International Journal of Advanced Science and Technology
Vol.114 (2018), pp.1-10
http://dx.doi.org/10.14257/ijast.2018.114.01

Squash Game Based on Augmented Reality

Seok-Han Lee

Department of Information and Communication Eng., Jeonju Univ.,
303 Chunjam-ro, Wansan-gu, Jeollabuk-do, Korea
seokhan@jj.ac.kr

Abstract

In this paper, we describe an augmented reality (AR) squash game, which is a squash-style game based on the AR technique. Our system operates based on the 2D–3D projective geometry. In the proposed system, a visual camera takes real-time images of a room that consists of several numbers of walls, and the geometric information of the room image is computed using planar markers, which are attached to the walls. Using the projective geometry and motion data of a ball created artificially, a realistic scene is generated in which a user can interact with a ball that makes realistic motions. A user watches the computer screen that displays the mixed image, and hits the virtual ball using a racket connected to a motion tracker. The ball motion is created based on the parabolic motion equations. Based on these motion data, the ball is created using the OpenGL library, and is overlaid on the real image received from the CCD camera. The position and the motion data of the racket are obtained from a 3D motion tracker. To compute the geometric information of the planes, which act as walls in the images, we compute the projective mapping and estimate the position and orientation of the planes using planar calibration objects.

Keywords: *Human–Computer Interaction, Augmented Reality, Hand Tracking, Computer Graphics, Projective Geometry*

1. Introduction

In this paper, we describe an augmented reality (AR) squash game, which is a squash-style game based on the AR technique. AR is a technology that involves the overlay of virtual objects or annotations on a real scene and can be used for a wide variety of applications. We made this game using the AR technology to estimate the geometric information of images taken from a visual camera. Figure 1 shows a brief diagram of the system. A CCD camera takes the real-time images of a room formed by some number of walls, and the geometric information of the image is estimated using markers attached to the walls. By the geometric information and motion data of a ball created internally, a realistic scene is generated in which the ball moves like a real one. Further, a user watches the computer screen that displays the mixed image, and tosses the ball using a racket connected to a motion tracker. The ball motion is created based on parabolic motion equations. With these motion data, the ball is created using the OpenGL graphic library, and is mixed onto the real image from the CCD camera. The position and the motion data of the racket are received from a 3D motion tracker. Figure 2 shows the block diagram of our system.

2. Geometric Information Estimation

To estimate the geometric information of the planes that act as walls in the images, we use the AR algorithm to estimate the position and orientation of the plane using a calibration

Received (October 21, 2017), Review Result (January 25, 2018), Accepted (January 31, 2018)

ISSN: 2005-4238 IJAST
Copyright ⓒ 2018 SERSC Australia

object, known as the "marker." For this algorithm, we use the AR toolkit library. The AR toolkit library is a software library for developing AR applications. Using the library, we can estimate the geometric information of the markers attached to the walls in the image easily, and therefore, those of the walls. This geometric information is used to estimate the relationships of the coordinates of the three-dimensional space projected into the camera [1, 2].

