FingerTouch: Touch Interaction Using a Fingernail-Mounted Sensor on a Head-Mounted Display for Augmented Reality

JU YOUNG OH1,2, JI-HYUNG PARK2, (Member, IEEE), AND JUNG-MIN PARK1,2, (Member, IEEE)
1NT-IT (HCI and Robotics), Korea University of Science and Technology, Daejeon 34113, South Korea
2Center for Intelligent and Interactive Robotics, Korea Institute of Science and Technology, Seoul 02792, South Korea
Corresponding author: Jung-Min Park (pjm@kist.re.kr)

This work was supported in part by the Global Frontier Research and Development Program on Human-Centered Interaction for Coexistence funded by the Korean Government (MSIP) under Grant NRF-M1AXA003-2011-0031380, and in part by the National Research Foundation of Korea grant funded by the Korea Institute of Science and Technology (KIST) Institutional Program under Project 2E30280.

ABSTRACT This study proposes FingerTouch, a method of touch interaction using a head-mounted display for mobile augmented reality. FingerTouch allows users to manipulate virtual content with one-finger touch interaction regardless of the material or tilt of the plane the finger is touching. In addition, users can interact freely with virtual contents using FingerTouch. As the prototype developed in this study uses only one inertial measurement unit sensor attached to a fingernail, it features high mobility and allows users to feel natural tactile feedback, which is important for performing everyday tasks. The user evaluation of FingerTouch indicated that it provides high accuracy, with an average cursor navigation error of 2.15 mm and an average finger gesture recognition accuracy of 95% across 22 participants, two surface orientations, and three surface materials.

INDEX TERMS Augmented reality, finger gesture, inertial sensor, nail-mounted sensor, touch interaction.

I. INTRODUCTION

Augmented reality (AR) is an innovative technology in which real-world objects are enhanced using interactive virtual elements [1]. Head mounted displays (HMDs) (e.g., Microsoft Hololens [2]) are among the types of devices that can be used to experience AR. A typical HMD has the following features. (1) It allows the wearer to see virtual objects while performing real tasks. (2) It ensures that only the wearer will see the virtual objects it displays. (3) It can be used in mobile environments. Unfortunately, the forms of interaction that are ideal for exploiting the features of AR HMDs remain unclear.

Several researchers have proposed guidelines for convenient interaction with virtual objects displayed on AR HMDs. As an AR HMD allows the wearer to see both real and virtual objects simultaneously, its interface should facilitate interaction with both the virtual objects displayed on the HMD and the real-world objects existing in the external environment. Morris et al. [3] argued that an always-available input is necessary for mobile computing. However, there are several requirements for an always-available input [3]. For example, the input device should be portable and it should be possible to perform real and virtual tasks simultaneously. In addition, as an AR HMD allows only the wearer to see virtual objects, users require socially acceptable gestures to interact with the virtual objects visualized on AR HMDs when in public places [4]. The results of a user study conducted by Tung et al. [4] showed that users who wore AR HMDs did not prefer to use handheld devices but rather preferred finger-based gestures performed under the face while in public places. Among the guidelines proposed by Hsieh et al. [5] for socially adaptable interactions, input devices should be usable independently of HMDs and present an interaction method that employs relative pointing for easy adaptation and use in various postures.
There are limitations to the interactions that can be achieved using commercial AR HMDs. Interactions involving hand movement recognition by a camera built into an HMD are easily identifiable because the hand must be lifted over the face, but this practice increases arm fatigue with prolonged usage. Voice interaction is difficult to use in public places as well as for continuous manipulation. Gaze interaction can be used discreetly but is not suitable for continuous manipulation. Touch interaction is a convenient interaction method that is often used in daily life; however, it is difficult to interact with virtual objects using a touchpad attached to an HMD while performing real tasks. Furthermore, the gestures used for touch interaction are noticeable in public places and cause arm fatigue when performed over prolonged periods.

Many interfaces have been proposed to recognize touch interaction at various locations in mobile environments. Touch gestures can be recognized by the built-in camera of an HMD, but the occlusion problem may arise when holding a real object. Recognizing touch interaction by attaching a touch sensor to the body enables unobtrusive interaction because no interaction occurs around the face. However, limitations exist in that the interaction can be recognized only at the position at which the sensor is attached because the location and range of the touch sensor are fixed and the interaction can be recognized only at a fixed position or in a specific posture. Although magnetic sensors can track the position of a finger when it is occluded by an object, the methods of recognizing touch interaction with these sensors require specific postures because of the limited magnetic sensor range.

As optical sensors can track object positions regardless of the recognition range, methods of recognizing touch interaction with such sensors have been proposed. However, attaching an optical sensor to the pad of a fingertip interferes with natural haptic feedback, and optical sensors encounter difficulty in recognizing gestures on some materials such as glass. Therefore, touch interaction must be recognizable regardless of the hand posture when holding a real object. Further, the interface should not interfere with the natural haptic feedback of a hand interacting with a real object.

To overcome these challenges, this paper proposes FingerTouch, which recognizes touch interaction, via a wireless IMU sensor attached to the fingernail, for use with AR HMDs in mobile environments (as shown in Fig. 1). FingerTouch is a new touch method that tracks the fingertip position and recognizes touch gestures on any plane by analyzing only the data obtained from the nine-axis IMU sensor attached to the fingernail. FingerTouch can also recognize touch interaction regardless of the orientation or material of the plane that the finger touches and can even recognize touch interaction when the hand is holding a real object. Moreover, the interface used for FingerTouch minimizes interference with real tasks because the sensor is attached to the fingernail. The contributions of the study are as follows.

- Touch interaction can be recognized using only nine-axis IMU sensors.
- Touch interaction can be identified regardless of the orientation or material of the plane the finger touches.
- The gestures made for touch interaction are unobtrusive and thus appropriate for use in public places.

The Related Work section describes related studies on interaction methods used in mobile HMDs and recognition methods of finger touch gestures. The Methods section describes the FingerTouch algorithm. The Experimental Evaluation section outlines how the performance and usability of FingerTouch are evaluated. The Results section analyzes the data obtained from the user experiments. Finally, the Discussion section describes the features of FingerTouch based on the experimental results.

