, G2, and G3. The actual player study was conducted from the moment that the participant stated to a researcher that they were ready.

We conducted the performance evaluation of the G1, G2, and G3 methods with regard to the fire, grab, attach, and reload actions. Table 3 presents the comparison results, where the time is measured in seconds from the moment the participant starts until the moment a specific action is finished. Table 3 shows that our method G3 allows the participant to perform in-game actions in a less time-consuming manner than G2. The performance of G1 is the fastest of the three methods. However, G1 does not provide actions such as grab and attach, and does not support two-handed interactions.

TABLE 3. Performance evaluation in seconds.

Action Name	G1	G2	G3
Fire	Less than 1 sec	Less than 1 sec	Less than 1 sec
Grab	N/A	3.317 sec	1.147 sec
Attach	N/A	8.401 sec	1.676 sec
Reload	0.670714 sec	2.151 sec	1.362 sec

In the usability ratings study, we used a Wilcoxon signed rank test to compare the usability ratings of each interaction design. Table 2 uses five-point Likert scale, and numerically displays the frequency distributions for each method, question, and participant P1-P10. For each question Q1-Q8, we computed the median and standard deviation. The player study demonstrated that our method G3 outperforms the other two methods G1 and G2 in every usability aspect in the questionnaire. For the questions Q1, Q2, Q3, and Q4, participants ranked the G2 method with the lowest rating, the G1 method with a middle rating, and our method with the highest among the three. The fifth question Q5 shows that participants preferred complex interaction designs to the simple one-handed method G1. The results for Q6 show that our method made players less tired than the other two methods.

From Q7, we can see that the complex method G2 causes frustration, and the participants rated this the lowest. The results of the last question Q8 prove that our haptic feedback design was rated the highest.

IX. CONCLUSION

In this research, we developed a novel interaction method for FPS in VR. We enhanced player presence in VR by designing two-handed interactions with guns. We mimic realistic behavior, provide vibration feedback for certain actions, and reduce the number of keys to be pressed in old-fashioned first-person shooter games. Using this approach, we provide stable results for player interactions. We believe that with this research, we can encourage developers to create more realistic and believable interactions and interfaces for games and other interactive media in VR.

ACKNOWLEDGMENT

The authors would like to thank the Futuretown Company for giving us permission to use their "A-10 VR" game screenshots. They would also like to thank to RUST, LLC, for permission to use screenshots of the game "Hot Dogs, Horseshoes, and Hand Grenades."

REFERENCES

[1] R. P. McMahan, D. A. Bowman, D. J. Zielinski, and R. B. Brady, "Evaluating display fidelity and interaction fidelity in a virtual reality game," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 4, pp. 626–633, Apr. 2012.

[2] J. Tedjokusumo, S. Z. Zhou, and S. Winkler, "Immersive multiplayer games with tangible and physical interaction," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 1, pp. 147–157, Jan. 2010.

[3] M. Nordvall, "The sightlence game: Designing a haptic computer game interface," in *Proc. DiGRA Int. Conf.*, vol. 7, 2013.

[4] C. Jennett, A. L. Cox, P. Cairns, S. Dhoparee, A. Epps, T. Tijs, and A. Walton, "Measuring and defining the experience of immersion in games," *Int. J. Hum.-Comput. Stud.*, vol. 66, no. 9, pp. 641–661, Sep. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581908000499

[5] S. X. Samur, "Comparing stage presence and virtual reality presence," *Revista Bras. Estud. Presença*, vol. 6, no. 2, pp. 242–265, Aug. 2016. [Online]. Available: http://dx.doi.org/10.1590/2237-266058902

[6] J. J. LaViola, Jr., "Bringing VR and spatial 3D interaction to the masses through video games," *IEEE Comput. Graph. Appl.*, vol. 28, no. 5, pp. 10–15, Sep./Oct. 2008.

[7] D. A. Bowman, E. Kruijff, J. J. LaViola, Jr., I. P. Poupyrev, *3D User Interfaces: Theory and Practice*. Boston, MA, USA: Longman 2004, ch. 1, pp. 1–9.

[8] M. Katzourin, D. Ignatoff, L. Quirk, J. J. Laviola, and O. C. Jenkins, "Swordplay: Innovating game development through VR," *IEEE Comput. Graph. Appl.*, vol. 26, no. 6, pp. 15–19, Nov./Dec. 2006.

[9] A. Zenner and A. Krüger, "Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 4, pp. 1285–1294, Apr. 2017.

[10] K. E. Kopecky and E. H. Winer, "MetaTracker: Unifying and abstracting 3-D motion tracking data from multiple heterogenous hardware systems," *IEEE Access*, vol. 4, pp. 189–203, 2017.

[11] B. Kwon, J. Kim, K. Lee, Y. K. Lee, S. Park, and S. Lee, "Implementation of a virtual training simulator based on 360° multi-view human action recognition," *IEEE Access*, vol. 5, pp. 12496–12511, 2017.