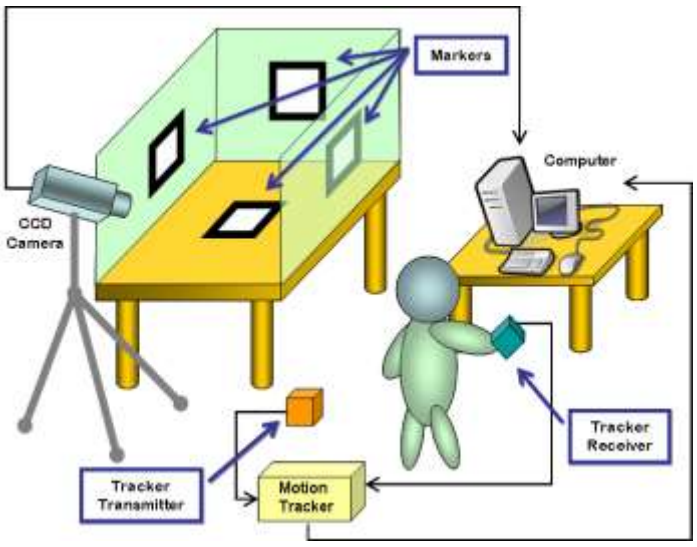


Figure 1. Conceptual Diagram of the System

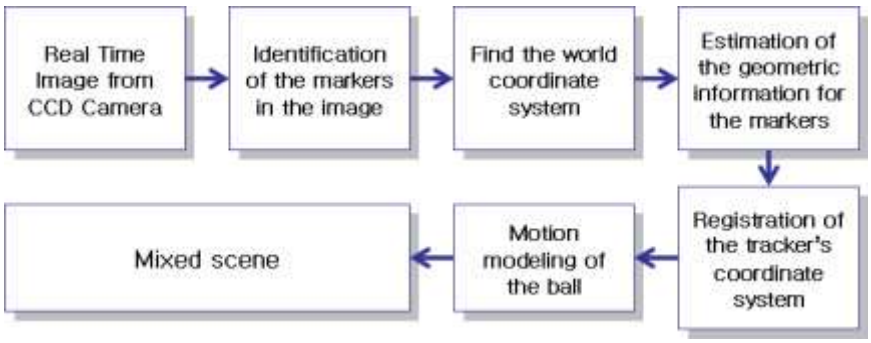


Figure 2. Proposed System's Workflow

2.1. Coordinate transformation

The position and orientation of the planes in the image are estimated by the markers. To realize the ball motion and the mixed scenes, we consider four coordinate systems:

1) World coordinate system

2) Object coordinate systems for each wall

3) Motion tracker's coordinate system

4) Camera coordinate system

The relationships among the coordinate systems are described in Figure 3, and from Eq. (1) to Eq. (4). In Figure 3, H1, H2, and H3 represent the transformations between the i-th object coordinate systems for each of the planes and the camera coordinate system. We consider the coordinate system of the floor plane as the world coordinate system. The transformation relationships among the coordinate systems can be described as follows.

 Copyright ⓒ 2018 SERSC Australia

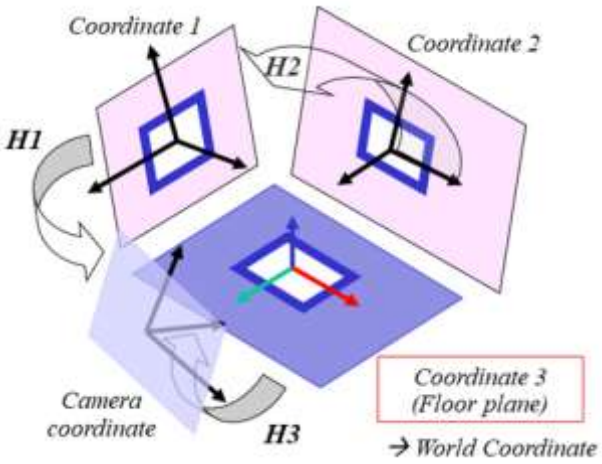


Figure 3. Coordinate Mappings for the Mixed Scene

$$\mathbf{P}_{ci} = \mathbf{H}_{wi}\mathbf{P}_{Mi}, = 1, 2, 3, \dots, N, \quad (1)$$

$$\mathbf{P}_{CB} = \mathbf{H}_{B}\mathbf{P}_{MB}, \quad (2)$$

$$\mathbf{P}_{MB} = \mathbf{H}_{B}^{-1}\mathbf{H}_{wi}\mathbf{P}_{Mi}, = 1, 2, 3, \dots, N, \quad (3)$$

$$\mathbf{H}_{T} = \mathbf{H}_{B}^{-1}\mathbf{H}_{wi}, = 1, 2, 3, \dots, N. \quad (4)$$

The position and orientation of the planes in the image are estimated by the markers. To produce the ball motion and the mixed scenes, we consider four coordinate systems. In the equations above, N denotes the total number of markers present in the image. $\mathbf{H}_{wi}$ represents the transformation matrix between the camera coordinate and i-th object coordinate system, $\mathbf{H}_{B}$ is the transformation matrix between the camera coordinate system and the floor plane coordinate system, and $\mathbf{P}_{Mi}$ represents the homogeneous coordinate of a point located in the i-th object coordinate system. $\mathbf{P}_{MB}$ denotes the homogeneous coordinate of the point located in the floor coordinate system. The transformation between the world coordinate system (*i.e.*, the floor coordinate system) and the i-th object coordinate system is estimated by $\mathbf{H}_{T}$ in Eq. (4). Therefore, a point on the i-th object coordinate system can be transformed into a point on the world coordinate system using $\mathbf{H}_{T}$. In the system, the coordinate systems of planes (*i.e.*, walls) are transformed into the camera coordinate system, and then transformed into the world coordinate system by $\mathbf{H}_{T}$.