II. RELATED WORK

A. METHODS OF INTERACTING WITH VIRTUAL CONTENT ON AR HMDs IN MOBILE ENVIRONMENTS

In commercialized HMDs that support mobile environments, typical methods of manipulating virtual content are mid-air hand interaction, handheld wand interaction, and touch interaction [6]. There are methods of manipulating virtual content using eye tracking and head motion, but these methods are used in conjunction with other interaction methods rather than independently [7]. Although voice recognition has the advantage that virtual content can be manipulated regardless of hand motions, continuous operations such as changing the location of virtual content are difficult, and recognition is limited in noisy places.
Mid-air hand interaction allows natural interaction with virtual content, which is similar to manipulating objects in real life. The methods of recognizing hand gestures using a camera [8], [9] enable easy simultaneous interaction with real objects because they allow interaction with virtual content without requiring other devices to be attached to the hand of the user. However, the user must interact with raised hands because the camera is embedded on the front of the HMD, and its angle of view is limited, and repeated interaction with virtual content through mid-air hand interaction increases physical fatigue in the arm of the user [10]. Moreover, hand interaction is limited when holding a real object. Mid-air hand interaction with a glove on can facilitate free interaction with virtual content because the recognition position is not fixed. However, interacting with a real object via a data glove on the hand can interfere with natural haptic feedback, which can cause discomfort when performing everyday tasks.

Handheld wand interaction allows the user to manipulate virtual content through various input sensors (buttons, IMUs, sensors, touch, etc.) in the wand while holding the wand in his or her hand. Users do not prefer the wand method because they must carry a wand in addition to wearing an HMD in public places [4]. Furthermore, it is difficult for this method of manipulating virtual content to work compatibly with the manipulation of real objects because the user must always hold the wand.

Touch is a popular interaction method for manipulating windows, icons, menus, and pointers (WIMP) on laptop computers and smartphones. Commercial HMDs (e.g., Google glasses [11], Samsung GearVR [12]) use attached touchpads for virtual content manipulation through touch interaction. Touchpads are highly mobile because they are attached to HMDs, but as with mid-air hand interaction, users must lift their hands to perform manipulations, and prolonged operation can increase user fatigue. Moreover, even though touch interaction is a familiar mode for users, it is difficult to use a touchpad because the touchscreen is on the side of the HMD [13]. Attaching the touchscreen to the front of the HMD allows direct manipulation of virtual objects through touch interaction and offers good mobility because the input device is attached to the HMD. However, this configuration is inappropriate for optical see-through HMDs [14] because the touchscreen would block the view of the real space. The touch recognition method using a camera embedded in an HMD [15] enables touch interaction recognition without requiring devices to be attached to the hand. However, as with mid-air hand interaction using a camera, the user must raise his or her hand and interact with the recognition range of the camera.

**B. METHODS OF RECOGNIZING TOUCH INTERACTION IN MOBILE ENVIRONMENTS**

All methods of recognizing touch interaction consist of one method for tracking the fingertip position and another for recognizing a finger gesture such as a tap. This section focuses on how to track fingertip positions in mobile environments, because a recognition method that uses a finger gesture such as a tap gesture for touch interaction can be replaced by adding an interface such as a button or pressure sensor. Four approaches currently exist for tracking fingertip positions in mobile environments.

1) **TOUCH SENSOR METHODS**

In this type of approach, a touch sensor is attached to a wearable device [16]–[20] or a body part for touch interaction [21]–[23]. Dobbelstein et al. [17] proposed a method of selecting an icon at a specific position with a touch sensor attached to the belt. Ashbrook et al. [16] and Ens et al. [18] performed touch interaction using unnoticeable gestures via a ring-shaped device in a mobile environment. Google Project Jacquard [19] and Karrer et al. [20] developed a fabric-type touch sensor and created a space in which the touch of cloth can be recognized. Kao et al. [22] and Weigel et al. [23] proposed a method of recognizing touch interactions by attaching sensors to the skin. These methods allow touch interactions to be performed in mobile environments, but it is difficult to perform touch interactions using certain hand positions or postures because the area in which touch interactions can be recognized is specified. Attaching a touch sensor to a fingernail allows touch interactions to be performed in various hand positions while enabling the touch interaction area to be recognized.

2) **CAMERA OR MICROPHONE METHODS**

An external sensor that is not attached to the hand or finger can be used to recognize the finger position. Approaches of this type include touch interaction recognition using a camera [24], [25], finger position tracking using speakers and microphones [26], [27], and methods involving radar [28]. Harrison et al. [24] proposed a technique for recognizing touch interactions anywhere by using a camera and a projector attached to the shoulder. Similarly, Niikura et al. [25] described a method of recognizing touch interactions using a camera attached to the wrist. Nandakumar et al. [26] and Zhang et al. [27] presented a method in which a specific waveform is generated through the speaker of a smartphone and the finger position is tracked with a microphone by recognizing the changes in the sound wave when the finger is moved. In the approach developed by Wang et al. [28] and Google Project Soli [29], finger gestures (e.g., finger rubbing) are recognized using a radar sensor. These methods do not involve attaching sensors to the hand and have wide recognition ranges compared to approaches in which sensors are attached to the hand. However, it is difficult to perform interactions with various hand postures because the finger cannot be recognized when it (or the hand) is hidden by the sensor. In the method proposed by Niikura et al. [25], the finger is rarely hidden by the sensor during daily work, but operation is difficult owing to the large size of the camera attached to the wrist.
3) MAGNETIC (ELECTROMAGNETIC) SENSORS METHODS
As magnetic field sensors and magnets (electromagnets) have become increasingly miniaturized, touch interaction methods attaching corresponding sensors have been proposed. The approach described by Chan et al. [30] can be used to recognize touch interactions by attaching a magnet to one fingernail and a sensor that can identify the magnet position to another fingernail. Yoon et al. [31] described a means of tracking finger positions or recognizing touch interactions by tracking the position of a magnet with a magnetic sensor embedded in a ring. Similarly, Chen et al. [32] proposed a method of tracking finger positions with electromagnets attached to fingernails. The technique presented by Zhang et al. [33] involves identifying touch interactions via an electromagnetic sensor embedded in a smartwatch that can recognize a specific waveform generated by a ring worn on a finger. However, these methods can only be used to interact with certain hand postures because of the fixed sensor position or the limited recognition range of the sensor.

4) OPTICAL SENSOR METHODS
Optical sensors are typically used in mice and have the advantage of being usable regardless of position. Yang et al. [34] proposed an interface in which an optical sensor is attached to the bottom of the finger, which is used to recognize touch interactions on various planes. Nirjon et al. [35] proposed an interface that allows the user to type text by wearing a ring embedded with proximity, acceleration, and optical sensors. Nguyen and Banic [36] suggested methods of manipulating 3D virtual objects through a ring-type interface with an embedded IMU and optical sensor. Because their methods involve an optical sensor attached to a finger, the natural haptic feedback may be disturbed, and the interaction may be limited depending on the material of the plane (for example, glass) contacted by the sensor.

