ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20210116090844
Preliminary communication

Rotation Correction Method Using Depth-Value Symmetry of Human Skeletal Joints for Single RGB-D Camera System

Sunghyun KIM, Won-Hyung LEE*

Abstract: Most red-green-blue and depth (RGB-D) motion-recognition technologies employ both depth and RGB cameras to recognize a user's body. However, motion-recognition solutions using a single RGB-D camera struggle with rotation recognition depending on the device-user distance and field-of-view. This paper proposes a near-real-time rotational-coordinate-correction method that rectifies a depth error unique Microsoft Kinect by using the symmetry of the depth coordinates of the human body. The proposed method is most effective within 2 m, a key range in which the unique depth error of Kinect occurs, and is anticipated to be utilized in applications requiring low cost and fast installation. It could also be useful in areas such as media art that involve unspecified users because it does not require a learning phase. Experimental results indicate that the proposed method has an accuracy of 85.38%, which is approximately 12% higher than that of the reference installation method.

Keywords: human recognition systems; kinect; kinect depth error; near-real-time; RGB-D camera; rotation correlation; software tool

1 INTRODUCTION

Concomitant with technological advances focus has been placed on making human-computer interactions (HCIs) simple, rapid, and easy to understand [1]. In particular, human tracking was developed as a control method in the video game industry [2]. Advances in human tracking have led to the development of gesture interfaces and, subsequently, realization of the concept of a natural user interface [3], which has been studied since 1971.

Recently, with the development of infrared (IR) sensors and various image-processing techniques, the focus has been on devices that do not need to be mounted on the human body. One example is Kinect, a motion- and voice-recognition device based on the combination of red-green-blue (RGB) images and single depth images [4]. Kinect, originally developed as a game controller, provides good recognition rates despite using an inexpensive sensor. Soon after its launch, Kinect was hacked, which prompted Microsoft to patch the vulnerability and release the Kinect software development kit (Kinect SDK).

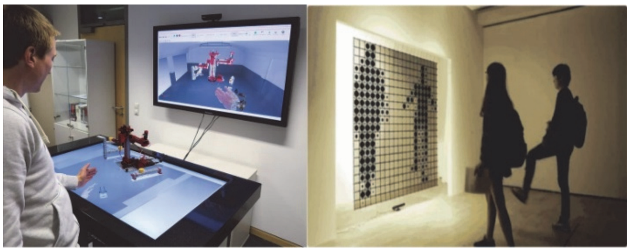


Figure 1 (Left) Example of an application system using a Microsoft Kinect device, which allows users to interact directly with both virtual and real objects in industrial Internet of things (IIoT) areas [5]. (Right) Example of an indoor media art installation based on a Kinect Pixel Mirror [6]

Kinect was soon applied in various fields, including the arts [7-10], industries [11-20], and healthcare [21-23]. It is widely used in various fields where intuitive HCI is required, as shown in Fig. 1, because it is a low-cost and relatively accurate motion-recognition device. However, in contrast to other motion-capture devices, Kinect creates 3D coordinates using a single 2D RGB image to determine the *x*, *y* coordinates and a single depth image to determine the *z* coordinate [2]. Therefore, to utilize Kinect, the user, Kinect, and screen must face each other, as shown in Fig. 1. Consequently, Kinect has installation limitations, and the recognition rate is significantly lower if they are violated. The limitations are caused by problems with rotation recognition, which cause a depth error unique to Kinect, and the requirement that Kinect, the user, and the screen must be aligned in a straight line. In addition, the original application of Kinect indoor video game playing can also be hindered if the user does not have sufficient space [2]. A fixed-coordinate-correction method [24], multiple cameras [25-27], and Kinect fusion [28] have been proposed as correction methods to overcome the installation limitation and decreased recognition rate. However, these methods employ a polar-coordinate system, which differs from the *x*, *y*, *z* coordinate system used in the Microsoft Kinect SDK. Although Microsoft released Kinect v2 for Windows and Xbox One, which addressed the recognition-rate problem to an extent [29], its installation locations are also limited [27, 30]. Moreover, various coordinate-correction methods that use cumulative data for the same learning method or converted coordinate systems have been studied [13, 31-34] with the same test sample. These methods, based on machine learning or multiple sensors, exhibit high accuracy (between 83% and 93%) but are not suitable for some application fields where machine learning is difficult to apply. As shown in Fig. 1, fields using motion-recognition interactions, such as IIoT and media art, require high accessibility, easy and fast installation, and rapid prototyping, which cannot be achieved with these machine learning-based coordinate-correction methods [35].

In this paper, we propose a near-real-time rotation-correction method using the depth-value symmetry of human skeletal joints with the unique Kinect coordinate system in a single RGB-D camera system. The proposed method can be quickly deployed in the existing Kinect applications without converting the unique Kinect coordinate system, and it enables quick installation, improved accessibility, and rapid prototyping by correcting coordinates without using machine learning or a sensor network. Furthermore, the proposed method is calibrated using only the upper-body coordinate features of the human body, and it can reduce the throughput and support an upper-body-only mode in Kinect for Windows. In addition, the proposed method is intended for correction at distances of 1 - 2 m, where the unique depth error of Kinect occurs [28]. An experiment was conducted for a distance of 3 m to demonstrate the reduction of the unique depth error of Kinect. The measured results were characterized by their standard deviation, standard error rate, and average error rate.

2 RELATED WORK

2.1 Motion Recognition Based on HCI

Gesture recognition without a wearable device is enabled by image recognition-based HCI, which helps the user adapt quickly and makes the system easily accessible [29]. Although a case of inputting the hand motion of a user via a mouse using general RGB cameras has been reported, it is difficult to separate user movements from background images in complex environments [36]. To overcome this issue, Kinect detects objects using both RGB and depth images, as shown in Fig. 2. The simultaneous use of both images facilitates the recognition of people and backgrounds even under poor lighting. As shown in Fig. 3a, Kinect uses a rectangular coordinate system for the x and y values, which are detected using an RGB camera. In contrast, it uses a spherical coordinate system for the z values (depths) of the background and users, which are separately detected using an IR radiation device and IR camera, respectively [2]. However, given that depth recognition using IR reflectance is based on real-world coordinates, we chose to use the polar-coordinate system, as shown in Fig. 3b.