2.2. Coordinate Alignment of the Motion Tracking Device

We use a 3D motion tracker to track the motion of the user's moving hand. The tracking device has one receiver that is a racket, which a user holds in his/her hand and moves it, and one transmitter to detect the position and orientation of the receiver. The coordinate system of the transmitter is the reference coordinate system of the tracker. To transform the coordinate of the tracker (the position of the receiver) into the world coordinate, the reference coordinate system of the tracker should be aligned to the world coordinate system. This alignment is shown in Figure 4, and Eq. (5). To register the coordinate system of the tracker, the position, orientation, and scales along each axis should be aligned according to those of the world coordinate system. We perform this alignment using Eq. (5).

$$\mathbf{P}_{TW} = \mathbf{H}_{T}\mathbf{H}_{TW}\mathbf{P}_{PT} - \mathbf{T}_{TW}. \quad (5)$$

Copyright ⓒ 2018 SERSC Australia

In Eq. (5), $\mathbf{P}_{TW}$ is the tracker coordinates transformed into the world coordinate system, and $\mathbf{T}_{TW}$ represents the translation from the tracker coordinate system to the world coordinate system. $\mathbf{T}_{TW}$ is defined as Eq. (6). $\mathbf{H}_T$ is the transformation matrix of Eq. (4). $\mathbf{H}_{TW}$ denotes the compensation matrix for the scale difference between the world coordinate system and the tracker coordinate system. We define $\mathbf{H}_{TW}$ as Eq. (7). In Eq. (7), S_x, S_y, and S_z denote the scale factors for x, y, and z axes, respectively, in the three-dimensional space.

$$\mathbf{T}_{TW} = [\, t_x \quad t_y \quad t_z \quad 1]^{\mathrm{T}}. \tag{6}$$

$$\mathbf{H}_{TW} = \begin{bmatrix} S_x & & & \\ & S_y & & \\ & & S_z & \\ & & & 1 \end{bmatrix}. \tag{7}$$

2.3. Problem caused by Uncalibrated Camera

The AR toolkit has its own camera parameters (*i.e.*, camera projection matrix). However, these camera parameters are not accurate and some problems may arise when using the given parameters. Figure 5 shows the problem, which can be caused by inaccurate camera parameters (*i.e.*, uncalibrated camera). Owing to the uncalibrated camera, the estimation for the geometric information of the planes may cause severe error, and the geometric information can be inaccurate. To avoid this problem, we perform camera calibration to obtain the correct camera projection matrix using Zhang's camera calibration algorithm [3].

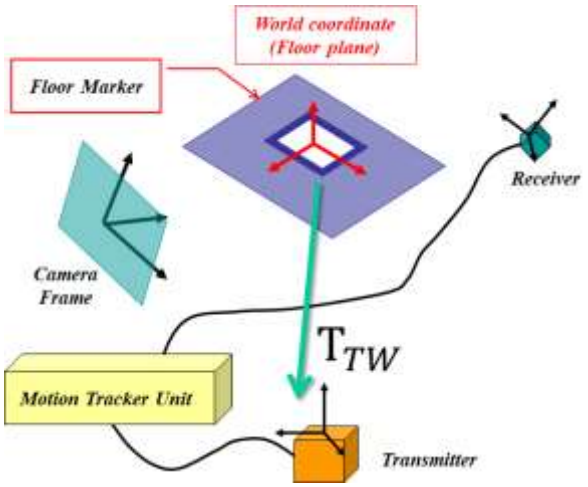


Figure 4. Coordinate Alignment of the Motion Tracker

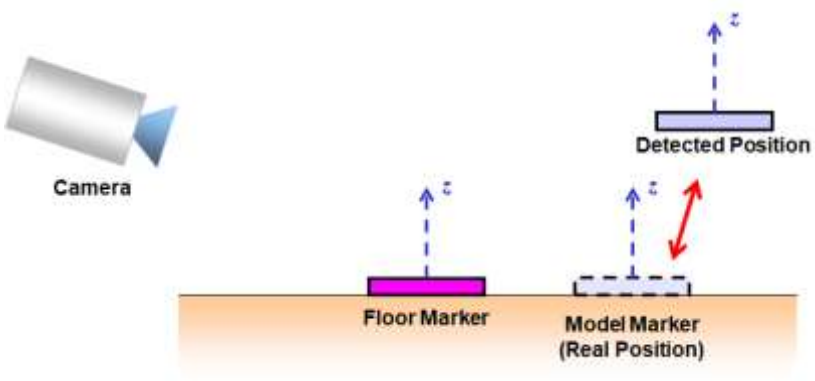


Figure 5. Problem Due to Uncalibrated Camera

Copyright ⓒ 2018 SERSC Australia

3. Motion Modeling and Collision Detection

3.1. Motion Modeling of the Moving Object

The ball motion is created based on the parabolic motion equations. The gravity direction is supposed to be in the opposite direction of the *z*-axis (*i.e.*, *-z* direction) of Figure 3, and we ignore the effect of air resistance. The virtual ball is created using the OpenGL graphic library, and the coordinate of the ball in the camera coordinate system is transformed into that of the world coordinate system using Eq. (4)[5].