5) IMU SENSOR METHODS
An IMU sensor, which consists of accelerometers, gyroscopes, and sometimes magnetometers, can measure its orientation regardless of its position. Fukumoto and Suenaga [37] proposed a method of inputting letters by recognizing finger tap gestures using a ring-shaped device (an accelerometer). Their method can be used to recognize finger gestures but not to track finger positions.

Kienzle and Hinckley [38] described a finger position tracking method using a ring-type interface to which a gyroscope and a proximity sensor are attached. Their approach can be used to track finger positions in limited finger postures but not to recognize finger gestures. Oh et al. [39] proposed an interface that recognizes gestures by tracking the finger position with an IMU attached to the fingernail. Unlike the method of Kienzle and Hinckley [38], that of Oh et al. [39] can be used to track finger positions and recognize finger gestures regardless of finger posture. However, it has limitations in that users must perform uncomfortable gestures to calibrate the plane.

III. METHOD
The FingerTouch algorithm consists of three steps. In Step 1, the slope of the plane that is touched by the finger is calculated. In Step 2, the relative position of the fingertip and the finger gesture are recognized. In Step 3, the finger position is mapped to a virtual space. Fig. 2 shows the flowchart for FingerTouch. The IMU sensor used for the FingerTouch algorithm consists of a three-axis accelerometer, a gyroscope, and a magnetometer.

A. STEP 1: CALCULATION OF SLOPE OF PLANE IN CONTACT WITH INDEX FINGER
When the user performs a double tap with a finger placed on the plane, as shown in Fig. 3, the slope of the plane touched
by the index finger is calculated. A double tap is recognized when the wave patterns of the acceleration (or velocity) and angular velocity (see the following subsection for details) match those of a double tap. If only the angular velocity wave pattern matches but that of acceleration (or velocity) does not (this gesture is called Cali Double Tap), the slope of the plane touched by the index finger is calculated. If the wave pattern of acceleration (or velocity) modified using the calculated slope of the plane matches that of a double tap, the slope of the plane is set accordingly.

The method used in FingerTouch to calculate the slope of the plane after recognizing a Cali Double Tap is as follows. As shown in Fig. 3, the slope of the plane \( q_{\text{plane}} \) is calculated from the angle \( q_{\text{IMU}} \) measured by the IMU attached to the fingertip and the angle \( q_{\text{finger2plane}} \) between the plane and the finger (see (1)). The slope of the plane \( q_{\text{plane}} \) calculated using (1) is employed to calibrate the angle \( q_{\text{IMU}} \) measured by the IMU with the slope of the plane (see (2)). The angle \( q_{\text{cali}} \) calculated using (2) allows finger position tracking and gesture recognition regardless of the slope of the plane.

\[
q_{\text{plane}} = q_{\text{IMU}} \cdot q_{\text{finger2plane}}^{-1} \quad (1)
\]
\[
q_{\text{cali}} = q_{\text{plane}}^{-1} \cdot q_{\text{IMU}} \quad (2)
\]

### B. STEP 2: TOUCH INTERACTION RECOGNITION
**USING FingerTouch**

This section explains how FingerTouch recognizes touch interactions regardless of the slope of the plane. First, it describes the method of recognizing the movement of the index fingertip regardless of the slope of the plane. Then, it explains the techniques used to recognize tap, double tap, and hold finger gestures, which are the basic touch gestures. Finally, it discusses the method employed to track the fingertip position and recognize the finger gesture simultaneously.

1) **FINGERTIP POSITION TRACKING**

This section describes the method of tracking the fingertip position on a tilted plane using the quaternion measured by the IMU. The index finger has four degrees of freedom (DOFs) [40]. When the index finger moves left and right, the metacarpophalangeal joint, wrist, and elbow are rotated; when the index finger moves forward and backward, the metacarpophalangeal, proximal interphalangeal, and distal interphalangeal joints are rotated.

The left–right movement of the index finger can be calculated as the z-axis value of the calibrated quaternion. The forward–backward movement of the index finger can be tracked according to the following two conditions: the finger moves on a plane and if no external force is applied, the angle between the proximal and intermediate phalanges when the angle between the intermediate and distal phalanges is \( \theta \) can be defined as \( 3/2 \ast \theta \) [39]. If the quaternion calibrated by (2) is used instead of the quaternion measured by the IMU, the forward–backward movement of the finger can be tracked even on a tilted plane.

\[
(PPL + MPL + DPL) \cdot \sin (q_{\text{finger2plane}}) = DPL \cdot \sin (q_{\text{cali}}) + MPL \cdot \sin (q_{\text{cali}})
\]
\[
-\theta + PPL \cdot \sin (q_{\text{cali}} - \frac{5}{2} \theta) \quad (3)
\]

In (3), \( PPL, MPL, \) and \( DPL \) are the proximal, middle, and distal phalanx lengths, respectively and each \( q \) is the y-axis value of the Euler angle.

2) **FINGER GESTURE RECOGNITION**

In touch interaction, the main finger gestures are tap, double tap, and hold. Touch interaction requires a method of recognizing finger gestures regardless of the slope of the plane in contact with the finger. The finger gestures are recognized via calculations using the three-axis angular velocity and three-axis measured acceleration by the IMU. When a finger gesture is recognized, the angular velocity waveform is measured regardless of the slope of the plane, but the acceleration waveform is measured very differently depending on the slope of the plane.

Because the acceleration obtained by the IMU includes gravitational acceleration, the gravitational acceleration is removed using (4), and the acceleration is calibrated based on the calculated slope of the plane.

\[
a_{\text{cali}} = q_{\text{plane}}^{-1} \cdot a_{\text{IMU}} - \vec{G} \quad (4)
\]

In (4), \( a_{\text{IMU}} \) is the acceleration obtained by the IMU, \( q_{\text{plane}} \) is the quaternion of the slope of the plane, and \( \vec{G} \) is the gravitational acceleration (9.8 ms\(^{-2}\)). The velocity is calculated by applying a low-pass filter (LPF) to the calibrated acceleration \( a_{\text{cali}} \) and then integrating it as shown in (5):

\[
\vec{v} = \int \text{LPF} (a_{\text{cali}}) \, dt \quad (5)
\]