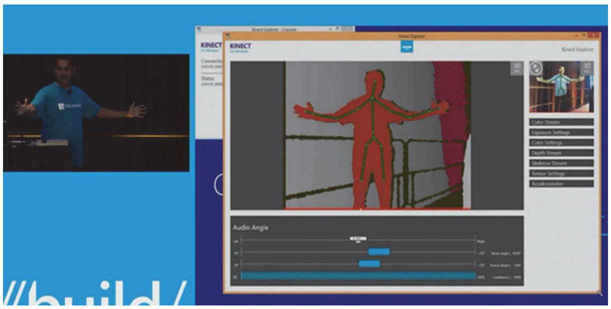


Figure 2 Illustration of Kinect human recognition using RGB and depth images

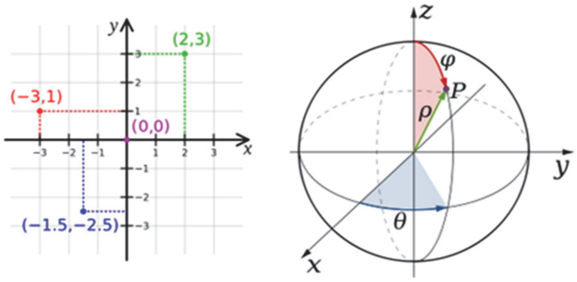
The unique combined coordinate system of Kinect is excellent in single-plane applications for tracking objects in front of it [2], which is why Kinect can provide good-quality 3D motion tracking as a single low-cost device. However, according to related research [2, 24, 30, 37], it produces numerous errors when tracking users who are rotated at angles with respect to the Kinect, rather than facing it directly. Methods such as multi-tracking and coordinate-system transformation have been used to overcome this disadvantage [24-27]. The polar-coordinate transformation method requires high-level understanding of the unique Kinect coordinate system, several computation processes [37-39], and an accurate installation angle [40].

A rotational transformation in Cartesian coordinates is expressed as follows:

$$\begin{aligned} x' &= x\cos\theta - y\sin\theta, \\ y' &= x\sin\theta - y\,cos\,\theta, \\ \begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} \cos\theta - \sin\theta \\ \sin\theta - cos\,\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \end{aligned} \tag{1}$$

As shown in Eq. (1) and Fig. 4, the rotational transformation requires the rotation angle θ. According to Kumar et al. [44], the Kinect coordinates do not include θ. Thus, θ must be derived from two different 3D transformations, making intuitive installation and fast user interaction difficult.

In the case of multi-Kinect tracking [24-26], the use of more than one Kinect sensor network will result in further space constraints. Although this method is highly accurate, the relative positions of each pair of sensors must be accurate, and it is difficult to obtain the exact position of a Kinect device if it is reinstalled or if the user changes.



(a) (b)

Figure 3 Two coordinate systems used in Kinect: (a) rectangular and (b) spherical coordinate

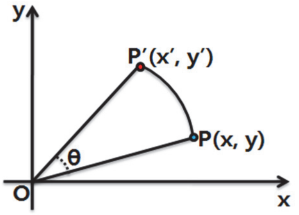


Figure 4 Rotation transformation from *P* to *P′* in the rectangular coordinate system

2.2 Combined Coordinate System

Kinect uses a unique combined 3D coordinate system (Fig. 5) created by combining the values obtained in two different coordinate systems [24]. The *Z* value of this coordinate system is obtained using the Euclidean distance from the origin in 3D spherical coordinates. This value is combined with 2D rectangular coordinates to obtain the 3D coordinates. As shown in Fig. 6, this unique coordinate system has a depth error that depends on the distance between the device and the user as well as the field-of-view (FOV). This depth error limits the installation and utilization of a single Kinect. The user, Kinect, and screen must be in a straight line to achieve accurate results.

The unique combined coordinate system of Kinect is used despite these disadvantages because it does not require wearable equipment or remote sensors. Thus, the equipment size can be reduced while maintaining rapid and accurate measurements in limited installations, unlike existing tracking equipment such as RGB cameras, motion trackers, and controllers.

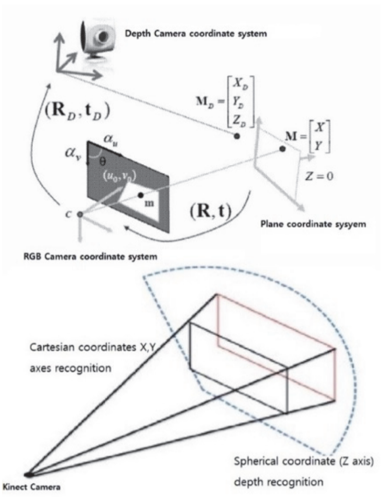

Depth Camera coordinate system
$(\mathbf{R}_D, \mathbf{t}_D)$
$\mathbf{M}_D = \begin{bmatrix} X_D \\ Y_D \\ Z_D \end{bmatrix}$
$\mathbf{M} = \begin{bmatrix} X \\ Y \end{bmatrix}$
α_u
α_v
θ
(u_0, v_0)
m
c
$Z = 0$
$(\mathbf{R}, \mathbf{t})$
Plane coordinate sysyem
RGB Camera coordinate system
Cartesian coordinates X,Y axes recognition
Spherical coordinate (Z axis) depth recognition
Kinect Camera

Figure 5 (Top) Plane of a calibration object relative to the coordinate system of a depth camera and an RGB camera [2, 17, 28]. (Bottom) Structural diagram of the unique combined coordinate system of Kinect

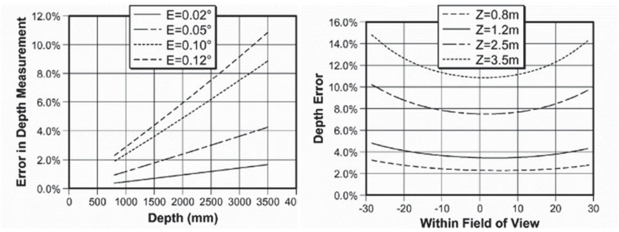
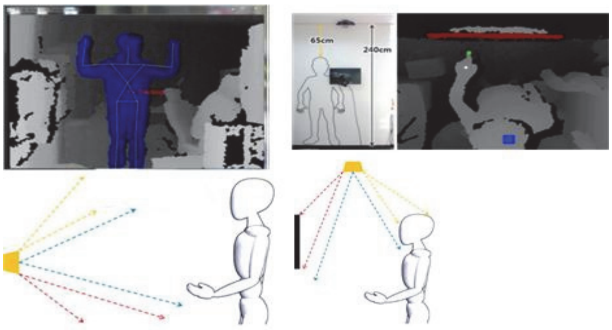