[12] C. H. Lee, "Location-aware speakers for the virtual reality environments," *IEEE Access*, vol. 5, pp. 2636–2640, 2017.

[13] K.-U. Doerr, H. Rademacher, S. Huesgen, and W. Kubbat, "Evaluation of a low-cost 3D sound system for immersive virtual reality training systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 2, pp. 204–212, Apr. 2017.

[14] T.-C. Chen, C.-F. Chiu, A. Klimenko, and T. K. Shih, "Toward a Holodeck like Edutainment game using wearable device and motion sensors," in *Proc. UMEDIA*, Aug. 2015, pp. 242–247.

[15] L. Chittaro and F. Buttussi, "Assessing knowledge retention of an immersive serious game vs. a traditional education method in aviation safety," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 4, pp. 529–538, Apr. 2015.

[16] M. Zyda, "From visual simulation to virtual reality to games," *Computer*, vol. 38, no. 9, pp. 25–32, Sep. 2005.

[17] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York, NY, USA: Harper & Row, 1990, ch. 1, pp. 1–18.

[18] R. Pausch, R. Gold, T. Skelly, and D. Thiel, "What HCI designers can learn from video game designers," in *Proc. Conf. Companion Hum. Factors Comput. Syst.*, Apr. 1994, pp. 177–178.

[19] R. W. Picard, "M.I.T media laboratory affective computing," MIT, Cambridge, MA, USA, Tech. Rep. 321, Sep. 1997. [Online]. Available: http://affect.media.mit.edu/pdfs/95.picard.pdf

[20] S. W. Draper, "Analysing fun as a candidate software requirement," *Pers. Technol.*, vol. 3, no. 3, pp. 117–122, Sep. 1999.

[21] R. Rouse, III, *Game Design: Theory and Practice*, 2nd ed. Plano, TX, USA: Wordware, 2005, ch. 1, pp. 1–19.

[22] E. Bergman, *Information Appliances and Beyond: Interaction Design for Consumer Products*. San Francisco, CA, USA: Morgan Kaufmann, 2000, ch. 10, pp. 300–304.

[23] P. Barr, J. Noble, and R. Biddle, "Video game values: Human-computer interaction and games," *Interact. Comput.*, vol. 19, no. 2, pp. 180–195, 2007.

[24] R. J. Pagulayan, K. Keeker, D. Wixon, R. L. Romero, and T. Fuller, *Player-Centered Design in Games*. Hillsdale, MI, USA: Lawrence Erlbaum Associates, 2002. [Online]. Available: http://www.studiosplayerresearch.com/HCI_Handbook_Chapter.pdf

[25] D. Johnson and J. Wiles, "Effective affective user interface design in games," *Ergonomics*, vol. 46, nos. 13–14, pp. 1332–1345, Oct. 2003.

[26] L. Caroux, K. Isbister, L. Le Bigot, and N. Vibert, "Player–video game interaction: A systematic review of current concepts," *Comput. Human Behav.*, vol. 48, pp. 366–381, Jul. 2015.

[27] D. Clergeaud and P. Guitton, "Pano: Design and evaluation of a 360° through-the-lens technique," in *Proc. 3DUI*, Mar. 2017, pp. 2–11.

[28] D. J. Zielinski, D. Nankivil, and R. Kopper, "Specimen box: A tangible interaction technique for world-fixed virtual reality displays," in *Proc. 3DUI*, Mar. 2017, pp. 50–58.

[29] V. A. de J. Oliveira, L. Brayda, L. Nedel, and A. Maciel, "Designing a vibrotactile head-mounted display for spatial awareness in 3D spaces," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 4, pp. 1409–1417, Apr. 2017.

[30] E. Langbehn, P. Lubos, G. Bruder, and F. Steinicke, "Bending the curve: Sensitivity to bending of curved paths and application in room-scale VR," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 4, pp. 1389–1398, Apr. 2017.

[31] A. L. Simone, I. Mavridou, and W. Powell, "Altering player movement behavior in virtual environments," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 4, pp. 97–106, Apr. 2017.

PRZEMYSŁAW KROMPIEC received the B.S. degree in computer science from the Opole University of Technology, Poland, in 2008, the M.S. degree in digital media content design from Ulsan University, in 2012, and the Ph.D. degree in computer graphics and virtual reality from Chung-Ang University, Seoul, South Korea, in 2018, where he is currently a Researcher. His research interests include non-photo realistic rendering, virtual reality, volume rendering, and game design.

KYOUNGJU PARK received the B.E. degree in computer engineering from Ewha Woman's University, in 1997, and the M.S. and Ph.D. degrees in computer and information science from the University of Pennsylvania, in 2000 and 2005, respectively. After receiving her Ph.D., she was with Rutgers University, as a Research Professor, and with Samsung Electronics, as a Senior Engineer. In 2007, she joined Chung-Ang University, Seoul, South Korea, as a Faculty Member. Her research interests include virtual reality, and computer graphics and interaction.