Table 1 depicts the angular velocity and velocity waveforms for defined finger gestures. When a tap gesture is performed, the y-axis waveform of the angular velocity shows a positive peak followed by a negative trough, and the amounts of change in the x- and z-axis angular velocities are much smaller than that in the y-axis angular velocity. Additionally, the z-axis waveform of velocity shows a positive peak.
TABLE 1. Graphs of angular velocity and velocity for defined finger gestures.

| Finger Gesture | Angular Velocity Waveform | Velocity Waveform |
|----------------|---------------------------|------------------|
| **Tap**       | ![Tap Angular Velocity](image) | ![Tap Velocity](image) |
| **Double Tap** | ![Double Tap Angular Velocity](image) | ![Double Tap Velocity](image) |
| **Hold**      | ![Hold Angular Velocity](image) | ![Hold Velocity](image) |

followed by a negative trough, and when performing a tap gesture, the amounts of change in the x- and y-axis velocities are much smaller than that in the z-axis velocity. As a double tap is a gesture in which the tap gesture is performed twice consecutively, it can be recognized by the angular velocity and velocity waveforms corresponding to a tap gesture appearing twice consecutively. A hold is recognized when the acceleration and angular velocity waveforms do not change for a certain length of time (approximately 0.5 s).

3) INTEGRATION OF FINGERTIP POSITION TRACKING AND FINGER GESTURE RECOGNITION

For touch interaction, the index fingertip position must be tracked and the finger gesture must be recognized simultaneously. However, problems arise when simply performing fingertip tracking and finger gesture recognition in parallel. When performing a (double) tap gesture, the fingertip positions when starting and ending the gesture are different because the fingertip position during forward–backward movement above the plane is still tracked by the proposed position-tracking algorithm. This characteristic makes it difficult for users to perform finger gestures at the correct positions. Therefore, a method of not tracking the finger position during finger gesture execution is required. This section describes a method of determining whether or not to track the fingertip position using the velocity calculated as explained under “2) FINGER GESTURE RECOGNITION” is described.

The tap and double tap gestures consist of raising and lowering the finger; thus, the acceleration changes more along the z-axis acceleration than along the other axes. Therefore, the determination of whether to track the fingertip position is made after comparing the z-axis waveform with the x- and y-axis waveforms. The z-axis velocity is compared not with the signals of the other axes, but rather with the values obtained by applying (5):

\[
Z_{\text{filter}} = \begin{cases} 
F_0 \cdot e^{-\frac{t}{C}}, & \text{if } |\vec{v}_z| < F_0 \cdot e^{-\frac{t}{C}} \\
|\vec{v}_z|, & \text{if } t = 0, F_0 = |\vec{v}_z| 
\end{cases}
\]  

(6)

In (6), \(t\) is the time elapsed from the moment at which the z-axis of the calculated velocity becomes larger than \(Z_{\text{filter}}\). \(F_0\) is the absolute value of the calculated z-axis velocity when the z-axis of the calibrated velocity is larger than \(Z_{\text{filter}}\), and \(C\) is a positive constant value.

Equation (7) compares the z-axis velocity calculated using (6) with the absolute values of the x- and y-axis

\[
Z_{\text{filter}} = \begin{cases} 
F_0 \cdot e^{-\frac{t}{C}}, & \text{if } |\vec{v}_z| < F_0 \cdot e^{-\frac{t}{C}} \\
|\vec{v}_x|, & \text{if } t = 0, F_0 = |\vec{v}_x| 
\end{cases}
\]  

(7)
TABLE 2. Waveforms of the designed filter in tap and double tap gesture.

| Finger Gesture | Tap | Double Tap |
|----------------|-----|------------|
| Angular Velocity Waveform | ![Waveform Image] | ![Waveform Image] |
| Velocity Waveform | ![Waveform Image] | ![Waveform Image] |
| Velocity Filter Waveform | ![Waveform Image] | ![Waveform Image] |

velocities:

\[
C_{\text{passfilter}} = \begin{cases} 
0, & Z_f > T \cap Z_f > |\vec{v}_z| \\
0, & Z_f > T \cap Z_f > |\vec{v}_x| + |\vec{v}_y| \\
1, & \text{otherwise} 
\end{cases} \tag{7}
\]

In (7), \(T\) is the threshold. If \(C_{\text{passfilter}}\) is high, the fingertip position is not tracked; if it is low, the fingertip position is tracked. Table 2 illustrates the waveforms of \(Z_f\) and \(C_{\text{passfilter}}\) for the tap and double tap gestures.

In (7), \(T\) is the threshold and \(Z_f = Z_{\text{filter}}\). If \(C_{\text{passfilter}}\) is low, the fingertip position is not tracked; if it is high, the fingertip position is tracked. Table 2 illustrates the waveforms of \(Z_f\) and \(C_{\text{passfilter}}\) for the tap and double tap gestures.

C. STEP 3: MAP FINGER POSITION

COORDINATE MAPPING

Fig. 4 shows the range of movement of the fingertip when the index finger is moved. When the cursor is navigated using the fingertip position, the range of the left–right movement of the index finger varies according to the bending of the index finger on the plane, which in turn changes the range of movement of the cursor. Therefore, a method of moving the cursor position to the left and right by the same amount regardless of the bend of the finger is required. This section explains a means of converting the fan-shaped space of the index finger movement into a rectangular space.

To convert a fan-shaped space into a rectangular space, as shown in Fig. 4, it is necessary to determine the maximum and minimum rotation angle measured when the user moves the index finger the maximum amount to the left or right, as well as the maximum and minimum distance from the wrist to the fingertip when the user moves the index finger forward and backward. In general, the range of movement is limited by anatomical factors when the user moves the index fingertip on a plane. In Fig. 4, “Init. Point” is the point at which a Cali Double Tap was performed to calculate the slope of the plane in contact with the finger.

By referring to studies on the range of movement of the index finger [40, 41], we set the range in which the index finger can move based on the position of “Init. Point”. Based on the angle obtained from the IMU sensor at the position of “Init. Point”, the minimum and maximum angles are the z-axis angles of \(-30^\circ\) and \(+30^\circ\), respectively, and the maximum and minimum distances are calculated using (3) when the angle with respect to the x-axis is \(-85^\circ\), as shown in Fig. 4. The fan-shaped space can be mapped to the
coordinates of the rectangular space by using the maximum and minimum of the set quaternion and the maximum and minimum length as depicted in Fig. 4. Therefore, the cursor movement range will be constant regardless of the bend of the finger.

