

RESEARCH

Open Access



Infrared bundle adjusting and clustering method for head-mounted display and Leap Motion calibration

San Park¹, Seoungjae Cho², Jisun Park², Kaisi Huang², Yunsick Sung² and Kyungeun Cho^{2*} 

*Correspondence:

cke@dongguk.edu

²Department of Multimedia

Engineering, Dongguk

University-Seoul, 26

Pildong-ro 1-gil, Jung-gu,

Seoul 04620, Republic

of Korea

Full list of author information

is available at the end of the

article

Abstract

Leap Motion has become widely used due to its ability to recognize intuitive hand gestures or accurate finger positions. Attaching a Leap Motion to a virtual reality head-mounted display (VR HMD) is highly interoperable with virtual objects in virtual reality. However, it is difficult for a virtual reality application to identify the accurate position where the Leap Motion is attached to the HMD. This causes errors in the positions of the actual user's hands and the virtual hands, which makes the interaction in virtual reality difficult. In this paper, a method that calibrates an output area in VR HMD and a sensing area in Leap Motion is proposed. The difference in the origin coordinate between VR HMD and Leap Motion is derived using the proposed method. The position of the Leap Motion attached to the HMD was determined through an experiment using the proposed calibration technique, and the error was approximately 0.757 cm. Accordingly, it enables more intuitive interactions in virtual reality applications.

Keywords: Head-mounted display, Virtual reality, Natural user interface, Multi-sensor calibration, Bundle adjustment

Introduction

Leap Motion (LM) is a sensor device that recognizes the user's fingers. It offers excellent interoperability with the head mounted display (HMD) [1] and is thus frequently used while attached to one. The user's immersive experience with multiple content items in three-dimensional (3D) virtual reality [2] can be enhanced by a method in which LM is attached to the front of the HMD [3–5] is used to recognize the 3D locations of the user's hands [6] and the results are outputted to the HMD [7]. This method requires accurate input and output in the x, y, and z axes to interact with the 3D virtual reality environment [8, 9].

However, when LM is attached to a VR HMD, which shows videos as a scene view, it is difficult to apply LM calibration approaches used in optical see-through HMD. This is because the output display of the VR HMD is a VR environment, which is different from a real environment owing to the characteristics of the VR HMD. A user who wears a VR HMD in front of eyes cannot see the LM and his/her hands. Thus, intuitively interacting with objects becomes difficult in a virtual environment. In particular, when the virtual

finger 3D model is visualized in a VR environment, the user experiences awkwardness when interacting with virtual objects using his/her hands because the virtual finger position does not coincide with the actual finger position [10].

Because virtual reality applications do not obtain the position of the LM accurately, a method of determining the relative position of the VR HMD and the LM is required to show the exact position of the user hands on the VR HMD screen. If a device that simultaneously performs the functions of the LM and the VR HMD is manufactured, the relative 3D coordinates between the LM and the VR HMD can be grasped by pre-calibration [11]. However, this approach incurs high manufacturing costs and thus cannot be applied to the existing commonly used devices. Thus, another approach is needed that can accurately estimate the relative 3D location a VR HMD display and a LM by calibrating the 3D location relationship between the VR HMD display and the LM [12].

In this paper, a technique that can match the output region in the VR HMD display and detection region in LM by calibrating the relative 3D location between VR HMD and LM is proposed. The difference in the origin coordinates between VR HMD and LM is thereby derived and can be applied to content using VR HMD and LM. This allows the user to interact more intuitively within the virtual reality through the virtual hands moving the same as the motion of the real hands.

The remainder of this paper organized as follows “[Related work](#)” section provides an overview of the previous studies related to VR HMD-LM calibration methods. “[Head-mounted display and Leap Motion calibration method](#)” section explains how the proposed system works. Finally, in “[Experimental evaluation](#)” section, the 3D location of LM attached to the HMD is measured, and the relative 3D location is identified via a program created using the proposed system. The performance of the proposed system is then evaluated using the difference between the distance measured by the program and the actual distance.

Related work

Qian et al. [13] proposed a method using a 3D marker to calibrate the space of a holographic image and the range of motion recognition sensor. In that study, the calibration between the motion recognition sensor and display viewed by the user in the 3D space was performed using a method in which the user adjusted a virtual reticle—a net of guidelines displayed in the HMD screen—in relation to a 3D marker. This method may incur a variation in accuracy based on the characteristics of users. Meanwhile, an interaction method that adjusts the virtual reticle may also be inconvenient to users.

Moser et al. [14] proposed a user calibration technique that aligns a finger or stick recognized by a LM to a reticle, which visible on the HMD video. Using this technique, the 3D location relationship between the optical see-through HMD and the user’s hand can be identified. In that paper, however, when a test was conducted without calibrating the 3D location between HMD and LM, each user had to individually perform the calibration, and the error range was very large when the number of tests was small.

Zhang et al. [15] proposed a method of calibrating the position of the eye viewed through the optical see-through HMD [16] and the finger position recognized by a LM by attaching an eye recognition camera to the HMD. In this method, errors from the eyeball tracking camera can affect the calibration performance. Thus, the eyeball tracking camera must be

attached inside the HMD when applied to the Virtual Reality (VR) HMD. This approach is difficult for the users to employ. Furthermore, since the accuracy of the calibration varies according to the manner in which the HMD is attached, or the characteristics of the user's eyes, calibration must again be performed when the user changes.

The above methods that calibrate the 3D location between the optical see-through HMD and LM can be performed in an augmented reality (AR) environment based on the real environment. This approach is more appropriate when employing device characteristics that do not require external views to be reflected in the user's eyes [10, 14]. Additionally, the HMD display must be attached closely around the user's eyes so the user cannot peripherally view outside the HMD. Thus, a method is required to calibrate and measure the relative 3D location between the VR HMD and LM [17]. For example, a method that can identify a relative 3D location through sensing the HMD where LM is attached can be applied using external sensors.