3.2. Collision Detection of the Moving Object

To detect the collision of the moving object with the planes, we examine the distance between the center of the object and each plane. As we consider the object as a ball, the collision detection problem will be that for a spherical object. If the distance is smaller than the radius of the object, we consider that the collision had occurred. This is shown in Figure 6. Using Eq. (1)–Eq. (4), we estimate the normal vectors of every plane in the three-dimensional space as follows.

$$\overrightarrow{n_i} = \frac{p_1 \times p_2}{|\overrightarrow{p_1} - \overrightarrow{p_2}|} = a\overrightarrow{u_x} + b\overrightarrow{u_y} + c\overrightarrow{u_z},\ i = 1, 2, 3, \ldots, N. \tag{8}$$

In Eq. (8), N is the total number of planes in the image; p_1 and p_2 are two arbitrary vectors of each plane; u_x, u_y, and u_z denote the unit vectors of each axis of the plane coordinate system. Using Eq. (8), we can obtain the equation of every plane.

$$ax + by + cz + d = 0. \tag{9}$$

For the plane of which its equation is Eq. (9), the distance between the plane and the ball is obtained as Eq. (10).

$$D_t = \frac{ax_c + ay_c + az_c + d}{\pm\sqrt{a^2 + b^2 + c^2}}. \tag{10}$$

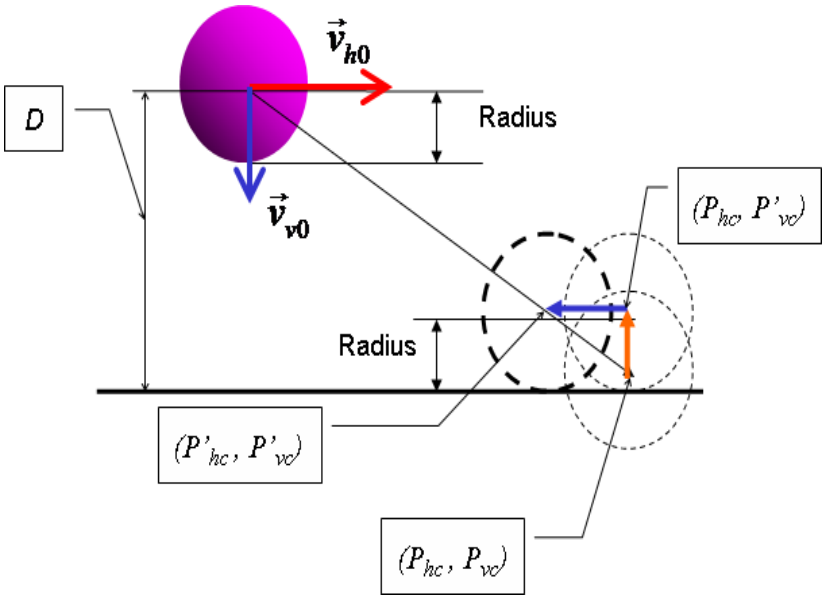


Figure 6. The First Collision Case

Copyright ⓒ 2018 SERSC Australia

In Eq. (10), D_t is the distance between the center of the object and the plane. x_c, y_c, and z_c denote the coordinates of the center of the object along each axis. The negative value of Eq. (10) means the opposite direction of the collision, a value of which we ignore. For every plane, we specify two cases in which a collision occurs.