Figure 6 Graphs of Kinect depth error with respect to distance and FOV. (Left) Depth error rate with respect to distance for different device-user angles. (Right) Depth error rate with respect to FOV for different device-user distances (taken from [2]) (here, *E* = angle, and *Z* = depth = distance from Kinect)

2.3 Tracking and Application Method of Kinect

According to the patented in-house depth-camera calibration of Microsoft [2], the basic tracking method involves the placement of equipment at least 2 m in front of a user, as shown in Fig. 7a. This tracking method does not have a shade area for most human motions and is recommended to obtain the best recognition rate when using Kinect. However, Kinect has a significant limitation in terms of its installation environment and usage because the Kinect, screen, and user must be aligned in a straight line [2, 11].

An alternative tracking method involves installing equipment at a right angle (90°) to the line connecting the user and screen, as shown in Fig. 7b [2, 11]. This method can overcome a few installation limitations. However, it does not overcome the depth error shown in Fig. 6.



(a) (b)

Figure 7 Kinect IR camera screen and illustration for front and top tracking. (a) Reference front tracking. (b) Top tracking with axis conversion

2.4 Gesture Interface Recognition and Subject Rotation Recognition

To facilitate gesture recognition at various angles using Kinect, various methods have been applied, such as multilayer perception, artificial neural networks, and support vector machines (SVMs) [31-34, 44]. Tab. 1 summarizes the approaches, datasets, and accuracy of the previous studies that have attempted calibration using alternative methods. Numerous studies have been conducted on rotation recognition, most of which involved the recognition of gestures or sign language. Monir et al. [32] measured standing and sitting positions at distances in the range 1.3 - 3.5 m. They found that full-body tracking at 2 - 2.5 m yielded the highest accuracy. Tracking is difficult if the distance is too high or too low, if tracking is performed for only a part of the body, or if a rotation exists between the subject and the device.

Table 1 Summary of comparison of the proposed method and related studies

Author and year	Approach	Dataset	Distance from device	Range of detection	Accuracy / %	Initial accuracy before learning phase / %
Patsadu et al. [31], 2012	Joint positions, SVM, decision tree, neural network, reference Kinect setup, front-only tracking	Three gestures	2 - 3 m	90° body rotation in *xy* plane	93.72%	-
Monir et al. [32], 2012	3D joint positions, angular features, priority matching, full-body tracking	Four postures	2 - 2.5 m	0° - 30° body rotation in 3D space	89.1%	-
Almeida et al. [33], 2014	Hand shape, movement, position, SVM	34British sign language signs	2 m	hand signal detection only in *xy* plane	80%	48%
Uebersax et al. [34], 2011	Hand orientation, average neighborhood margin maximization algorithm	Seven American sign language alphabets	80 cm	0° - 130° hand rotation detection only in *xy* plane	88%	40%
Limet al. [19], 2016	Serial particle filter, covariance matrix, two-way tracking (0°, 90°)	50 American sign language signs	N/A	0° - 90° hand rotation detection only using depth plane	87.33%	-
Kumar et al. [44], 2017	Affine transformation, dynamic features, hidden Markov model, SVM, adaptive tracking (0° - 90°)	30 Irish sign language signs	2 m	0°, 45°, 90° body detection in 3D space	83.77%	34.74%
Proposed Method	3D hand joints, side correction, dynamic features,proportional and symmetric human skeletal joint positions, adaptive tracking (0° - 90°)	Three distances, three postures	1 m, 2 m, 3 m	0° - 90° body detection in 3D space	85.38%	-

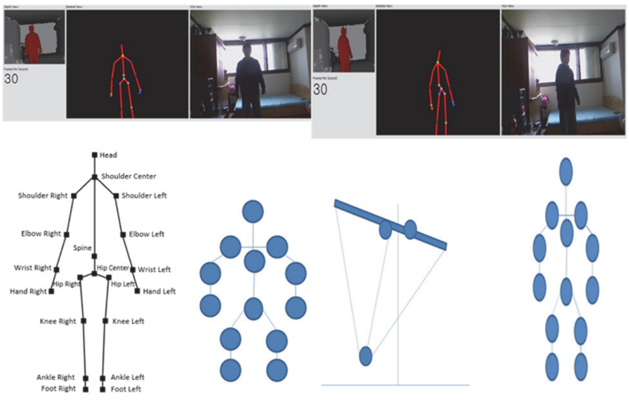
In the studies listed in Tab. 1, the distance [19, 31, 33, 34, 44] or angle [19, 31-34] between the subject and device was considered constant. Sign language recognition using hand rotation and gestures involves the use of a learning algorithm to compensate for the low initial recognition accuracy [33, 34, 44].

3 PROPOSED COORDINATE-CORRECTION METHOD

In this paper, we propose a rotation-correction solution for a single Kinect device based on the symmetry of the depth values of the human skeleton. The proposed solution focuses on the distance and rotation errors mentioned in Section 2 to enable fast installation and easy access to applications. It is also an initial-value-correction solution without a learning process, which is beneficial for applications involving unspecified users, such as media art or sculptures installed in public places.

The human body exhibits overall organic movements, rather than individual movements of one part or organ. However, some body parts, such as the shoulders of ordinary people without trauma or damage, are usually on the same depth plane in a symmetrical state. Therefore, we set a calibration reference point in the human body. Fig. 8a shows an example of a skeleton recognized by Kinect when the attention stance is tracked using the recommended front-installation method. The right side of Fig. 8a shows a simplified model of this skeleton, and the simplified model has the same depth values as the original model. Thus, the coordinate values can be expressed as follows:

$$\begin{aligned}&\text{Right hand}(x,y,z)=\left(R_{x1},R_{y1},R_{z1}\right),\\&\text{Left hand}(x,y,z)=\left(L_{x1},L_{y1},L_{z1}\right),\\&\text{Left shoulder depth value}\left(LS_{z1}\right),\\&\text{Center shoulder depth value}\left(CS_{z1}\right),\\&\text{Right shoulder depth value}\left(RS_{z1}\right),\\&R_{z1}=L_{z1}=RS_{z1}=LS_{z1}=CS_{z1}.\end{aligned}\tag{2}$$



(a) (b)

Figure 8 Illustration and Kinect camera screen of coordinate shift due to rotation between Kinect and user. (a) Skeleton tracking with front installation. (b) Skeleton tracking with rotation. In the first scan, if the subject is not in the front, Kinect fails to recognize that the subject is in a rotated state and instead recognizes the subject to be narrow, as shown in (b)

Owing to the characteristics of Kinect, the tracked x, y, z coordinates are twisted more frequently when the measurement is performed with the user at a greater rotation, as shown in Fig. 8b. This behavior is caused by the unique combined coordinate system of Kinect, in which the 2D rectangular coordinates are superimposed with the depth values in 3D spherical coordinates, without using the rotation equations or the same reference point. Therefore, when a user rotates, Kinect does not recognize this rotation. Instead, it registers a narrowing of the human body [2].