Various image processing techniques are used to extract features from images [18, 19]. The bundle adjustment algorithm (BAA) [20, 21] finds common feature points among images after several two-dimensional images are inputted by cameras. It then tracks a distance using moving 2D locations of all feature points according to the image progress. However, the BAA derives a cloud of all 3D points of common feature points with regard to all points in the two-dimensional (2D) camera image. Therefore, when images obtained by LM and HMD become a 3D point cloud, point clouds of HMD and LM as well as point clouds of all features are derived. Thus, it is difficult to find an accurate origin coordinate of HMD and LM. A method that finds the 2D location of HMD and LM inside the 2D camera image is needed. Moreover, a method that can estimate the 2D location of HMD and LM inside the 2D camera image can be applied after an Infrared (IR) LED shared by HMD and LM is inputted using an infrared sensor.

In addition, a study was conducted to make each point into a single binary large object (BLOB) using a pixel value in each point in the image [22]. The above study changed a mostly grayscale image into a binary image, which was then compared with the surrounding pixels. However, the above method incurred many errors in images in which the grayscale values had a gradient structure.

A circle detection method using the Hough filter method [23] was studied by employing color images to solve the grayscale problem. This circle detection method uses the change in gradient between a pixel and a subsequent pixel using the Hough filter. However, the accuracy of this method is degraded if a circle shape is not accurately estimated.

In this paper, a calibration technique is proposed using color and IR images. The scientific contribution of the proposed technique is summarized in pointwise as follows:

1. Obtaining user-independent results, since this technique does not track user eyes or require user adjustments.
2. Proposing a precise calibration method between video see-through HMD and LM, which has not been studied yet.

Head-mounted display and Leap Motion calibration method

In this paper, color and IR images of the VR HMD and LM are acquired using a color-IR camera to estimate the position of the VR HMD and LM attached to the device. Figure 1 depicts the methodological flow of the proposed system.

Color and IR images should be acquired at the same time and taken at least three different positions. This paper utilizes Microsoft Kinect to simultaneously acquire color and IR images. Figure 2 shows an example of Kinect capturing the LM-attached VR HMD.

The Bundle Adjuster estimates the positional relationship between each pixel of the obtained color image. Images of VR HMD and LM with IR camera are obtained and it is proceeded to the proposed method, it is possible to estimate the 3D positional relationship of all the pixels of the image as one coordinate system.

On the other hand, the BLOB detector estimates the coordinates of the IR LEDs shown in IR images. If the IR pixel coordinates are classified into the HMD group and the LM group, and the IR LED coordinates are grouped by the number of IR LEDs, the clusters of all the IR LEDs can be estimated from the IR image. Finally, it is determined which equipment each IR LED cluster corresponds to. The 3D positions of HMD and LM can be expressed on one coordinate system.

Using the proposed method, three or more color and IR images can be used to represent the position of the VR HMD and LM on a single 3D coordinate system. The relative position of the LM attached to the HMD can be measured by substituting the physical position difference of the IR LEDs in a device into the 3D coordinate system.