Case 1: Figure 6 shows the first collision case. The moving object is not in contact with any wall, or the racket before the collision occurs. In this case, the corrected position of the object when the collision occurs is estimated as Eq. (11), and Eq. (12)

$$P'_{vc} = P_{vc} - R_a, \tag{11}$$

$$P'_{hc} = P_{hc} + |v_{h0}|t_c. \tag{12}$$

In Eq. (11) and Eq. (12), P'_{vc} and P'_{hc} denote the corrected center of the object after the collision, whereas P_{vc} and P_{hc} mean the center of the object when the collision occurs. R_a is the radius of the object, and v_{h0} means the horizontal velocity vector of the object. t_c is the time taken to move from the previous position to the position where the collision occurs and can be estimated as Eq. (13).

$$t_c = \frac{-v_{v0} \pm \sqrt{|\vec{v}_{v0}|^2 - 2|\vec{g}|D}}{|\vec{g}|}. \tag{13}$$

In Eq. (10), D is the distance between the center of the object and the plane. $\vec{g}$ denotes the gravity acceleration vector of the object, and a positive value of $\sqrt{|\vec{v}_{v0}|^2 - 2|\vec{g}|D}$ is due to the opposite direction of the collision, a case of which we ignore. The corrected velocity vectors of the object motion can be obtained as Eq. (14) and Eq. (15).

$$\overrightarrow{v'_h} = \overrightarrow{v_{h0}}t_c, \tag{14}$$

$$\overrightarrow{v'_v} = \overrightarrow{v_{v0}} - \vec{g}t_c. \tag{15}$$

In Eq. (14) and Eq. (15), $\overrightarrow{v'_h}$ and $\overrightarrow{v'_v}$ denote the horizontal and vertical motion vectors of the previous position, respectively.

Case 2: The second collision case is shown in Figure 7. This is for the case where the object is in contact with the walls or thes racket before the collision. In this case, we first estimate a line l_0 of Figure 7. The intersection of l_0 and the plane is computed as follows.

$$\alpha x + \beta y + \gamma z + d = 0, \tag{16}$$

$$x = m_1 + \alpha t, \;\; y = m_2 + \beta t, \;\; z = m_3 + \gamma t. \tag{17}$$

Eq. (16) is the equation of the plane, and Eq. (17) is the parametric equation of l_0. From the two equations, we obtain the intersection point (x_0, y_0, z_0). Further, from the vector $\overrightarrow{P_0}(x_0, y_0, z_0)$, the vector $\vec{n} = (m_1, m_2, m_3)$ that represents the direction of the correct object center can be estimated.

 Copyright ⓒ 2018 SERSC Australia

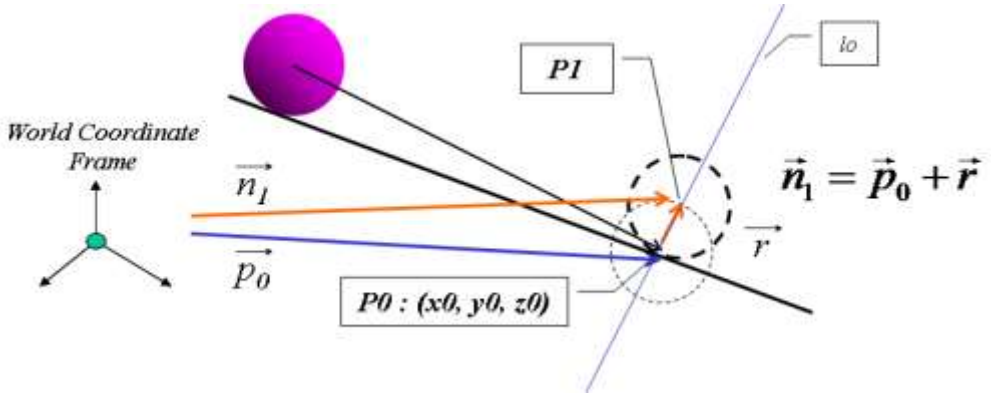
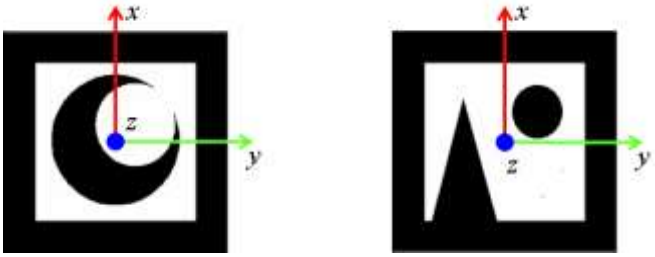


Figure 7. The Second Collision Case



(a) (b)

Figure 8. Square Markers for the Identification of Planes in the Scene: (a) the Marker for the World Coordinate Systems, and (b) for other Coordinate Frames

$$\vec{n} = \overrightarrow{p_0} + \vec{r}. \tag{18}$$

Here, $\vec{r}$ is the vector from the intersection to the correct coordinate of the object center, which is defined as Eq. (19).

$$\vec{r} = R_a\left(\overrightarrow{u_x} + \overrightarrow{u_y} + \overrightarrow{u_z}\right). \tag{19}$$

From Eq. (18), $\vec{n}$ points the correct position of the object center; therefore, we can estimate the correct center (m_1, m_2, m_3).