We used linear proportionality constants for the shoulder-arm ratio, distance, and rotation to compensate for the rotational errors caused by these structural characteristics. When the user rotates, the coordinate values are as follows:

$$\begin{aligned}&\text{Right hand}(x,y,z)=\left(R_{x2},R_{y2},R_{z2}\right),\\&\text{Left hand}(x,y,z)=\left(L_{x2},L_{y2},L_{z2}\right),\\&\text{Left shoulder depth value}\left(LS_{z2}\right),\\&\text{Center shoulder depth value}\left(CS_{z2}\right),\\&\text{Right shoulder depth value}\left(RS_{z2}\right).\end{aligned}\tag{3}$$

$$\begin{aligned}&R_{z2}\neq L_{z2},RS_{z2}\neq LS_{z2}\neq CS_{z2},\\&L_{z2}=LS_{z2},R_{z2}=RS_{z2}.\end{aligned}\tag{4}$$

Our approach is based on the fact that when a certain part of the human body, such as the shoulders or arms, is scanned in the Kinect reference position, it exists symmetrically on the same depth plane [41, 42]. The following equation can be derived from the data obtained using Eq. (2) to Eq. (4):

$$\begin{aligned}&L_{x1}:LS_{z1}=L_{x2}:LS_{z2},\\&L_{x1}=\frac{L_{x2}\cdot LS_{z1}}{LS_{z2}},\\&R_{x1}:RS_{z1}=R_{x2}:RS_{z2},\\&R_{x1}=\frac{R_{x2}\cdot RS_{z1}}{RS_{z2}},\\&LS_{z1}-RS_{z1}=\text{Differece of reference dept hvalue }Z_{\delta},\\&L_{z1}=L_{z2}-Z_{\delta}.\end{aligned}\tag{5}$$

As expressed above, the opposite side is calibrated based on a depth value from one side (the left or right side) by using a proportionality equation to allow for recognition as if both sides were symmetrically located at the same depth. Each coordinate modified into the same depth plane can be expressed as follows:

$$\begin{aligned}&\text{Left handposition}\left(L_{x1}=\frac{L_{x2}\cdot LS_{z1}}{LS_{z2}},L_{y1},L_{z2}-Z_{\delta}\right)\\&\text{Right hand position}\left(R_{x1},R_{y1},R_{z1}\right),\\&\text{Howevwe, }Z_{\delta}=LS_{z2}-RS_{z2}\\&\text{Suppose }RS_{z1}=RS_{z}.\end{aligned}\tag{6}$$

$$\text{Left handposition}\left(L_{x1}, L_{y1}, L_{z21}\right),$$

$$\text{Right hand position}\left(R_{x1} = \frac{R_{x2} \cdot RS_{z1}}{RS_{z2}}, R_{y1}, R_{z2} - Z_{\delta}\right), \quad (7)$$

$$\text{Howevwe, } Z_{\delta} = RS_{z2} - LS_{z2}$$

$$\text{Suppose } LS_{z1} = LS_{z}.$$

If $LS_z < RS_z$, use Eq. (6).

If $LS_z > RS_z$, use Eq. (7).

Eq. (6) and Eq. (7) express the depth values at the left and right sides, respectively, when the angles are modified by calibrating the opposite side. In the proposed method, Eq. (6) and Eq. (7) are used when the Kinect is located on the right and left sides of a subject, respectively. The proposed method should be corrected by considering the shoulder of the user that is closer to the device as the reference point. Otherwise, as shown in Fig. 6, the method is affected by the depth error due to the distance and FOV. This characteristic results in inverse correction, which increases the error rate.

4 EXPERIMENT

4.1 Experimental Method and System Setup

In this experiment, we used Kinect v1 because it is the most popular RGB and depth (RGB-D) dataset motion-capture device and is used in various fields [36, 43-45]. The unique depth error rate of Kinect has been reduced in Kinect v2 [29]. However, the resolution of the IR camera is 512 × 424 pixels, which is not directly proportional to the 1920 × 1080 pixel resolution of the RGB camera. The resolution of the RGB camera is common, but that of the IR camera is uncommon, which can make it difficult for end users to gain easy access to development and application [29].

Wasenmüller and Stricker [42] reported that Kinect v2 has a relatively constant depth-measurement accuracy compared to Kinect v1. However, it is affected by temperature, color, and multipath interference effects. It is also difficult to improve image recognition through algorithms. They also explained that devices based on Kinect v1 are suitable for fast access to applications and rapid prototyping.

Other 3D skeleton capture devices based on Kinect technology, such as XTION and XTION2 from ASUS and PrimeSense, use IR cameras with common resolutions of 640 × 480 pixels for depth recognition, similar to Kinect v1 [2, 40]. The basic principle of this device is the same as that of Kinect v1. Therefore, investigations using Kinect v1 can be expected to provide solutions that are compatible with other devices, such as XTION.

Fig. 9 shows a block diagram of the program we developed for measurement. This program simultaneously measures the coordinates corrected by the proposed method and the uncorrected coordinates. It caches data every 500 ms and outputs and saves the data to a file upon completion. Fig. 10 presents a screenshot of the interface of the coordinate measurement program for simultaneously measuring the original and calibrated coordinates. The experiment was performed in an environment in which obstacles did not obstruct the body of the subject. We used a location in which objects were placed naturally, such as a room, because we inferred that the sensor might cause errors between the subject and neighboring objects. In addition, we performed the experiment in an environment in which shadows were not directed toward the sensor by illuminating the environment using an LED light from above.