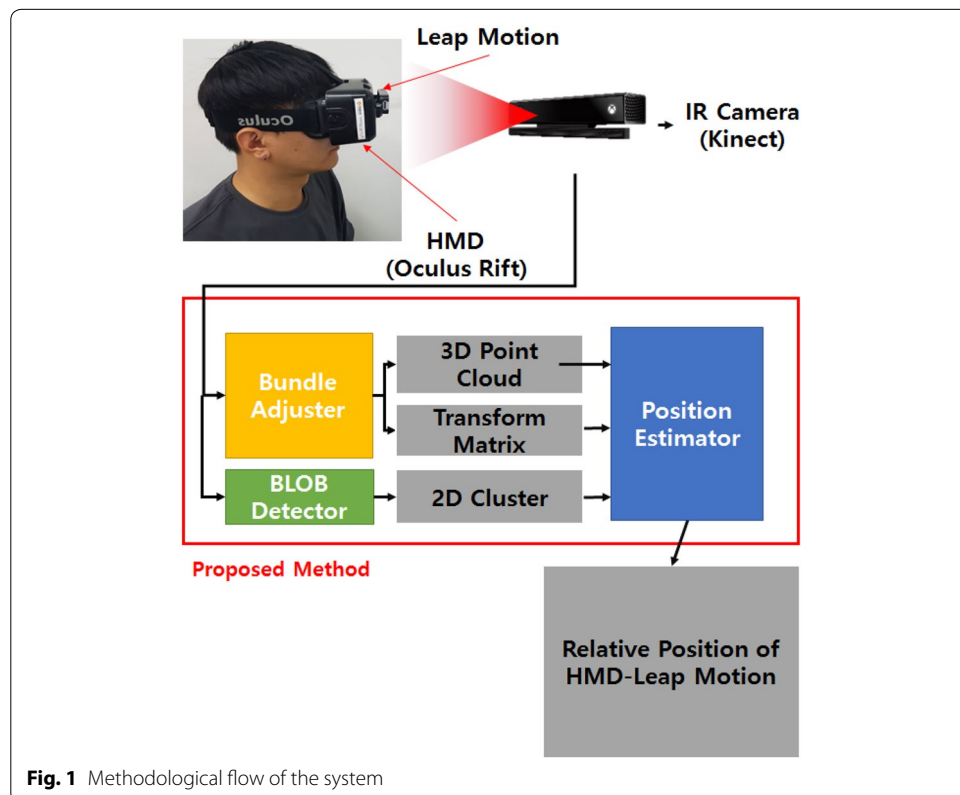


Fig. 1 Methodological flow of the system

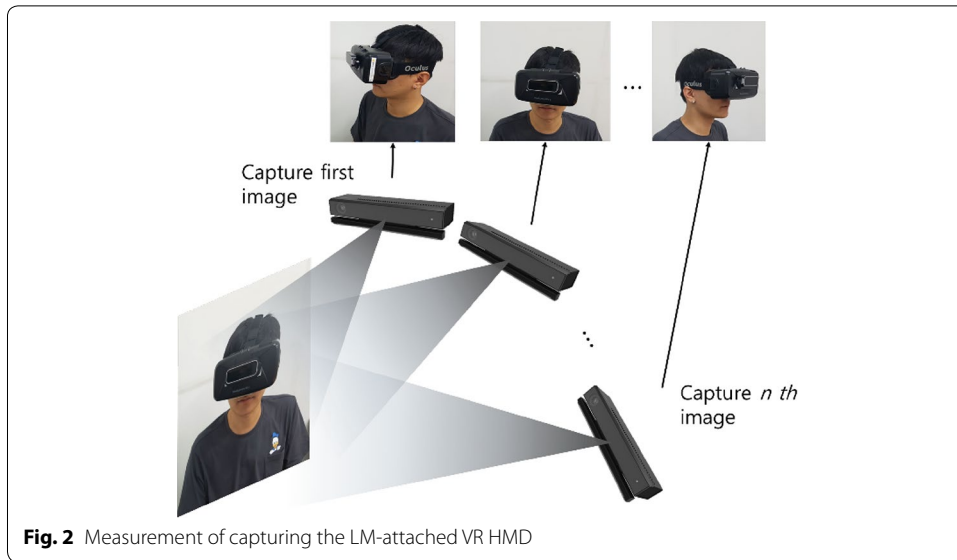


Fig. 2 Measurement of capturing the LM-attached VR HMD

Estimation of image-pixel relative 3D location using the bundle adjuster

The bundle adjuster uses BAA to derive 3D locations of all pixels from the gathered images. The BAA results are a point cloud and transform matrices of the same number as those of the image frames. Consequently, the space coordinate of the 2D image pixel can be known.

The bundle adjuster is operated as follows. Infrared images of HMD where LM is attached are inputted. The infrared images are converted into a few series of sparse matrices. A 3D point cloud and transform matrices (the same number as infrared images) are estimated from the sparse matrices by the sparse bundle adjuster. When the projection matrix and 3D point cloud are multiplied, the 2D point coordinate that is projected onto the corresponding 2D image is generated.

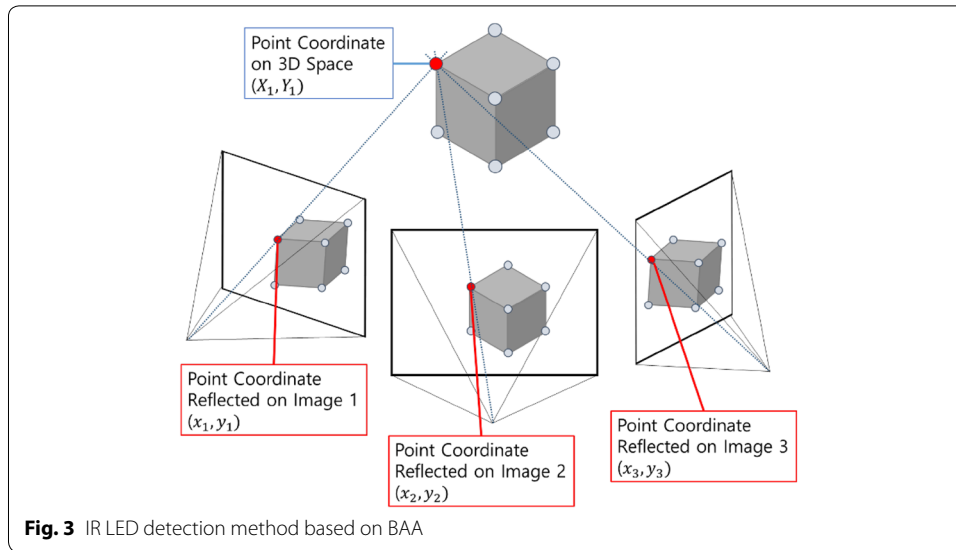
First, a pixel value of the obtained image is received to produce a sparse matrix, M_p , to reduce the computation amount of the bundle adjustment. Here, $i_{t,x,y}$ refers to a pixel value of the image acquired at time t . Pixel $i_{t,x,y}$ refers to a grayscale value in the range from 0 to 255. The calculated sparse matrix is defined as M_t . Assuming that α refers to the image width and β refers to the infrared image height, M_t has the size of $\alpha \times \beta$, and the element of the sparse matrix is defined as $m_{t,x,y}$.

In addition, threshold θ is a constant specified for comparison with each pixel. As specified in Eq. (1), if $i_{t,x,y}$ is less than threshold θ , which is determined by user testing, zero is assigned to $m_{t,x,y}$; otherwise, one is assigned to it.

$$f(i|\forall i, i \in I), f(i) = \begin{cases} 0, & i_{t,x,y} < \theta \\ 1, & i_{t,x,y} \geq \theta \end{cases} \quad (1)$$

A single point cloud $P = \{(x_0, y_0, z_0), (x_1, y_1, z_1), \dots\}$ and n transform matrices are calculated to acquire the 3D location of the image pixel after receiving n sparse matrices.

Once n infrared images and sparse matrices are inputted, 3D location coordinate is estimated based on the sparse matrices through the sparse bundle adjuster to configure point cloud P . Figure 3 shows the IR LED detection method based on BAA.



This paper is based on HMD images obtained at different points of sight. In Fig. 3, the circles in Figs. 1, 2 and 3 refer to image pixels. Here, common feature points are searched using a method that compares feature points between pixels with regard to all pixels. Once the feature points are found, 3D location $p(X, Y, Z)$, which minimizes the change value of all pixels inside the image, is calculated using the Levenberg–Marquardt optimization technique [24].

The bundle adjuster estimates the 3D space coordinate of image pixels after receiving a number of 2D color images. As a result, the point cloud and transform matrix are derived via the BAA.

Pixel matching using infrared BLOB detector

Although the bundle adjuster can estimate a 3D space coordinate from the 2D image pixel, it should be determined whether a pixel is on HMD or LM. In the BLOB detector, infrared images of HMD and LM are inputted, and pixels emitted from the IR LEDs of the device become point clouds. Accordingly, it can be determined whether each of the pixels in the 2D image is located in the infrared light of HMD and LM.

The BLOB detector generates a cluster of circle shapes using images of HMD and LM inputted via the infrared sensor. It then calculates the center point of the cluster and the distance to the farthest point from the center point. Three clusters whose distances are the farthest from the center point become the clusters of the infrared image where the IR LEDs of LM are located.