IV. EXPERIMENTAL EVALUATION

To evaluate the proposed FingerTouch method, two user studies were conducted and a questionnaire survey was administered. The first experiment was designed to evaluate the accuracy of cursor position navigation using finger gestures. The second experiment, in which a participant selected a virtual object by using FingerTouch, involved measuring the cursor position and finger gesture recognition accuracy as well as the completion time according to the experimental conditions. The survey, which was administered using a system usability scale (SUS) questionnaire [42], compared the usability of FingerTouch with that of a touchpad.

A. CURSOR NAVIGATION TASK

An experiment was performed to assess the performance of cursor navigation using the fingertip position. The participants conducted a task in which they drew the borders of figures displayed on a 27-inch monitor with a moving cursor. We asked the participants to draw the outlines of a square, an equilateral triangle, and a circle by moving the cursor in parallel, diagonal, and curved lines using FingerTouch. As shown in Fig. 5, a 13 cm × 13 cm square, an equilateral triangle with a side length of 13 cm, and a circle with a 6.5 cm radius were displayed on the monitor. The order of drawing was random, and the participants moved their fingers on a wooden desk to perform drawing. Before beginning the experiment, \( q_{\text{finger2plane}} \) was measured after calibrating the IMU sensor attached to the fingernail on the index finger of each participant.

Apparatus: A personal computer with an Intel i5 4690 CPU, 8 GB of RAM, and an NVIDIA GTX 1060 graphics card was used. For the wireless IMU sensor attached to the fingernail, an E2Box model EBIMU24GV2 was used.

Participants: There were 17 participants in this experiment (mean age \( M \) 26.4 years, standard deviation (std) 2.6 years; 13 males and 4 females). All of them were right-handed and had experience using PCs.

B. FINGER GESTURE RECOGNITION WITH CURSOR NAVIGATION TASK

In the second experiment, the finger gesture recognition (for tap and double tap) and cursor position accuracy as well as the completion time were measured while participants wearing an AR HMD in mobile environments selected a virtual object with FingerTouch. The finger gesture recognition accuracy was the degree to which the intended virtual object was selected by the users. The errors in finger gesture recognition
FIGURE 6. Plane materials and plane angles for FingerTouch gesture recognition rate and position accuracy measurements.

FIGURE 7. 4 × 3 objects with 10 mm diameters arranged.

included the errors of not recognizing the finger gesture and not selecting a virtual object due to shaking of the cursor position even though the finger gesture was recognized. The task completion time was measured in various situations to identify the effects of the situation on virtual object selection. The objective of the proposed method is to allow users wearing HMDs to interact conveniently with virtual content using one-finger touch gestures during their daily tasks.

To simulate representative daily tasks, the participants used FingerTouch on two plane angles (horizontal and vertical) and three plane materials (iron, acrylic, and wood), as shown in Fig. 6, to select 12 (4 × 3) virtual objects, as shown in Fig. 7. Three sizes (10 mm, 5 mm, and 2.5 mm) were used for the diameters of the virtual objects. The plane angle, plane materials, positions of the virtual objects, and sizes of the virtual objects were given in random order. In the user experiments, each participant performed a tap first, followed by a double tap. Before starting the experiment, the IMU sensor was calibrated in the same manner as described in the first experiment. The participants practiced with FingerTouch for more than 5 min in a sitting position before beginning the experiment.

Participants: There were 24 participants in this experiment (M = 26.79 years, std = 3.02 years; 16 males and 8 females). All of them were right-handed, and 13 of them had experienced AR previously.

C. USABILITY OF FingerTouch TASK

The usability of FingerTouch was compared with that of a touchpad, a well-known interaction tool. The participants conducted an experiment in which they selected virtual objects with a touchpad for approximately 10 min, then answered the SUS questionnaire and the survey questions in Table 3 on a five-point scale (1–5 points). Subsequently, we conducted individual interviews regarding the usability of FingerTouch.

TABLE 3. Survey questions.

| Q1  | Is the tested interface appropriate for use in a mobile environment? |
| Q2  | Can you perform the tested interaction unnoticeably in a public space? |
| Q3  | When you hold a physical object, does it interfere with the tested interface? |

Apparatus: The All-in-One Media Keyboard [43] was used as a touchpad for comparison with FingerTouch.

Participants: The same participants were involved in this experiment as in the second experiment.

V. RESULTS

Data were collected by a system log program developed on Unity3D in each experiment, and the opinions of the participants were collected through the questionnaire.
and interviews. Shapiro-Wilk tests were performed for data normality. The normally distributed data were analyzed using the repeated measures analysis of variance (RMANOVA) tests, and the other data were analyzed using Friedman tests.

A. CURSOR NAVIGATION USING FINGER MOVEMENT
1) POSITION ERROR
The cursor position error was calculated as the L2 norm of the minimum distance between the figure outline and the cursor position, and the results are shown in Fig. 8. The Friedman test for the cursor position error showed a significant difference ($\chi^2(2) = 24.875, p < 0.001$). The Wilcoxon signed-rank post hoc test revealed that the average position error for the circle ($M = 1.50$ mm) was lower than those for the square ($M = 2.35$ mm) ($Z = -3.361, p < 0.005$) and equilateral triangle ($M = 2.40$ mm) ($Z = -3.516, p < 0.005$). The position errors for the equilateral triangle and square did not differ significantly.

To measure the position error in detail, the x- and y-axis position errors were examined separately. The Friedman test showed significant differences in the x-axis position error ($\chi^2(2) = 24.875, p < 0.001$) and y-axis position error ($\chi^2(2) = 32.000, p < 0.001$). A Wilcoxon signed-rank test performed on the x-axis position error revealed that the square ($M = 2.34$ mm) had a higher error than either the equilateral triangle ($M = 2.02$ mm) ($Z = -2.858, p = 0.010$) or the circle ($M = 1.10$ mm) ($Z = -3.464, p = 0.001$), and the equilateral triangle had a higher error than the circle ($Z = -3.516, p < 0.001$).

The same post hoc test performed on the y-axis position error revealed that the square ($M = 0.14$ mm) had a lower error than either the equilateral triangle ($M = 1.29$ mm) ($Z = -3.516, p < 0.001$) or the circle ($M = 1.01$ mm) ($Z = -3.516, p < 0.001$), and the equilateral triangle had a higher error than the circle ($Z = -3.516, p < 0.001$).

2) DRAWING COMPLETION TIME
The completion time for drawing a shape is the time it takes to draw the outline of the shape in one stroke. The results are shown in Fig. 9.