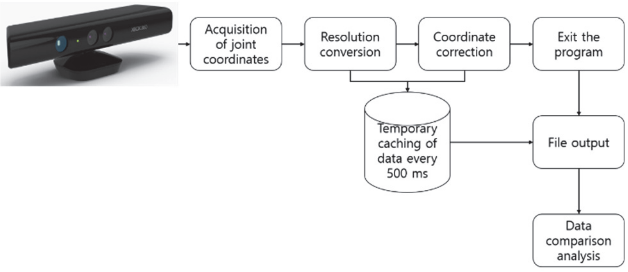
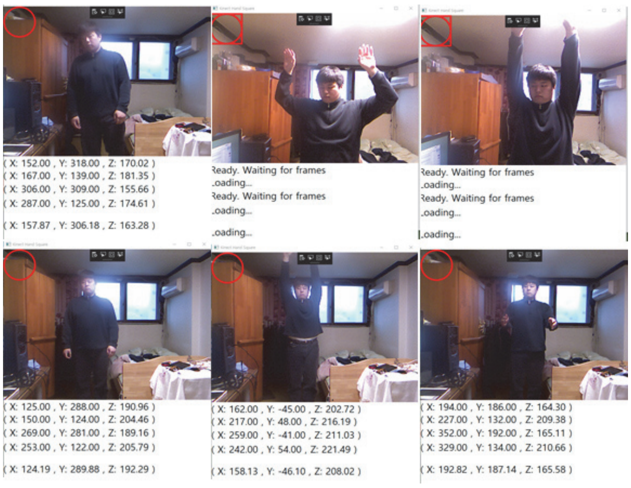


Figure 9 Block diagram of the program for measuring the coordinates corrected by the proposed solution and comparison with uncorrected coordinates



```
( X: 152.00 , Y: 318.00 , Z: 170.02 )
( X: 167.00 , Y: 139.00 , Z: 181.35 )
( X: 306.00 , Y: 309.00 , Z: 155.66 )
( X: 287.00 , Y: 125.00 , Z: 174.61 )

( X: 157.87 , Y: 306.18 , Z: 163.28 )
```

```
Ready. Waiting for frames
.oading...
Ready. Waiting for frames
.oading...

.oading...
```

```
Ready. Waiting for frames
.oading...
Ready. Waiting for frames
.oading...

.oading...
```

```
( X: 125.00 , Y: 288.00 , Z: 190.96 )
( X: 150.00 , Y: 124.00 , Z: 204.46 )
( X: 269.00 , Y: 281.00 , Z: 189.16 )
( X: 253.00 , Y: 122.00 , Z: 205.79 )

( X: 124.19 , Y: 289.88 , Z: 192.29 )
```

```
( X: 162.00 , Y: -45.00 , Z: 202.72 )
( X: 217.00 , Y: 48.00 , Z: 216.19 )
( X: 259.00 , Y: -41.00 , Z: 211.03 )
( X: 242.00 , Y: 54.00 , Z: 221.49 )

( X: 158.13 , Y: -46.10 , Z: 208.02 )
```

```
( X: 194.00 , Y: 186.00 , Z: 164.30 )
( X: 227.00 , Y: 132.00 , Z: 209.38 )
( X: 352.00 , Y: 192.00 , Z: 165.11 )
( X: 329.00 , Y: 134.00 , Z: 210.66 )

( X: 192.82 , Y: 187.14 , Z: 165.58 )
```

Figure 10 Screenshots of the program interface for several datasets

4.2 Experimental Results

In the experiments, we selected the following three basic postures for actions performed by seven different people: attention posture, hands-up posture, and hands-half-up posture. Measurements were performed for each posture at a distance of up to 2 m. This distance range has the most significant effect on the depth error. Owing to the nature of Kinect, the depth error does not occur at approximately 3 m (Fig. 6). Therefore, for demonstration, we performed measurements of only the attention posture at 3 m, where the error value sharply decreased.

In this experiment, the Kinect device was fixed at a location, and the subject was made to look to the front (A). Subsequently, the subject rotated until the angle between the Kinect and the subject became 90° (Fig. 11a, Fig. 11b and Fig. 11c). Next, the subject faced the front again (Fig. 11c, Fig. 11b, Fig. 11a). This procedure was followed to measure the calibration accuracy for a range of motion.

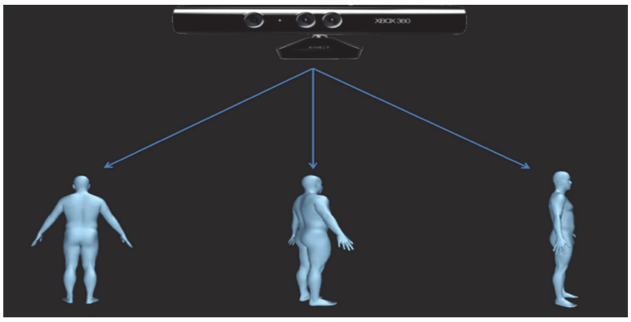


Figure 11 Subject at different viewing angles: (a) front view at 0°, (b) side view at approximately 45°, and (c) side view at approximately 90°

4.3 Standard Deviation Analysis

This section presents analyses of the raw and calibrated data. In Section 2, we described the depth error based on the subject rotation and FOV. The error rate increases with the subject distance and rotation angle [2]. Consequently, the correction rate decreases as the distance increases. Tab. 2 lists the standard deviations for the corrected and raw data as well as the improvement due to the correction or correction rate. According to the table, the highest correction rate occurs at 1 m, and the correction at 2 m is less than that at 1 m. In addition, there is almost no correction at 3 m. Experiments were performed in the rotation range 0° - 90°, as mentioned in Section 4.2.

The standard deviation is a measure of how close the correction data are to the frontal standard. Fig. 12 shows the standard deviation of the experimental results. From left to right, the data correspond to the 1 m attention stance, 1 m hands-up stance, 2 m attention stance, 2 m hands-up stance, 2 m hands-half-up stance, and 3 m attention stance. The standard deviation of correction data is reduced because the data are corrected close to the frontal standard. The most-corrected states are the 1 m hands-up and 2 m hands-half-up stances.