The BLOB detector receives n infrared images i_t and outputs cluster $G_{t,p}$, center point $c_{t,p}$, and $d_{t,p}$, which is the distance to the farthest point from the center point inside the cluster. The p -th cluster with regard to infrared image i_t obtained at time t is defined as $G_{t,p}$. $g_{t,p,q}$ refers to the q -th element in cluster $G_{t,p}$. The center point of $g_{t,p,q}$ is defined as $c_{t,p} = (c_{t,p,x}, c_{t,p,y})$. The distance to the point farthest from the center point is defined as $d_{t,p}$.

The BLOB detection algorithm is applied to derive the 2D locations of bulbs in the infrared image. In the images, the surrounding objects are removed based on infrared

scattering. A shape of BLOB is displayed as a circle referencing the part where the IR LED, the light source, is obtained. The Hough circle transform algorithm is used as an algorithm that extracts a circle shape from the image. Center point $c_{t,p} = (c_{t,p,x}, c_{t,p,y})$ of the circle, obtained thereafter, can be calculated via Eq. (2). Euclidean distance E_d is derived via Eq. (3), and distance $d_{t,p}$ to the point farthest from the center point is derived via Eq. (4).

$$c_{t,p} = \frac{1}{n} \left(\sum_{i=0}^n g_{t,p,i}, \sum_{j=0}^n g_{t,p,j} \right) \tag{2}$$

$$E_d(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \tag{3}$$

$$d_{t,p} = \max(E_d(c_{t,p}, p_1), E_d(c_{t,p}, p_2), \dots) \quad (p \in P) \tag{4}$$

Afterward, cluster $G_{t,p}$, center point $c_{t,p}$, and distance $d_{t,p}$, which were produced from the BLOB detector, as well as 2D point cloud P'_p are inputted, and point cloud P'_t is separately configured as cluster $G'_{t,p}$ when each point of point cloud P'_t is located inside element $g_{t,p}$. Accordingly, it is determined whether the estimated 2D point cloud P'_t corresponds to a 2D IR LED location. The cluster that collects 2D point cloud P'_t belonging within cluster $G_{t,p}$ is defined as $G'_{t,p}$. If the Euclidean distance of center point $c_{t,p}$ obtained from the BLOB detector and the acquired 2D location coordinate $p'_t(x'_t, y'_t)$ —calculated via Eq. (3)—is less than $d_{t,p}$, the 2D location coordinate is included in cluster $G'_{t,p}$. Cluster $G'_{t,p}$, which includes 2D point cloud P' , is outputted.

The infrared clustering can be achieved with infrared images using the BLOB detector. In the 2D color images, because locations of HMD and LM cannot be determined, clustering based on infrared images is conducted to determine the location of the device inside the image.

Detection of relative 3D location between HMD and LM using location estimator

The point cloud in the 3D space coordinate system of the 2D color image pixels is estimated by the bundle adjuster. The 2D locations of the IR LEDs in a 2D infrared image is determined by the BLOB detector. Then, 3D locations of HMD and LM are estimated by finding the 3D locations of the IR LEDs over the 3D space coordinate through the location estimator. Using the size of the infrared cluster, it can be determined whether the infrared cluster is corresponding to an IR LED cluster of HMD or that of LM. The device shapes can be estimated by determining the 3D locations of the IR LEDs of devices.

Once the coordinate of the 2D points with regard to the coordinate of all 3D points is calculated, the BLOB detector calculates whether the 2D point coordinate is located inside the cluster of LM or HMD. The center points of the 3D point coordinates, which correspond to the 2D point coordinate inside the LM cluster, are calculated. This becomes the 3D location coordinate system of LM. Similarly, the center points of the 3D point coordinates, which correspond to 2D point coordinate inside the HMD cluster, are calculated. This becomes the 3D location coordinate system of HMD.

The location detector receives 3D point cloud P acquired from the 2D point detector and $G'_{t,p}$ produced from the BLOB detector. It then outputs the origin coordinate of LM, $P_L = (x_p, y_p, z_l)$, and the origin coordinate of HMD, $P_O = (x_o, y_o, z_o)$. The original 3D location of LM is defined as $P_L = (x_p, y_p, z_l)$, and the original 3D location of HMD is $P_O = (x_o, y_o, z_o)$. The 3D locations of the IR LEDs of HMD and LM are shown in Fig. 4.

The location detector receives 3D point cloud P , which is approximated in the sparse bundle adjuster, and n transform matrices. It then derives 2D point cloud P' after identifying the location relationship between 3D point cloud P and the infrared image. The 2D point cloud projected onto the infrared image is $P'_t = \{(x'_{t,0}, y'_{t,0}), (x'_{t,1}, y'_{t,1}), \dots\}$. It is then multiplied with all transform matrices in 3D point cloud P . Consequently, 2D point cloud P' , which is the 2D location of images formed on the 2D camera coordinate system, can be derived. When 3D point cloud P is multiplied with the transform matrices, P is first transformed into a column vector and then multiplied. The 3D point cloud P and 2D point cloud P' projected at P are thereby produced.

Furthermore, the 3D location of cluster $G'_{t,p}$ can be calculated using 3D point cloud P , 2D point cloud P' , and cluster $G'_{t,p}$, which is acquired in the BLOB detector. The mean of the point p coordinate value inside 3D point cloud P is calculated prior to the projection with regard to all points p' inside 2D point cloud P' projected at the 2D point detector. It is included in $G'_{t,p}$ and then substituted with center point $c'_{t,p}$ of $G'_{t,p}$. Once the 3D center point of the cluster is derived, original 3D location P_L of LM and original 3D location P_O of HMD are calculated.

The mean of the center points of the three clusters—whose radius $d'_{t,p}$ of cluster $G'_{t,p}$ is the largest—is set to the 3D location coordinate of an IR LED P_L of LM. The mean of the remaining center points in the cluster is set to a 3D location coordinate of an IR LED P_O of HMD. If the original 3D location of LM is moved to the original 3D location of the coordinate in HMD, the changed 3D location of LM becomes $P'_L(x_o - x_p, y_o - y_p, z_o - z_l)$. Figure 5 shows the derived 3D location of LM and HMD.

When a user employs the LM-attached HMD, the origin in the coordinate systems of LM and HMD becomes the same if the value of $P'_L(x_o - x_p, y_o - y_p, z_o - z_l)$ is used as the origin of the coordinate system in LM.