4. Results

To estimate the geometric information of the wall, we use the AR toolkit library and several square markers [4]. Figure 8(a) and Figure 8(b) show the square markers used in our system. The marker in Figure 8(a) is used for the world coordinate system, and this marker is located on the floor plane. The marker in Figure 8(b) is for the identification of each plane. Figure 9 represents the motion tracker used for the system. The receiver module is attached to the back side of a ping pong racket such that a user feels that he/she is actually playing a ping pong game. The primary system consists of a computer with an Intel CPU running Windows 7, a frame grabber that collects the real-time image stream, and a progressive scan CCD camera with a resolution of 640 × 480. The system shows an average frame rate of 24–30 fps. Before the system is switched on, we first calibrate the CCD camera. The calibration is performed using Zhang's algorithm as shown in Figure 10[3]. Zhang's algorithm requires at least three images of the calibration object to calibrate the camera. Therefore, we use nine images for the camera calibration. The camera calibration should be performed when the camera is connected to the system for the first time, and

Copyright ⓒ 2018 SERSC Australia

subsequent calibrations are not necessary every time the system starts to run. Figure 11 shows the demonstration of the system. In the figure, the screen was projected onto a wall using a projector to provide a larger view to the user.

5. Conclusion

In this paper, we presented a squash-style game based on the AR technique. Some demonstrations were illustrated, and the system showed satisfactory performance as a prototype. However, some problems were also detected. First, jitters on the planes (walls) in the virtual (camera) coordinate may occur when the large-sized square markers are not used, or the marker image is severely blurred. These jitters may cause serious problems to the output image. To minimize this problem, a refinement algorithm may be used, which causes some trade-offs in the real-time process. Further, more entertaining components of a game should be enhanced. We use a beam projector to display the augmented scene, and also plan to use a head-mounted device (HMD) for the system. However, the overall weight of the camera-mounted HMD may cause several problems. This problem can be avoided if we use a mobile device such as a tablet computer, smart phone, or lightweight HMD.