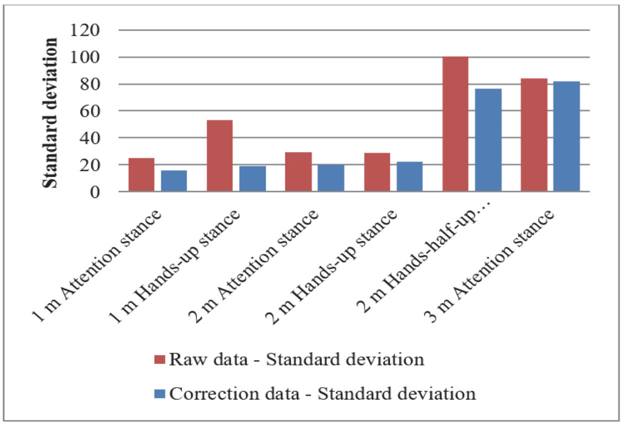


Figure 12 Standard deviation of experimental results

The 1 m hands-up stance shows the highest correction rate among the experimental results. In this stance, the average error decreases by 52% from approximately 36.47 to approximately 17.15, and the standard error decreases by 42% from approximately 7.7 to approximately 4.4. The 2 m hands-half-up stance shows the lowest correction rate among the experimental results. In this stance, the average error rate is 31%, corresponding to a correction rate of 23% with respect to the raw-data value of 40.7. Although the error rate of 31% is acceptable for applications that do not require accurate interaction, such as media art, it is insufficient for applications such as games, which require highly accurate input.

4.4 Error Analysis

Fig. 13 and Fig. 14 show the standard and average error rates, respectively. The error rate is the lowest at 6.96 in the 1 m attention stance and the highest at 31.1 in the 2 m hands-half-up stance. At distances greater than 3 m, which are beyond the recognition range of the device, both the raw and correction values are 37.66, and the proposed method did not correct the data. For simple gestures at 1 m and 2 m, it is possible to obtain meaningful data for motion recognition with an accuracy of 80% - 90%. However, for complicated motion at distances greater than 2 m, there is a possibility that motion recognition is not performed correctly, because the accuracy is 69%.

According to a related study on rotation calibration using an SVM [44], the accuracy is approximately 40% in the first attempt and approximately 71% after 50 - 60 iterations of machine learning. In the proposed method, the initial accuracy in the 2 m hands-half-up stance is 69%, which is similar to the accuracy of 71% in the results obtained after approximately 50 iterations of machine learning in the related study. However, the final correction rate is lower by approximately 10% because the proposed method does not involve post-processing.

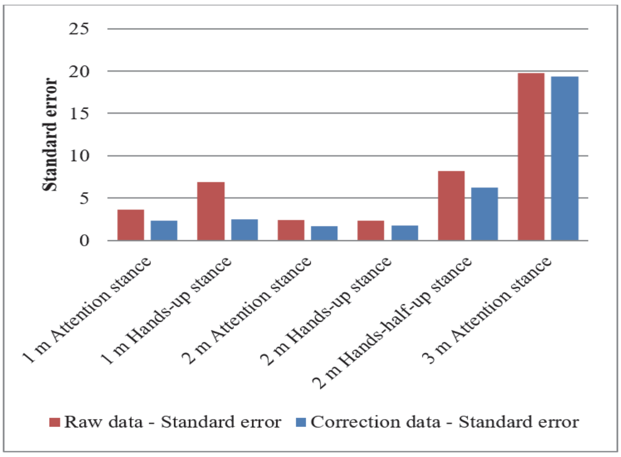


Figure 13 Standard error of experimental results

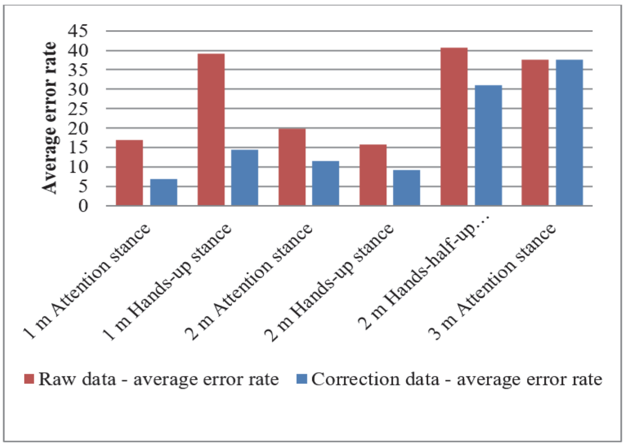


Figure 14 Average error rate of experimental results

Table 2 Experimental results obtained using the proposed method for the following parameters: standard deviation, standard error, and average error rate

Distanceand posture	Correction data	Raw data	Improvement rate / %
1 m attention stance	16.09	25.06	35.79
1 m hands-up stance	19.24	53.2	63.83
2 m attention stance	20.39	29.16	30.08
2 m hands-up stance	22.56	28.76	21.56
2 m hands-half-up stance	76.37	100.09	23.7
3 m attention stance	81.95	83.95	2.38
1 m attention stance	2.32	3.62	35.91
1 m hands-up stance	2.48	7.71	63.9
2 m attention stance	1.67	2.39	30.13
2 m hands-up stance	1.8	2.30	21.74
2 m hands-half-up stance	6.26	8.20	23.66
3 m attention stance	19.32	19.79	2.37
1 m attention stance	6.96	17	59.06
1 m hands-up stance	14.42	39.14	63.16
2 m attention stance	11.43	19.75	42.13
2 m hands-up stance	9.2	15.75	41.59
2 m hands-half-up stance	31.1	40.72	23.62
3 m attention stance	37.66	37.66	0

5 CONCLUSION

In this paper, we introduced a rotation-correction method based on the depth-value symmetry of human skeletal joints in a single RGB-D camera system. The correction method utilizes the body specificity of the user and can easily be employed in RGB-D camera applications. The experimental results showed that the proposed framework is robust, with an overall accuracy of 85.38% and an average error rate of 14.62, which are better than the corresponding values obtained before the learning phase in most previous studies. However, the accuracy is 1.9% - 7% lower than those in related studies in which long-term machine learning was applied.