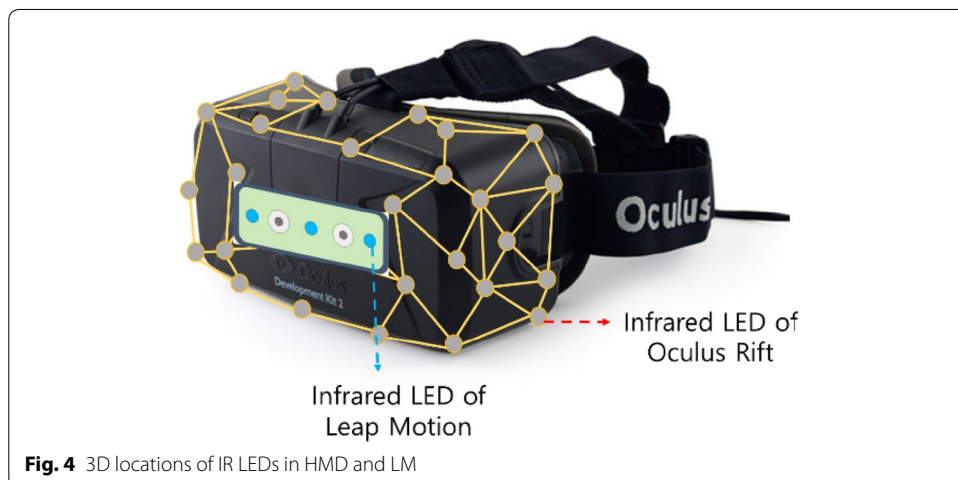
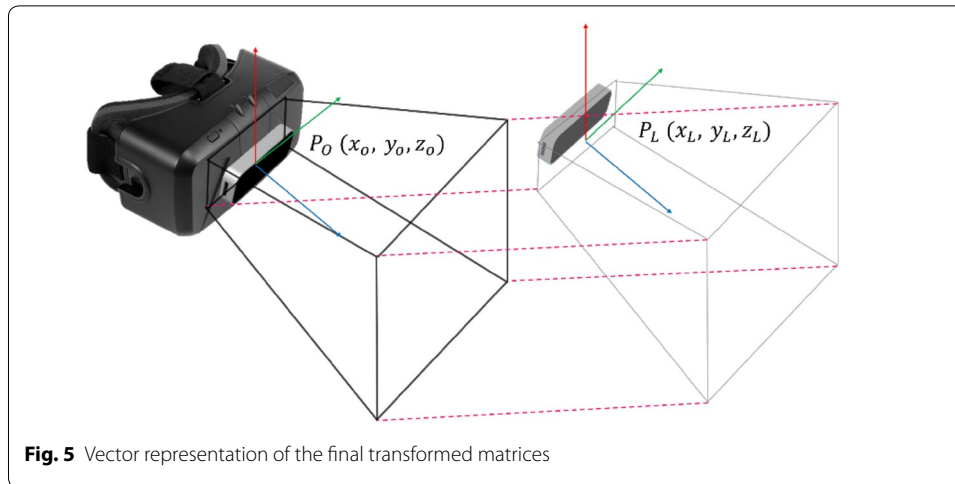


Fig. 4 3D locations of IR LEDs in HMD and LM



The location estimator can find the type of IR LED clusters and 3D location. Accordingly, the original 3D location of the device can be determined using the 3D locations of the IR LEDs of the device.

Experimental evaluation

An experiment was conducted to determine whether the calibration of the original 3D locations of LM and HMD can be used in the future as a pre-processing technique for content employing the LM-attached HMD. To this end, a calibration program was fabricated using the method proposed in this paper. To validate the method's performance, the error of results using the calibration technique was calculated using experimental and control groups to discern the physical difference between HMD and LM. The calibration of the optical see-through HMD has been studied a lot, but not for the video see-through HMD, so the results of this experiment could not be compared with the existing studies.

In this experiment, the location of the LM-attached HMD was set as the variable. That is, optical see-through HMDs that block the HMD display according to the LM location were not used. Instead, VR HMDs were used to conduct the experiment. Furthermore, a Kinect sensor was used as a color-IR camera, color to get the real image of HMD with LM and IR to identify the 2D IR LED locations on LM and HMD. For the experiment, the calibration of computers was directly conducted using Visual Studio 2013 (for Windows 8.1), and the specifications were an Intel i7-6700 3.40-GHz CPU, GeForce GTX 1060 GPU from NVidia, and 16 GB DDR5 2133 MHz of RAM. And the Oculus Rift DK2 is used as the HMD device. One man participated in the experiments as a subject, who is 29 years old is used to utilizes VR devices.

In this paper, HMD with LM is photographed using Kinect. Kinect is placed approximately 40 cm away from the HMD. Although the front and side of the HMD have IR LEDs on all sides, the front of the LM must be filmed during image acquisition using Kinect because the LM has IR LEDs only at the front. In addition, because color and IR images of the HMD and LM are taken at various angles and the feature points of each image are compared to estimate the 3D information, color and IR images are

taken at three or more positions at different positions. In this paper, at least 30 color and IR images were taken, and experiments were performed. The 3D coordinates obtained from the experiments were generated in a coordinate system parallel to the x , y , and z axes in the coordinate system of the estimated 3D point cloud. In the program for the experiment, the subject checks the IR images taken using Kinect and adjusts θ so that all the IR LEDs of the HMD and LM are visible.

The location coordinate of the origins in the Oculus Rift and LM is derived in centimeter units. Thus, if calibration was performed after moving the location of LM attached to the Oculus Rift and obtaining this with Kinect, it could measure how well the proposed calibration technique accurately revealed the performance to determine the accuracy of the 3D relative location coordinate of Oculus Rift and LM.

The program that implemented the proposed technique was performed in accordance with the following method. The IR LEDs embedded in the Oculus Rift and LM were obtained, and the center point of each device was estimated. The Kinect infrared sensor captured 40 IR LEDs of the Oculus Rift and three IR LEDs of LM attached to the Oculus Rift.