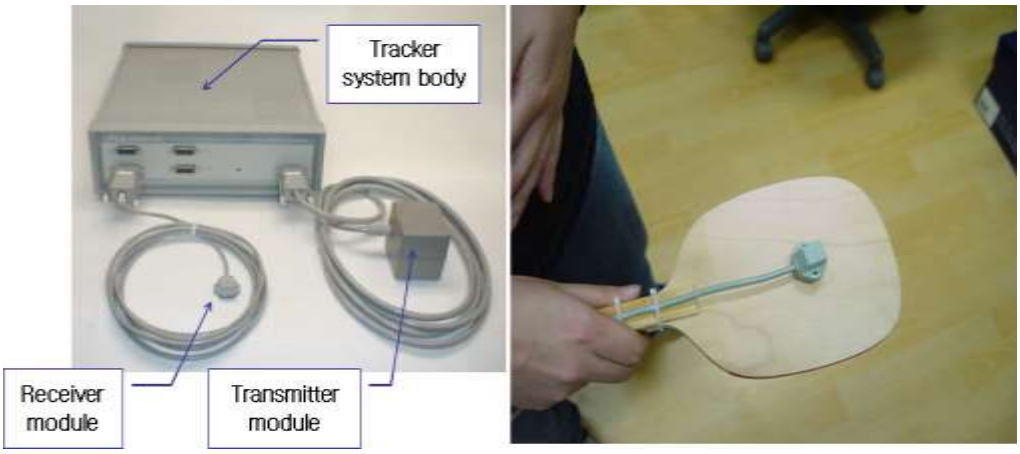


Figure 9. Motion Tracker used for the System

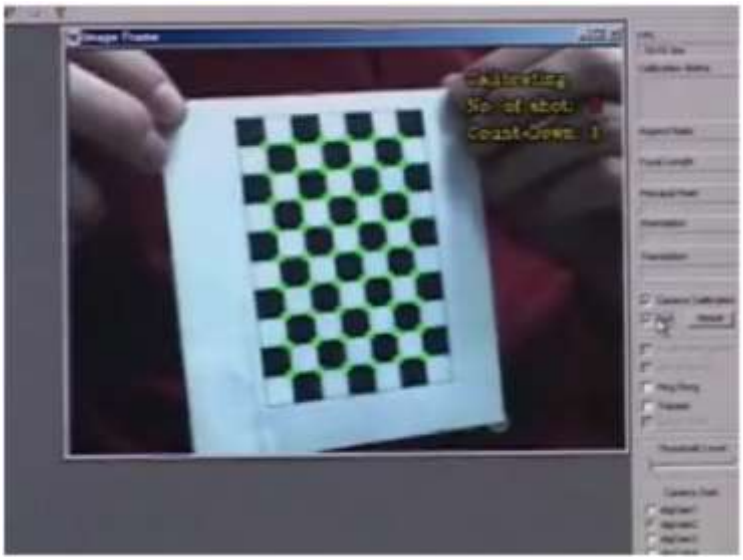


Figure 10. Camera Calibration

 Copyright ⓒ 2018 SERSC Australia

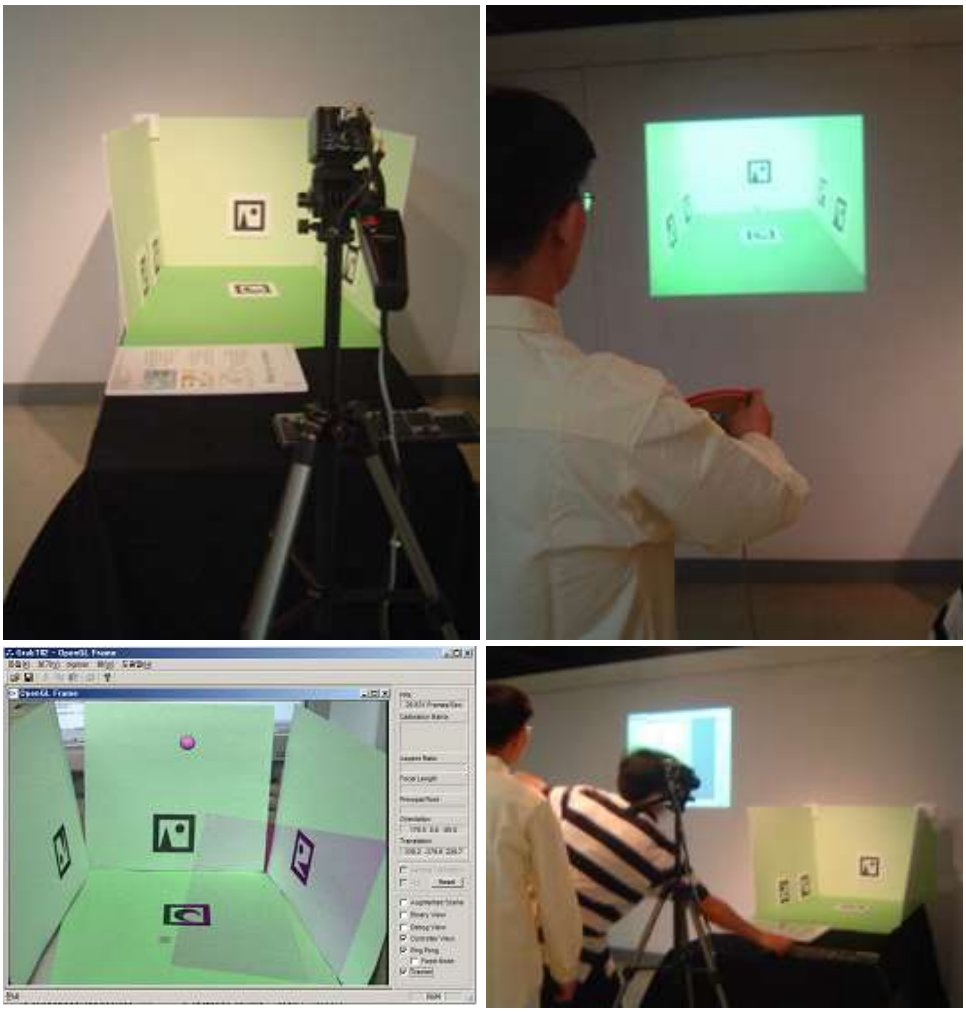


Figure 11. Demonstration of the Proposed System

Acknowledgment

This paper is a revised and expanded version of a paper entitled, "Interactive Game using the Augmented Reality Technique," presented at the "5th International Conference on Interdisciplinary Research Theory and Technology," Daejeon University, Korea, December 21–23, 2017.