Our method is advantageous for initial value correction in the short term, but its accuracy is lower than that of machine learning, which utilizes long-term and cumulative data. Nevertheless, the absence of a long-term learning phase makes our method useful for applications such as media art, in which users are unspecified. In the future, we will attempt to combine our method with machine learning. Because the initial calibration value of our solution is higher than the initial value obtained in related studies based on machine learning, we believe that the length of the learning phase can be reduced by using machine learning in combination with our method. Such a combination can potentially be applied as a short-term machine learning solution in areas such as smart industries, in which users can store and accumulate data.

Acknowledgements

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (2019S1A5C2A04082405).

6 REFERENCES

[1] Apperley, T. H. (2013). The body of the gamer: Game art and gestural excess. *Digital Creativity*, *24*(2), 145-156. https://doi.org/10.1080/14626268.2013.808967

[2] Masalkar, P. J., Stachniak, S. P., Leyvand, T., Zhang, Z., Del Castillo, L., & Mathe, Z. (2014). In-home depth camera calibration. U.S. Patent 8,866,889 B2, Microsoft Corporation, Redmond, WA, USA.

[3] Wigdor, D. & Wixon, D. (2011). Brave NUI world: Designing natural user interfaces for touch and gesture. Elsevier. https://doi.org/10.1016/B978-0-12-382231-4.00002-2

[4] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., & Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, *56*(1), 116-124. https://doi.org/10.1145/2398356.2398381

[5] Stork, A. (2015). Visual computing challenges of advanced manufacturing and Industrie 4.0. *IEEE Computer Graphics and Applications*, *35*(2), 21-25. https://doi.org/10.1109/MCG.2015.46

[6] YS Tech Share (2019). Kinect Pixel Mirror. http://tech.yeesiang.com/kinect-pixel-mirror/. Accessed 16 November 2019.

[7] Lee, H. Y., Kim, J. Y., & Lee, W. H. (2014). Interactive Digital Art based on user's Physical Effort with Sensor Technology. *Int. J. of Software Engineering & Its Applications*, *8*(3).

[8] Rodrigues, D. G., Grenader, E., Nos, F. D., Dall'Agnol, M. D., Hansen, T. E., & Weibel, N. (2013). MotionDraw: A tool for enhancing art and performance using Kinect. *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, Paris, France, 1197-1202. https://doi.org/10.1145/2468356.2468570

[9] Chen, K. & Wong, S. (2014). Interactive sand art drawing using Kinect. *7th International Symposium on Visual Information Communication and Interaction*, Sydney, Australia, 78. https://doi.org/10.1145/2636240.2636846

[10] Lee, H. Y., Kim, J. Y., & Lee, W. H. (2013). Interactive digital art using sensor technology. *Science and Engineering Research Support Society*, *39*, 94-98. https://doi.org/10.14257/astl.2013.39.18

[11] Novak-Marcincin, J., Torok, J., Novakova-Marcincinova, L., Barna, J., & Janak, M. (2014). Use Of Alternative Scanning Devices For Creation Of 3D Models Of Machine Parts. *Tehnicki vjesnik/Technical Gazette*, *21*(1).

[12] Kim, J. Y. & Lee, W. H. (2014). Design and Modelling Immersive Game Contents System for Virtual Reality Technology. *Technology*, *4*(5), 6. https://doi.org/10.14257/astl.2014.46.49

[13] Riyad, A., Huang, J., & Michael, Y. (2012). Study on the use of Microsoft Kinect for robotics applications. *IEEE/ION Position*, *Location and Navigation Symposium*, Myrtle Beach, SC, USA, 1280-1288.

[14] Macknojia, R., Chavez-Aragon, A., Payeur, P., & Laganiere, R. (2013). Calibration of a network of Kinect sensors for robotic inspection over a large workspace. *IEEE Workshop on Robot Vision* (*WORV*), Clearwater, FL, USA, 184-190. https://doi.org/10.1109/WORV.2013.6521936

[15] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., & Fitzgibbon, A. (2011). Kinect Fusion: Real-time dense surface mapping and tracking. *International Symposium on Mixed and Augmented Reality*, Basel, Switzerland, 127-136. https://doi.org/10.1109/ISMAR.2011.6092378

[16] Jan, S., Jancosek, M., & Pajdla, T. (2013). 3D with Kinect. Consumer Depth Cameras for Computer Vision, UK, 3-25. Springer: London. https://doi.org/10.1007/978-1-4471-4640-7_1

[17] Kim, J. Y. & Lee, W. H. (2018). Walking simulation for VR game character using remote sensoring device based on AHRS-motion recognition. *Ieee Access*, *7*, 19423-19434. https://doi.org/10.1109/ACCESS.2018.2878237

[18] Lim, K. M., Tan, A. W. C., & Tan, S. C. (2016). A feature covariance matrix with serial particle filter for isolated sign language recognition. *Expert Systems with Applications*, *54*, 208-218. https://doi.org/10.1016/j.eswa.2016.01.047

[19] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., & Fitzgibbon, A. (2011). Kinect Fusion: Real-time 3D

reconstruction and interaction using a moving depth camera. *24th Annual ACM Symposium on User Interface Software and Technology*, Santa Barbara, CA, USA, 559-568. https://doi.org/10.1145/2047196.2047270

[20] Kim, J. H. & Park, M. (2018). Visualization of concrete slump flow using Kinect sensor, *Sensors*, *18*(3), 771. https://doi.org/10.3390/s18030771

[21] Bernacchia, N., Scalise, L., Casacanditella, L., Ercoli, I., Marchionni, P., & Tomasini, E. P. (2014). Non contact measurement of heart and respiration rates based on Kinect. *IEEE International Symposium on Medical Measurements and Applications* (*MeMeA*), Lisboa, Portugal, 1-5. https://doi.org/10.1109/MeMeA.2014.6860065

[22] Yao, L., Yang, W., & Huang, W. (2019). An Improved Feature-Based Method for Fall Detection. *Tehnički vjesnik*, *26*(5), 1363-1368. https://doi.org/10.17559/TV-20190411015902

[23] Han, J., Shoo, L., Xu, D., & Shotton, J. (2013). Enhanced computer vision with Microsoft Kinect sensor: A review. *IEEE Transactions on Cybernetics*, *43*(5), 1318-1334. https://doi.org/10.1109/TCYB.2013.2265378

[24] Zhu, M., Huang, Z., Ma, C., & Li, Y. (2017). An objective balance error scoring system for sideline concussion evaluation using duplex Kinect. *Sensors*, *17*(10), 2398. https://doi.org/10.3390/s17102398