B. FINGER GESTURE RECOGNITION WITH CURSOR NAVIGATION
1) FINGER GESTURE RECOGNITION ACCURACY
The experimental data were analyzed to determine the accuracy of recognition of the proposed finger gestures (tap and double tap). The finger gesture recognition accuracy was calculated as $((\text{number of successes})/(\text{number of attempts})) \times 100$, and the results are presented in Figs. 10–13.
An RMANOVA (with the Huynh–Feldt correction for the sphericity violation) performed on the gesture recognition accuracy showed significant main effects for finger gesture ($F(1,19) = 11.186, p = 0.003, \eta^2 = 0.371$), plane angle ($F(1,19) = 7.095, p = 0.015, \eta^2 = 0.271$), and target size ($F(1.779, 33.807) = 72.137, p < 0.001, \eta^2 = 0.792$).

Pairwise comparisons revealed that the double tap gesture accuracy ($M = 97.28\%$) was significantly higher than the tap gesture accuracy ($M = 94.77\%$) ($p = 0.003$). The finger gesture recognition accuracy on the vertical plane ($M = 95.50\%$) was significantly lower than that on the horizontal plane ($M = 96.56\%$) ($p = 0.015$). The Bonferroni correction revealed that the finger gesture recognition accuracy for the 2.5 mm target ($M = 93.05\%$) was significantly lower than that for either the 10 mm target ($M = 97.88\%$) ($p < 0.001$) or the 5 mm target ($M = 97.15\%$) ($p < 0.001$).

2) CURSOR POSITION ERROR

The experimental data were analyzed to determine the position error between the cursor and the virtual object when a participant selected an object by cursor navigation using FingerTouch. The cursor position error was calculated as the L2 norm of the error between the cursor position and the virtual object position when a finger gesture was recognized.

When the target virtual object was successfully selected. The results obtained when the virtual object was successfully selected are presented in Figs. 14–17. The RMANOVA test revealed a statistically significant difference in the cursor position error with the angle of the plane ($F(1,21) = 6.697, p = 0.017$). The cursor position error also differed with the target size with statistical significance. Pairwise comparisons
indicated that the cursor position error on the horizontal plane \((M = 1.91 \text{ mm})\) was lower than that on the vertical plane \((M = 2.05 \text{ mm})\) \((p = 0.017)\). Obviously, the position error for the 10 mm target \((M = 2.91 \text{ mm})\) was larger than that for either the 5 mm target \((M = 1.77 \text{ mm})\) \((p < 0.001)\) or the 2.5 mm target \((M = 1.27 \text{ mm})\) \((p < 0.001)\).

The cursor position error was again analyzed by separating the x- and y-axis values. The RMANOVA test of the x-axis cursor position error showed statistically significant differences with the plane angle \((F(1,21) = 11.435, p = 0.003, \eta^2 = 0.353)\) and target size \((F(2,42) = 113.562, p < 0.001, \eta^2 = 0.844)\). The pairwise comparison using the Bonferroni test demonstrated that in contrast with the total error, the x-axis position error on the horizontal plane \((M = 1.19 \text{ mm})\) was higher than that on the vertical plane \((M = 1.10 \text{ mm})\) \((p = 0.003)\). The x-axis position error for the 10 mm target \((M = 1.78 \text{ mm})\) was larger than that of either the 5 mm target \((M = 1.02 \text{ mm})\) \((p < 0.001)\) or the 2.5 mm target \((M = 0.65 \text{ mm})\) \((p < 0.001)\).

Similarly, the RMANOVA test of the y-axis error revealed statistically significant differences with the plane angle \((F(1,21) = 28.485, p < 0.001, \eta^2 = 0.576)\) and target size \((F(2,42) = 134.128, p < 0.001, \eta^2 = 0.865)\). As with the total cursor position error, the pairwise comparison showed that the y-axis position error of the cursor on the horizontal plane \((M = 1.23 \text{ mm})\) was smaller than that on the vertical plane \((M = 1.48 \text{ mm})\) \((p < 0.001)\).

When selection of the target virtual object failed. The results obtained when the target virtual object was not correctly selected are presented in Figs. 18–21. The cursor position was analyzed separately in the x- and y-directions. The error in the x-axis position exhibited a statistically significant difference with finger gesture \((F(1,21) = 9.766, p = 0.005, \eta^2 = 0.317)\). Other factors did not differ with statistical significance. The error in the x-axis position for the tap gesture \((M = 1.93 \text{ mm})\) was smaller than that for the double tap gesture \((M = 2.05 \text{ mm})\) \((p = 0.005)\). The error in the y-axis position showed a statistically significant difference by finger gesture \((F(1, 21) = 17.649, p < 0.001, \eta^2 = 0.457)\) and plane angle \((F(1,21) = 7.454, p = 0.013, \eta^2 = 0.262)\). The error in the y-axis position for the tap gesture \((M = 4.06 \text{ mm})\) was much larger than that for the double tap gesture \((M = 2.77 \text{ mm})\) \((p < 0.001)\). Unlike the error in the x-axis position, the error in the y-axis position on the horizontal plane \((M = 3.04 \text{ mm})\) was greater than that on the vertical plane with statistical significance \((M = 3.80 \text{ mm})\) \((p = 0.013)\).

3) COMPLETION TIME FOR SELECTION

The completion time for selection was calculated from the time the previous selection was completed to the time the current selection was completed. The results are presented in Figs. 22–25. An RMANOVA (with the Huynh–Feldt correction for the sphericity violation) performed on the selection time revealed significant main effects for finger gesture \((F(1,19) = 2.463, p = 0.001, \eta^2 = 0.463)\), plane angle \((F(1,19) = 17.327, p = 0.001, \eta^2 = 0.477)\), plane material \((F(2,38) = 9.475, p < 0.001, \eta^2 = 0.333)\), and target size \((F(1.977,37.563) = 95.229, p < 0.001, \eta^2 = 0.834)\).
on the iron plane ($M = 3.15$ s) than on the wood plane ($3.56$ s ($p = 0.001$) or the acrylic plane ($3.57$ s ($p = 0.001$). The final finding is related to Fitt’s law: the participants selected the 10 mm targets ($M = 2.69$ s) significantly faster than the 5 mm targets ($3.31$ s ($p < 0.001$) or 2.5 mm targets ($M = 4.28$ s ($p < 0.001$). In addition, the participants selected the 5 mm targets significantly faster than the 2.5 mm targets ($p < 0.001$).