References

[1] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," 2nd edition, Cambridge Press, New York, **(2011)**.
[2] O. Faugeras, "Three-Dimensional Computer Vision", The MIT Press, Massachusetts, **(1993)**.
[3] Z. Zhang, "A Flexible New Technique for Camera Calibration and the Application," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, **(2000)**, pp. 1330-1334.
[4] http://www.hitl.washington.edu/people/poup/research/ar.htm.
[5] T. Akenine-Moller, E. Haines and N. Hoffman, "Real-Time Rendering", 3rd edition, AK Peters, Massachusetts, **(2008)**.
[6] E. Trucco and A. Verri, "Introductory Techniques for 3-D Computer Vision", Prentice Hall, New Jersey **(1998)**, pp. 146-148.
[7] L. Vacchetti, V. Lepetit and F. Fua, "Stable Real-Time 3D Tracking Using Online and Offline Information," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 10, **(2004)**, pp. 1385-1391.
[8] O. Faugeras, "The Geometry of Multiple Images", The MIT Press, Massachusetts, **(2001)**.

Copyright ⓒ 2018 SERSC Australia

[9] M. Mekhtiche, M. Bencherif, M. Algabri, M. Alsulaiman, M. Hedjar, M. Faisal and K. AlMutib, "Real Time Object Detection & Tracking over a Mobile Platform", Indian Journal of Science and Technology, vol. 8, no. 22, **(2015)**, pp. 1-7.
[10] D. Lee and Y. Park, "Vision-Based Remote Control System by Motion Detection and Open Finger Counting", IEEE Transactions on Consumer Electronics, vol. 55, no. 4, **(2009)**, pp. 2308-2313.
[11] M. Chandrajit, R. Girisha, T. Vasudev and M. Hemesh, "Data Association and Prediction for Tracking Multiple Objects", Indian Journal of Science and Technology, vol. 9, no. 23, **(2016)**, pp. 1-13.
[12] K. Cho, H. Kim and Y. Lee, "Augmented Reality Coloring Book with Transitional User Interface", Indian Journal of Science and Technology, vol. 9, no. 20, **(2016)**, pp. 1-5.
[13] Z. Ren, J. Yuan and Z. Zhang, "Robust hand gesture recognition based on finger-earth movers distance with a commodity depth camera", Proceedings of ACM Multimedia, Scottsdale, AZ, **(2011)** November 28-December 1.
[14] A. Davison, I. Reid, N. Morton and O. Stasse, "MonoSLAM: Real-time single camera SLAM", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, **(2007)**, pp. 1052-1067.
[15] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces", Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara Japan, **(2007)** November 13-16.
[16] M. Pupilli and A. Calway, "Real-time camera tracking using a particle filter", Proceedings of British Machine Vision Conference, Oxford, UK, **(2005)** September 5-8.
[17] K. Xu, K. W. Chia and A. D. Cheok, "Real-time camera tracking for marker-less and unprepared augmented reality environments", Image and Vision Computing, vol. 26, no. 5, **(2008)**, pp. 673-689.
[18] E. Eade and T. Drummond, "Scalable monocular SLAM", Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, New York, NY, **(2006)** June 17-23.
[19] E. Eade and T. Drummond, "Monocular SLAM as a graph of coalesced observations", Proceedings of IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil **(2007)** October 14-20.
[20] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas and A. Calway, "Robust real-time visual SLAM using scale prediction and exemplar based feature description", Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, Anchorage, AK, **(2007)** June 17-22.
[21] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas and A. Calway, "Real-time and robust monocular SLAM using predictive multi-resolution descriptors", Proceedings of International Symposium on Visual Computing, Lecture Notes in Computer Science, vol. 4292, **(2006)**, pp. 276-285.
[22] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar and N. Navab, "Marker-less tracking for AR: A learning-based approach", Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, **(2002)** September 30-October 1.
[23] R. Subbarao, P. Meer, and Y. Genc, "A balanced approach to 3D tracking from image streams", Proceedings of the 4th IEEE and ACM International Symposium on Mixed and Augmented Reality, Vienna, Austria **(2005)** October 5-8.

Authors

Seok-Han Lee, he received his B.S. degree in Electronic Engineering in 1999, and M.S., and Ph.D. degree in Image Engineering, in 2001, and 2009, respectively from Chung-Ang University, Seoul, Korea. From 2001 to 2004, he worked as an engineer in LG Electronics Inc, and from 2010 to 2013, a research professor of Chung-Ang University. He has been working as a professor of Jeonju University since 2013. His research interests include real-time camera tracking, augmented reality, and 3D computer vision.

 Copyright ⓒ 2018 SERSC Australia