[25] Pezzuolo, A., Guarino, M., Sartori, L., & Marinello, F. (2018). A feasibility study on the use of a structured light depth-camera for three-dimensional body measurements of dairy cows in free-stall barns. *Sensors*, *18*(2), 673. https://doi.org/10.3390/s18020673

[26] Chen, C., Yang, B., Song, S., Tian, M., Li, J., Dai, W., & Fang, L. (2018). Calibrate multiple consumer RGB-D cameras for low-cost and efficient 3D indoor mapping. *Remote Sensing*, *10*(2), 328. https://doi.org/10.3390/rs10020328

[27] Liao, Y., Sun, Y., Li, G., Kong, J., Jiang, G., Jiang, D., Cai, H., Ju, Z., Yu, H., & Liu, H. (2017). Simultaneous calibration: A joint optimization approach for multiple Kinect and external cameras. *Sensors*, *17*(7), 1491. https://doi.org/10.3390/s17071491

[28] Pagliari, D., Menna, F., Roncella, R., Remondino, F., & Pinto, L. (2014). Kinect Fusion improvement using depth camera calibration. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*(5), 479. https://doi.org/10.5194/isprsarchives-XL-5-479-2014

[29] Pagliari, D. & Pinto, L. (2015). Calibration of Kinect for Xbox One and comparison between the two generations of Microsoft sensors. *Sensors*, *15*(11), 27569-27589. https://doi.org/10.3390/s151127569

[30] Auvinet, E., Multon, F., & Meunier, J. (2015). New lower-limb gait asymmetry indices based on a depth camera. *Sensors*, *15*(3), 4605-4623. https://doi.org/10.3390/s150304605

[31] Patsadu, O., Nukoolkit, C., & Watanapa, B. (2012). Human gesture recognition using Kinect camera. *International Joint Conference on Computer Science and Software Engineering*, Bangkok, Thailand, 2012, 28-32. https://doi.org/10.1109/JCSSE.2012.6261920

[32] Monir, S., Rubya, S., & Ferdous, H. S. (2012). Rotation and scale invariant posture recognition using Microsoft Kinect skeletal tracking feature. *International Conference on Intelligent Systems Design and Applications*, Kochi, India, 404-409. https://doi.org/10.1109/ISDA.2012.6416572

[33] Almeida, S. G., Guimarães, F. G., & Ramírez, J. A. (2014). Feature extraction in Brazilian sign language recognition based on phonological structure and using RGB-D sensors, *Expert Systems with Applications*, *41*(16), 7259-7271. https://doi.org/10.1016/j.eswa.2014.05.024

[34] Uebersax, D., Gall, J., Van den Bergh, M., & Gool, L. V. (2011). Real-time sign language letter and word recognition from depth data. *IEEE International Conference on Computer Vision Workshops* (*ICCV Workshops*), Barcelona, Spain, 383-390. https://doi.org/10.1109/ICCVW.2011.6130267

[35] Lun, R. & Zhao, W. (2015). A survey of applications and human motion recognition with Microsoft Kinect. *International Journal of Pattern Recognition and Artificial Intelligence*, *29*(5), 1555008. https://doi.org/10.1142/S0218001415550083

[36] GallostraAcín, J. (2017). *Reconstruction of scenes using a hand-held range imaging camera*. Bachelor's Thesis, Escola Tècnica Superior d'Enginyeria Industrial de Barcelona, Barcelona, Spain.

[37] Di, K., Zhao, Q., Wan, W., Wang, Y., & Gao, Y. (2016). RGB-D SLAM based on extended bundle adjustment with 2D and 3D information. *Sensors*, *16*(8), 1285. https://doi.org/10.3390/s16081285

[38] Tang, S., Zhu, Q., Chen, W., Darwish, W., Wu, B., Hu, H., & Chen, M. (2016). Enhanced RGB-D mapping method for detailed 3D indoor and outdoor modeling. *Sensors*, *16*(10), 1589. https://doi.org/10.3390/s16101589

[39] Antensteiner, D., Štolc, S., & Pock, T. (2018). A review of depth and normal fusion algorithms. *Sensors, 18*(2), 431. https://doi.org/10.3390/s18020431

[40] Lulu, R., Kurhade, V., Pande, N., Gaidhane, R., Khangar, N., Chawhan, P., & Khaparde, A. (2015). Web camera based mouse controlling using RGB colours. *International Journal of Advanced Research in Computer Science and Engineering*, *4*(2), 372-375.

[41] Rougier, C., Auvinet, E., Meunier, J., Mignotte, M., & De Guise, J. A. (2011, August 30–September 3). Depth energy image for gait symmetry quantification. Annual *International Conference of IEEE Engineering in Medicine and Biology Society*, Boston, USA, 5136-5139. https://doi.org/10.1109/IEMBS.2011.6091272

[42] Wasenmüller, O. & Stricker, D. (2016). Comparison of Kinect v1 and v2 depth images in terms of accuracy and precision. *Asian Conference on Computer Vision International Workshops*, Taipei, Taiwan, 34-45.

[43] Kim, J., Hasegawa, T., & Sakamoto, Y. (2016). Hazardous object detection by using Kinect sensor in a handle-type electric wheelchair. *Sensors*, *17*(12), 2936. https://doi.org/10.3390/s17122936

[44] Kumar, P., Saini, R., Roy, P. P., & Dogra, D. P. (2018). A position and rotation invariant framework for sign language recognition (SLR) using KINECT. *Multimedia Tools and Applications*, 77(7), 8823-8846. https://doi.org/10.1007/s11042-017-4776-9

[45] Papadopoulos, G. T., Axenopoulos, A., & Daras, P. (2014). Real-time skeleton-tracking-based human action recognition using Kinect data. *International Conference on Multimedia Modeling*, Dublin, Ireland, 473-483. https://doi.org/10.1007/978-3-319-04114-8_40

Contact information:

Sunghyun KIM, PhD Candidate
Department of Multimedia Art & Technology,
The Graduate School of Advanced Imaging Science,
Chung-ang University, Seoul 06974, Republic of Korea
E-mail: kimkshgugu@naver.com

Won-Hyung LEE, PhD, Professor
(Corresponding author)
Department of Multimedia Art & Technology,
The Graduate School of Advanced Imaging Science,
Chung-ang University, Seoul 06974, Republic of Korea
E-mail: whlee@cau.ac.kr