The Bonferroni-corrected post hoc test revealed that the participants performed the double tap gesture ($M = 2.94$ s) significantly faster than the tap gesture ($M = 3.90$ s) ($p = 0.001$). Further, it took significantly longer to use the FingerTouch interface in the vertical plane ($M = 3.71$ s) than in the horizontal plane ($M = 3.14$ s) ($p < 0.001$). Interestingly, the participants performed the gestures faster on the iron plane ($M = 3.15$ s) than on the wood plane ($3.56$ s ($p = 0.001$) or the acrylic plane ($3.57$ s ($p = 0.001$). The final finding is related to Fitt’s law: the participants selected the 10 mm targets ($M = 2.69$ s) significantly faster than the 5 mm targets ($3.31$ s ($p < 0.001$) or 2.5 mm targets ($M = 4.28$ s ($p < 0.001$). In addition, the participants selected the 5 mm targets significantly faster than the 2.5 mm targets ($p < 0.001$).

C. USABILITY OF FingerTouch

The average SUS scores are shown in Fig. 26. The average score for the touchpad ($M = 74.05$) was higher than that for FingerTouch ($M = 70.12$), but the paired $t$ test showed that the difference was not statistically significant ($t(20) = -0.788$, $p = 0.440$).

The results for the questions listed in Table 3 are presented in Fig. 27. The Wilcoxon signed-rank test revealed a statistically significant difference between FingerTouch and the touchpad. Participants answered that while wearing an AR HMD, it was more convenient to use the FingerTouch interface ($M = 3.24$) than the touchpad device ($M = 2.00$) ($Z = 2.751$, $p = 0.006$). On Question 2 in Table 3, the participants answered that FingerTouch interaction ($M = 3.67$) allows for less noticeable interactions than interaction with the touchpad ($M = 2.19$) ($Z = -3.573$, $p < 0.001$). On the last question, they answered that FingerTouch ($M = 4.05$) was more convenient to interact with than the touchpad ($M = 1.91$) when holding the real object ($M = 1.91$) ($Z = -3.852$, $p < 0.001$).
Through the user interviews, we received the positive feedback that FingerTouch was an excellent portable interface for use anywhere, was readily usable regardless of the material or slope of the plane, provided intuitive interaction and was easy to learn how to use. Some of participants provided negative feedback that the sensor should be smaller than the 5 fingernail.

VI. DISCUSSION

A. CURSOR NAVIGATION

We conducted user experiments to verify that the proposed FingerTouch can be used to operate WIMPs in AR HMDs. Specifically, we examined how accurately cursor navigation can be performed using FingerTouch.

In this experiment, the position error for the circle was the lowest. The position errors for the equilateral triangle and square were similar, but the position error for the square was slightly lower than that for the equilateral triangle. Interesting results were obtained when the position errors were separated into x- and y-axis components. The error in the x-axis position for the square was higher than those for the other shapes, whereas the error in the y-axis position for the square was lower than those for the other figures. The reason that the x-axis position error of the square was higher than those of the other shapes may be that the mental mapping was inconsistent because the fingertip moved in a fan shape, but the cursor moved horizontally. On the other hand, when the finger was bent, the fingertip movement path and y-axis movement of the cursor were both straight, so the mental mapping was not vastly different. The low y-axis position error of the square indicates that the users could conveniently control the y-axis position of the cursor. Since the y-axis position of the cursor was manipulated according to the degree of bending of the finger, the users could change it more conveniently than the x-axis position.

One reason for the high two-dimensional position error of the equilateral triangle is that it was necessary for the users to pay attention to both the x- and y-axis movement of the cursor when they drew the sides of the equilateral triangle with FingerTouch. In particular, the y-axis position error of the equilateral triangle was the highest because when the users moved their finger diagonally, the change in the bend of the finger did not affect the y-axis position of the cursor as the users intended. Specifically, it was necessary for the users to move their fingertip left and right to adjust the x-axis position of the cursor and simultaneously to bend their finger to change the y-axis position of the cursor. To move the cursor diagonally, it was necessary for the participants to draw curves with their fingertip, which means that the error increased due to mental mapping.

The reason that the x- and y-axis position errors of the circle were relatively low compared to those of other shapes may be the fact that the fingertip and cursor paths provided the most similar mental mapping in this case.

The first experiment showed that the FingerTouch method of moving the cursor in horizontal, vertical, and curved lines resulted in reduced position errors, but the position errors were greater when moving the cursor diagonally. The average position error of the equilateral triangle was 2.40 mm, which is relatively high.

B. FINGER GESTURE RECOGNITION WITH CURSOR NAVIGATION

In the second experiment, we examined the usability when a user wearing an AR HMD performed cursor navigation and finger gestures simultaneously using FingerTouch.

1) FINGER GESTURES

It is possible that the double tap gesture recognition rate was higher than the tap gesture recognition rate due to the learning effect, because all participants performed trials to select a virtual object with a tap and then with a double tap gesture. Nevertheless, the success rate for tap recognition in this study was higher than that reported by Oh et al. [39]. It was difficult to conduct the finger gestures at the correct positions using the method of Oh et al. [39] because finger gesture recognition was performed simultaneously with finger position tracking. In contrast, because our method does not track the finger position when a finger gesture is performed, using the filter described in the “3) INTEGRATION OF FINGERTIP POSITION TRACKING AND FINGER GESTURE RECOGNITION” subsection, it produced a high finger gesture recognition rate.

2) PLANE ANGLE

Manipulating the cursor in the vertical plane was difficult for the users, as a mouse is usually operated in the horizontal plane. Therefore, the finger gesture recognition rate for the vertical plane was lower than that for the horizontal plane, and the completion time on the vertical plane was longer than that on the horizontal plane. Further, some participants stated that it was more difficult to manipulate the cursor on the vertical plane than on the horizontal plane. Nevertheless, the rate of finger gesture recognition on the vertical plane was greater than 95%, and the completion time was 3.71 s, which was not much different from the completion time of 3.14 s on the horizontal plane.

3) MATERIAL

The finger gesture recognition rate did not differ according to the plane material, but the completion time was faster on the iron plane. Some of the participants said that acrylic planes were more difficult than other planes because of friction. There was no statistically significant difference in completion time between the wooden and acrylic materials. Therefore, additional experiments are needed in the future to ascertain whether frictional forces affect completion time.

4) TARGET SIZE

The object selection rate for the 10 mm and 5 mm targets was 97.88% and 97.15%, respectively, and the recognition rate between the two was less than 1%. However, the object
selection rate for the 2.5 mm target was 93.05%, more than 4% lower. In other words, the participants found it difficult to select the 2.5 mm target. According to Fitt’s law [44], selection time increases as target size decreases. The difference in the mean completion time between the 10 mm and 5 mm targets was greater than that between the 2.5 mm and 5 mm targets, which shows that it was difficult for participants to select the 2.5 mm target. Hence, we suggest that the target size should exceed 2.5 mm to provide a maximally positive user experience with FingerTouch.