From that point, n infrared images were inputted to the program fabricated using the proposed calibration method. The pixel matrix generator stored n infrared images inputted from Kinect as a form of n pixel matrices. The sparse bundle adjuster transformed all n pixel matrices into sparse matrices to reduce the computation amount. Next, n sparse matrices were approximated in the 3D space using BAA, which calculated a single 3D point cloud. Then, a single 3D point cloud and n transform matrices were inputted, thereby deriving n 2D point clouds, whereby a 3D point cloud was projected on each infrared image. The BLOB detector clustered the infrared images.

As a result, it was determined whether clusters detected from n infrared images were a bulb cluster of LM or that of Oculus. A total of 43 2D point clusters, center points of 43 2D point clusters, and distances to 43 2D point clusters were calculated. The distance to the 2D point cluster referred to the distance to the farthest 2D point from the center point of the 2D point cluster.

Next, it was determined whether the 3D point cloud belonged to an IR LED area. The distances between the 43 2D point clusters and all points within n 2D point clouds were calculated. All points located inside the cluster were categorized into separate IR LED pixel clusters after the 2D point clusters of 43 2D point clouds, center points of 42 2D point clusters, and distances to 43 2D point clusters were inputted.

Finally, the location estimator calculated the origin coordinates of the IR LEDs of LM and Oculus Rift using the IR LED pixel clusters and 3D point cloud. In this way, origin locations of LM and Oculus Rift were outputted from n infrared images.

Since 3D locations of IR LEDs could be identified based on infrared images of LM and Oculus Rift in this paper, the relationship of actual locations between Oculus Rift and LM could be determined by estimating the actual sizes of Oculus Rift and LM. The locations of Oculus Rift where LM was attached were divided into 11 locations, as shown in Fig. 6. The eight positions of the LM on the Oculus Rift are top left, top center, top right, middle left, middle right, bottom left, bottom center, and bottom right. The three locations for testing the case where the Oculus Rift is not parallel to the frontal vector are the left, right, and upper sides. These 11 positions

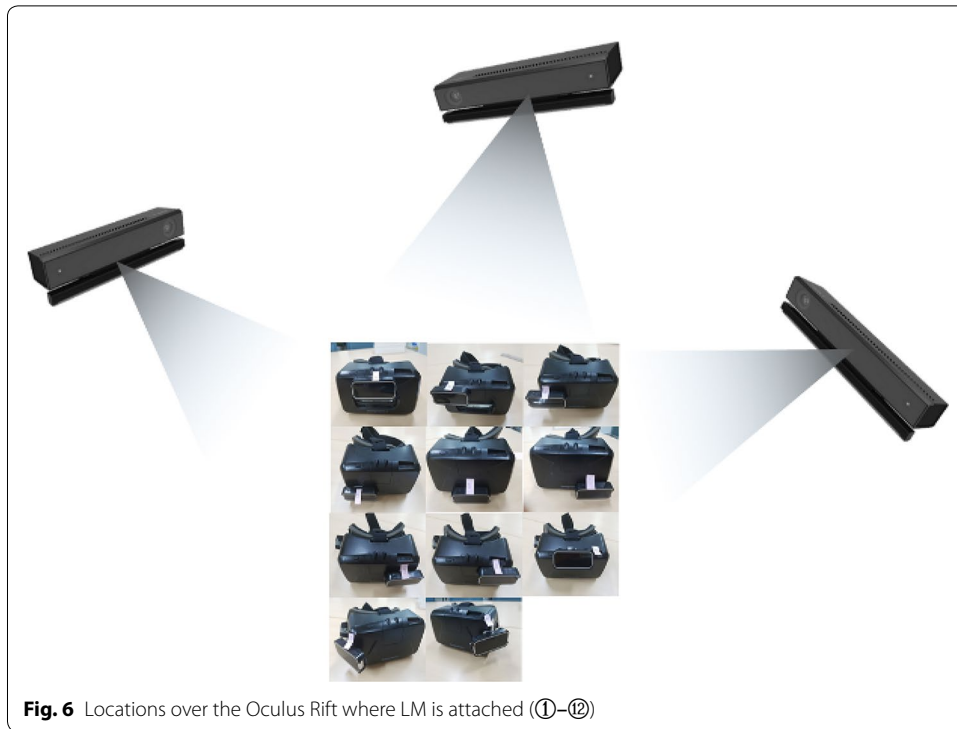
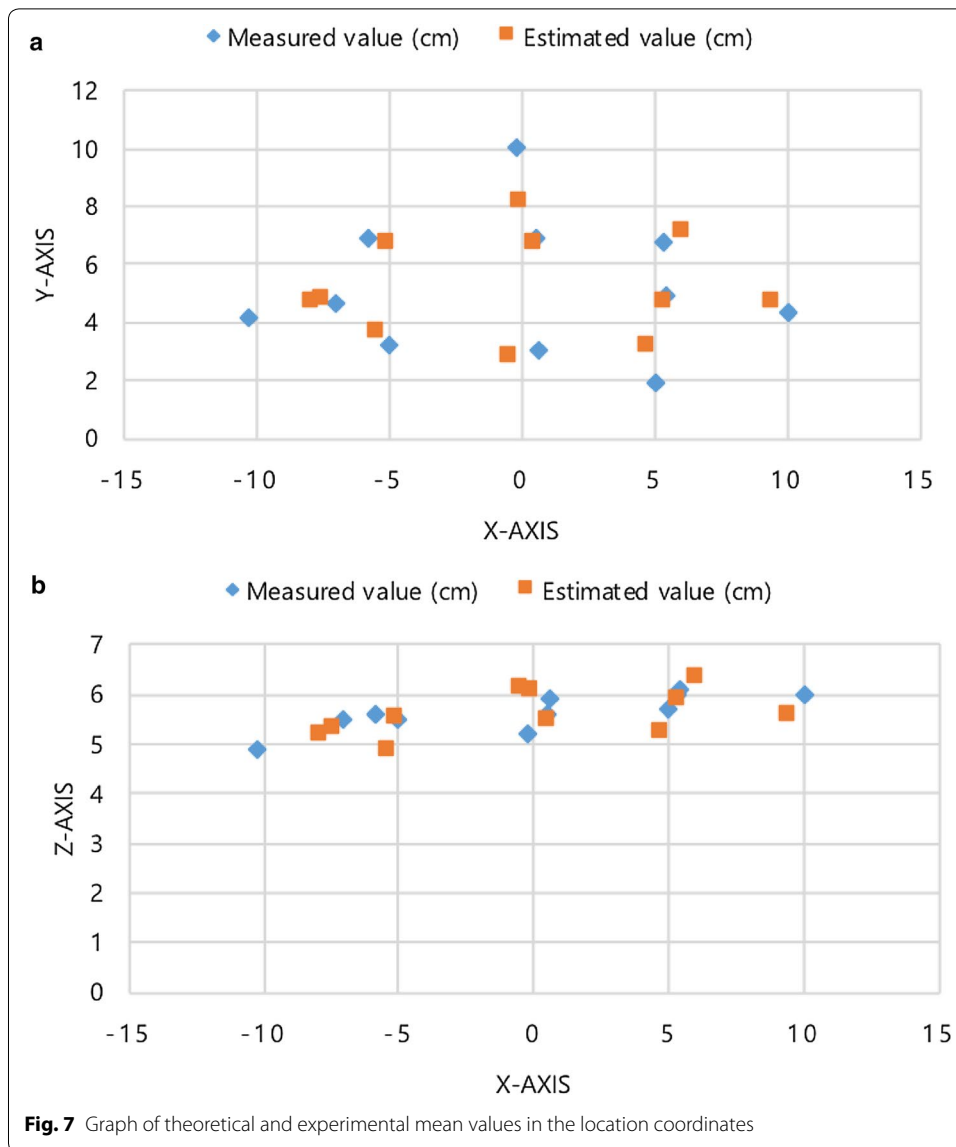


Table 1 Location differences of Oculus Rift and LM according to LM location and mean test values

Leap Motion location	Actual measured value (cm)			Estimated value (average, cm)		
	x	y	z	x	y	z
①	0.5	6.9	5.6	0.44	6.75	5.51
②	-5.8	6.9	5.6	-5.11	6.75	5.55
③	-7	4.6	5.5	-7.49	4.82	5.33
④	-5	3.2	5.5	-5.43	3.70	4.9
⑤	0.6	3	5.9	-0.48	2.82	6.16
⑥	5	1.9	5.7	4.72	3.23	5.27
⑦	5.4	4.9	6.1	5.31	4.7	5.91
⑧	5.3	6.8	6	6	7.16	6.36
⑨	-0.2	10.1	5.2	-0.09	8.21	6.07
⑩	-10.3	4.2	4.9	-7.93	4.72	5.22
⑪	10	4.4	6	9.38	4.77	5.6

cover the positions of all LMs that can be attached to the HMD. The calibration process was performed ten times for each location to validate the proposed method. The difference in locations between Oculus Rift and LM derived through the calibration was compared with that of the actual location to evaluate the proposed calibration technique.

Locations over the Oculus Rift where LM was attached are called ①–⑪. The differences in locations of Oculus Rift and LM through actual physical measurements and locations of Oculus Rift and LM derived through the calibration are presented in Table 1.



The differences in locations between the actual physical locations of Oculus Rift and LM, and locations derived through the experiment, are represented in the 3D coordinate system shown in Fig. 7.

In the figure, the x–y plane in Fig. 7a is a surface of the Oculus Rift. The x–z plane in Fig. 7b refers to the distance to the lens where the LM and Oculus Rift displays are outputted in the z-axis. The mean error distances in the respective x, y, and z coordinates with respect to all LM locations are 0.59 cm, 0.42 cm, and 0.34 cm.

The error rate was calculated using the difference between the actually measured value and the experimentally derived value. Specifically, it was calculated by dividing the difference of the experimental value and theoretical value (actual measured value) by the theoretical value.

The error distance of x, y, and z values in Experiments 9, 10, and 11 was larger than those of other results, as shown in Table 1. This was because locations of LM attached to

Table 2 Euclidean distance error and standard deviation between Oculus Rift and LM according to changes in LM location

Leap Motion location	Average distance error (cm)	Standard deviation
①	0.185	0.214
②	0.708	0.239
③	0.563	0.111
④	0.892	0.485
⑤	1.125	0.153
⑥	1.426	0.911
⑦	0.290	0.194
⑧	0.866	0.812
⑨	2.084	1.702
⑩	2.739	2.693
⑪	0.826	1.795

the Oculus Rift in Experiments 9 to 11 were not in parallel with the front of the Oculus Rift, which was different from those of Experiments 1 to 8. The results of the Euclidean distance error between the actual physical distance and the experimental results are listed in Table 2.

The actual physical distance and the average Euclidean distance derived from the experiment are about 1.064 cm. Of these, the average of the experimental errors at the 9, 10, and 11 locations is 3.734 cm, which is larger than the experimental average of 0.757 cm at the 1 to 8 locations. This was because the infrared ray emitted from the IR LEDs had different directions when an infrared image was captured.

The average Euclidean distance between the LM and Oculus Rift at locations 1 to 8 where the LM is attached is approximately 9.01 cm. Because the maximum error derived from this experiment is 0.757 cm, the accuracy of the proposed method is 91.6%. This result indicates that when the positional difference between the HMD and LM is calibrated using the proposed method, the position of the virtual and actual fingers in the VR environment is equal to 91.6%. Thus, the technique can be applicable to user content as a pre-processing technique.

Conclusion

The interaction method using LM over the VR HMD is expected to play an important role in enhancing reality for users. In this paper, a method to calibrate the difference in location between VR HMD and LM was proposed for enhancing the VR interaction. Experimental results revealed that error distances of 0.757 cm on average could be obtained in terms of location estimation of LM on the VR HMD surface. This value significantly reduced the positional difference between the actual user finger and that in the VR environment viewed by the user. Therefore, when a user interacts with the object in the VR environment, a feeling of immersion could be experienced.

The analysis in this paper found that the calibration error rate increased when LM was not attached in parallel with the VR HMD surface. This was because the infrared

ray emitted from the IR LEDs had different directions when an infrared image was captured. In future work, the error rate in this paper will be reduced by estimating the emission direction of the infrared ray using infrared stereo sensors in the infrared ray detection stage.