5) POSITION ERROR
The 12 object positions did not show significant differences in terms of finger gesture recognition accuracy, completion time, and position error. When object selection failed, the position error of the x-axis was not significantly different, but that of the y-axis was significantly larger. The reason for the large error in the y-axis position is that when a tap (or double tap) is performed slowly, the cursor position does not stay fixed, and the y-axis position moves temporarily if $Z_{filter}$ is lower than the threshold. However, the difference between the x- and y-axis position errors for the double tap gesture was markedly lower than that for the tap gesture in the subsequent experiments. It can be said that the position error was reduced in the double tap case because of the learning effect. One of the participants stated that it appeared that the double tap was better recognized because it was performed faster than the tap. The position error was not higher than those found in other studies [13], [15].

C. USABILITY OF FingerTouch
In the SUS questionnaire, FingerTouch scored lower than the commercially available touchpad, but not with statistical significance. Thus, FingerTouch can provide users with the basic usability of a touchpad.

The basic usability of the interface is important, but it is also critical for FingerTouch to be conveniently usable in AR HMDs. As expected, FingerTouch is more mobile than commercially available touchpads and enables unnoticeable motions to be performed in public. The user survey results showed that FingerTouch allows users to perform touch interactions with one figure regardless of the slope of the plane and is compatible with other tasks. However, additional user experimentation is required to assess whether FingerTouch is compatible with actual tasks performed in everyday life.

In the interview, some participants provided very positive feedback after finding that the touch interactions were perceived on curved thighs or arms rather than flat ones. In the future, it will be necessary to verify the usability of FingerTouch on curved surfaces and soft objects through user experiments.

D. LIMITATIONS
The FingerTouch finger gesture recognition method is based on an IMU sensor; consequently, calibration is required in environments due to changes in geomagnetism. Further, in a bent-finger posture with pressure on the fingertip, it is difficult to track the fingertip position accurately. In future work, the development of a wireless IMU sensor that is as small as a fingernail will enable further size reduction. FingerTouch allows users to sense natural haptic feedback even while wearing the input device. In future work, we plan to investigate a method that enables users wearing AR HMDs to interact with virtual content using FingerTouch while performing real tasks.

VII. CONCLUSIONS
This paper proposed FingerTouch for simple, convenient, flexible, and socially acceptable touch interaction with virtual content through AR HMDs. FingerTouch can provide robust fingertip position tracking and finger gesture recognition, similarly to a touchpad. In user experiments, FingerTouch exhibited a low fingertip position error and high finger gesture recognition accuracy regardless of the slope or material of the plane the finger contacted. The user experiments demonstrated that FingerTouch can provide feasible touch interaction in various finger touch conditions. The participants stated that FingerTouch could be used in mobile environments with wireless IMU sensors that could be attached to a fingernail and be socially acceptable as they could then interact via unobtrusive gestures in public places. FingerTouch allows users to operate WIMP manipulations with touch interaction on AR HMDs. In future work, we plan to study new interactions with FingerTouch to enable various real-world tasks to be performed using mobile AR and to investigate realistic interaction by combining haptic feedback with FingerTouch interaction to manipulate virtual content. We anticipate FingerTouch to be usable as an input device for many tasks involving mobile devices as well as AR HMDs.

REFERENCES
[1] R. T. Azuma, “A survey of augmented reality,” Presence, Teleoperators Virtual Environ., vol. 6, no. 4, pp. 355–385, Aug. 1997, doi: 10.1162/ptve.1997.6.4.355.
[2] Microsoft. (2017). Hololens. [Online]. Available: https://www.microsoft.com/en-us/hololens
[3] D. Morris, T. S. Saponas, and D. Tan, “Emerging input technologies for always-available mobile interaction,” in Foundations and Trends in Human-Computer Interaction, vol. 4, no. 4, Norwell, MA, USA: Now Foundations and Trends, Oct. 2011, pp. 245–316. [Online]. Available: https://ieeexplore.ieee.org/document/8187122/?arnumber=8187122, doi: 10.1561/978161851984876.
[4] Y.-C. Tung, C.-Y. Hsu, H.-Y. Wang, S. Chyou, J.-W. Lin, P.-J. Wu, A. Valstar, and M. Y. Chen, “User-defined game input for smart glasses in public space,” in Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst., Seoul, South Korea, 2015, pp. 3327–3336, doi: 10.1145/2702123.2702214.
[5] Y.-T. Hsieh, A. Jylhä, V. Orso, L. Gamberini, and G. Jacucci, “Designing a willing-to-use-in-public hand gestural interaction technique for smart glasses,” in Proc. CHI Conf. Hum. Factors Comput. Syst., San Jose, CA, USA, May 2016, pp. 4203–4215, doi: 10.1145/2858036.2858436.
[6] L.-H. Lee and P. Hui, “Interaction methods for smart glasses: A survey,” IEEE Access, vol. 6, pp. 28712–28732, May 2018, doi: 10.1109/ACCESS.2018.2831081.
JU YOUNG OH received the B.S. degree in electrical engineering from Korea University, South Korea, in 2014. He is currently pursuing the Ph.D. degree with NT-IT (HCI and Robotics), Korea University of Science and Technology, Daejeon, South Korea. He is also a Research Student with the Center for Intelligent and Interactive Robotics, Korea Institute of Science and Technology, Seoul, South Korea. His research interests include user interfaces for augmented reality and virtual reality, various applications in augmented reality and virtual reality, and image processing using machine learning and deep learning.

JI-HYUNG PARK (Member, IEEE) received the Ph.D. degree in mechanical engineering from Seoul National University. He is currently a Research Scientist with the Korea Institute of Science and Technology. His research interests include HCI, artificial intelligence, and human–robot interaction. He is a member of the ACM, KSPE, KSME, and KHCI.

JUNG-MIN PARK (Member, IEEE) received the B.S. and M.S. degrees in computer science and the Ph.D. degree in computer science and engineering from Ewha Womans University, Seoul, South Korea, in 1989, 1991, and 2008, respectively. In 1991, she joined the Korea Institute of Science and Technology (KIST), Seoul, where she is currently a Senior Research Scientist with the Center for Intelligent and Interactive Robotics. Her research interests include virtual reality, human-centered interaction, human–robot interactions, and software architecture in networked robots.

* * *