The results of this paper will be applicable to contents of virtual reality using a VR HMD. Moreover, they are expected to contribute to building a convenient user interaction environment in the augmented reality and virtual reality fields.

Authors' contributions

SP and KH performed the experiments and SP drafted the manuscript. SC and JP revised the manuscript. YS and KC provided full guidance. All authors read and approved the final manuscript.

Author details

¹ Research Group, Fxgear. Co. Ltd., 25, Hakdong-ro 46-gil, Gangnam-gu, Seoul 06099, Republic of Korea. ² Department of Multimedia Engineering, Dongguk University-Seoul, 26 Pildong-ro 1-gil, Jung-gu, Seoul 04620, Republic of Korea.

Acknowledgements

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Not applicable.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Funding

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2013-1-00684) supervised by the IITP (Institute for Information and communications Technology Promotion).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 17 August 2018 Accepted: 25 February 2019

Published online: 11 March 2019

References

- Ha T, Woo W (2006) Bare hand interface for interaction in the video see-through HMD based wearable AR environment. In: ICEC'06 Proceedings of the 5th international conference on entertainment computing, Cambridge, UK, September 20–22, pp 354–357
- Bang G, Yang J, Oh K, Ko I (2017) Interactive experience room using infrared sensors and user's poses. *J Inf Process Syst* 13(4):876–892
- Mentis HM, O'Hara K, Gonzalez G (2015) Voice or Gesture in the Operating Room. In: Computer-human interaction extended abstracts (CHI EA'15), Seoul, Korea, April 18–23, pp 773–780
- Park G, Choi H, Lee U, Chin S (2017) Virtual figure model crafting with VR HMD and Leap Motion. *Imaging Sci J* 65(6):358–370
- Vosinakis S, Koutsabasis P (2018) Evaluation of visual feedback techniques for virtual grasping with bare hands using Leap Motion and Oculus Rift. *Virtual Reality* 22(1):47–62
- Bruder G, Steinicke F, Sturzlinger W (2013) To touch or not to touch? Comparing 2D touch and 3D mid-air interaction on stereoscopic tabletop surfaces. In: *Spatial User Interaction '13 Proc. the 1st symposium on Spatial*, Los Angeles, California, USA, July 20, pp 9–16
- Han J, Gold N (2014) Lessons learned in exploring the Leap Motion sensor for gesture-based instrument design. In: *International conference on new interfaces for musical expression*, London, UK, June 30–July 4, pp 371–374
- Lin J, Guo X, Shao J, Jiang C, Zhu Y, Zhu S (2016) A Virtual reality platform for dynamic human-scene interaction. In: *SA '16 SIGGRAPH ASIA 2016 virtual reality meets physical reality: modelling and simulating virtual humans and environments*, Article No. 11, Macau, December 05–08, pp 1–4
- Bergh MV, Carton D, De Nijis R (2011) Real-time 3D Hand gesture interaction with a robot for understanding directions from humans. In: *IEEE Int. symposium on robot and human interactive communication (RO-MAN)*, Atlanta, GA, USA, July 31–August 3, pp. 357–362
- Jun H, Kim G (2016) A Calibration method for optical see-through head-mounted displays with a depth camera. In: *IEEE Virtual reality (VR)*, Greenville, SC, USA, March 19–23, pp 103–111

11. Colaco A (2013) Sensor design and interaction techniques for gestural input to smart glasses and mobile devices. In: The adjunct publication of the 26th annual ACM symposium on User interface software and technology (UIST'13), St. Andrews, UK, October 8–11, pp 49–52
12. Plopski A, Kiyokawa K, Nitschke C (2014) Conreal imaging in localization and HMD interaction. In: IEEE Int. Symposium on mixed and augmented reality (ISMAR), September 10–12, pp 397–400
13. Qian L, Azimi E, Kazanzides P, Navab N (2017) Comprehensive tracker based display calibration for holographic optical see-through head-mounted display. In: IEEE Int. symposium on mixed and augmented reality (ISMAR), March 16, pp 1–8
14. Moser KR, Swan II JE (2016) Evaluation of user-centric optical see-through head-mounted display calibration using a Leap Motion controller. In: 3D User interfaces (3DUI), IEEE symposium on, March 19–20, pp 159–167
15. Zhang Z, Weng D, Liu Y, Wang Y (2016) A Modular Calibration Framework for 3D Interaction system based on optical see-through head-mounted displays in augmented reality. In: 2016 Int. conf. on virtual reality and visualization, September 24–26, pp 393–400
16. Rolland JP, Holloway RL, Fuchs H (1995) A comparison of optical and video see-through head-mounted displays. In: SPIE, vol 2351 Telemanipulator and Telepresence Technologies, December 21, pp 293–307
17. Hinckley K, Pausch R, Goblel JC, Kassell NF (1994) A survey of design issues in spatial input. In: User interface software and technology (UIST) '94, Proc. of the 7th annual ACM symposium on user interface software and technology, Marina del Rey, California, USA, November 02–04, pp 213–222
18. Jun J-H, Kim M-J, Jang Y-S, Kim S-H (2017) Fire detection using multi-channel information and gray level co-occurrence matrix image features. *J Inf Process Syst* 13(3):590–598
19. Zhang J, Jiang T, Zheng Y, Wang J, Xie J (2018) A New operator extracting image patch based on EPLL. *J Inf Process Syst* 14(3):590–599
20. Triggs B, McLauchlan P, Hartley R, Fitzgibbon A (2000) Bundle adjustment—a modern synthesis. *Lecture Notes Comput Sci Vision Algorithms Theory Pract* 1883:298–372
21. Furukawa Y, Ponce J (2009) Accurate camera calibration from multi-view stereo and bundle adjustment. *Int J Comput Vision* 84(3):257–268
22. Minor LG, Sklansky J (1981) The detection and segmentation of blobs in infrared images. *IEEE Trans Syst Man Cybernetics* 11(3):194–201
23. Hough PVC (1959) Machine analysis of bubble chamber pictures. In: 2nd international conference on high-energy accelerators (HEACC 59), December 16, pp 1184–1191
24. Marquardt DW (1963) An algorithm for least squares estimation of nonlinear parameters. *J Soc Ind Appl Math* 11(2):431–441

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
