A Gesture Based Camera Controlling Method in the 3D Virtual Space

Jong-Oh Kim[1], Chan Park[1], Ji-Seong Jeong[1], Nakhoon Baek[2] and Kwan-Hee Yoo[1]*

[1] Department of Digital Informatics and Convergence, Chungbuk National University, 410 Seongbongro Heungdukgu Cheongju Chungbuk, South Korea

[2] Department of Digital Informatics and School of Computer Science and Engineering, Kyungpook National University,Daegu 702-701, South Korea

*{dmkdmk01,farland83, szell, *khyoo}@chungbuk.ac.kr, oceancru@gmail.com*

Abstract

In this paper, we propose a gesture user interface for controlling a virtual camera in the 3D virtual space, which can be used to navigate the 3D virtual space by users. In the proposed mechanism, Kinect [1] is used as an input device for detecting positions of user gestures. After obtaining the positions information from Kinect, the sequences of positions are filtered to be recognized as any kind of gestures for controlling the virtual camera. The recognized gestures will push the corresponding operation into the camera in the 3D virtual space so that the 3D space can be navigated according to the user-steered control. Experimental results show that the proposed method achieved the gesture recognition ratio of more than 75%.

Keywords: *hand gesture, user interface, virtual camera navigation, 3D virtual space*

1. Introduction

Traditional user interfaces for controlling a virtual camera in the 3D virtual space are achieved by using devices such as mice and keyboards. Since the equipments are not familiar with usage of the user interface in 3D space, gesture based user interfaces have been used to give better immersion to the users by allowing them to more naturally and friendly control the virtual camera and/or objects in 3D space. There are many attempts to support gesture user interfaces by using image based approaches [2-11]. Most systems capture the sequence of images from Webcam-like devices, process those images to enhance their qualities, and then recognize postures and dynamic gestures by comparing the images into predefined gesture images stored in the DB(data base). However, there are some problems with respect to the exactness and the performance since image recognition techniques cannot completely solve the problem in real time.

Recently, Microsoft Kinect-like devices, called motion sensing input devices, have been developed, and they enabled the users to control and interact with 3D virtual space without any need for touch screens, mice and/or keyboards, through the more natural user interface of using gestures and spoken commands. Even if Microsoft Kinect has been intended as an interface for the Xbox gaming console alone, many applications have been developed by using the device in various services. One of the recent research results is DepthJS [12] developed by MIT Media Laboratory. DepthJS allows users to visualize any web page through gesture based interaction between the web browser and Kinect-users. However,

* A corresponding author

DepthJS still does not completely support the camera navigation in 3D virtual space. In this paper, we propose a system for naturally controlling a virtual camera in 3D virtual space. To do so, our proposed system filters the sequences of positions captured from Kinect devices, and then recognizes the captured information as gestures for controlling a camera. The recognized gesture will push the corresponding operation into the camera in 3D virtual space so that the 3D space can be navigated according to user-steered control. Experimental results show that the proposed method can get the gesture recognition ratio of more than 75%.

This paper is organized as follows: Section 2 illustrates the proposed system for controlling a virtual camera based on the gestures. Section 3 describes its implementation results, and Section 4 points out further research directions

2. Gesture Based Camera Control System

In this paper, we propose the system to control a virtual camera in 3D virtual space based on hand gestures as shown in Figure 1. Kinect[1], which is a motion sensing input device made by Microsoft for the Xbox 360 video game console and Windows PCs, has been used to capture position information for tracking the user's skeleton images. The positions are presented as the location (x, y, depth), at each joint point of the skeleton from the Kinect. As shown in Figure 1, in the proposed system, hand moving positions will be used to control a virtual camera in 3D space.

Even if Kinect input devices are processing any filtering for the captured positions, the flicker on the sequence of positions can be still occurred. In this paper, two operations, depth calibration and depth cueing, for the sequence of positions are applied to reduce the flicker. Generally, users using the proposed system have difference lengths of their arms. In order to more accurately recognize their gestures, our system essentially records those lengths of their arms into the internal data structures. The information will be used to decide the region of gesture recognition. The length, A_l, of users' arms can be calculated as follows:

$$A_l = |(A_{lmax} - A_{lmin})| \quad (1)$$

Where A_{lmax} and A_{lmin} are depths of the arms captured from Kinect device during five seconds when users are maximally stretching their arms up, and when users are maximally busting their arms to their body, respectively.

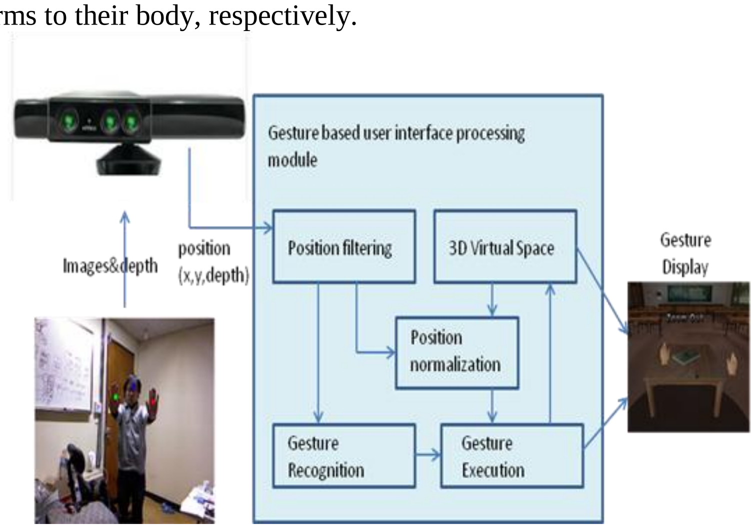


Figure 1. Actions of Gestures for Controlling a Virtual Camera

Therefore, the length A_l designated as the arm length of users is defined as absolute value of the difference of two values, A_{lmax} and A_{lmin}. The registered length will be applied to decide activation and non activation of recognition of users' gestures. Next, depth cueing is processed for smooth moving of hand positions by applying Kalman filter [13]. After initializing our gesture recognition system, users through performing their gestures will control a virtual camera in 3D virtual space. Activation and inactivation of gestures recognition are automatically processed as shown Figure 2. The space is divided into two subspaces according to the length of users' arms, which is obtained during that users perform their gestures. If the length is less than $0.2 * A_l$, the gestures are ignored, and if the length is greater than $0.2 * A_l$ and is less than $0.8 * A_l$, the gestures will be recognized as operations of camera controlling gestures.

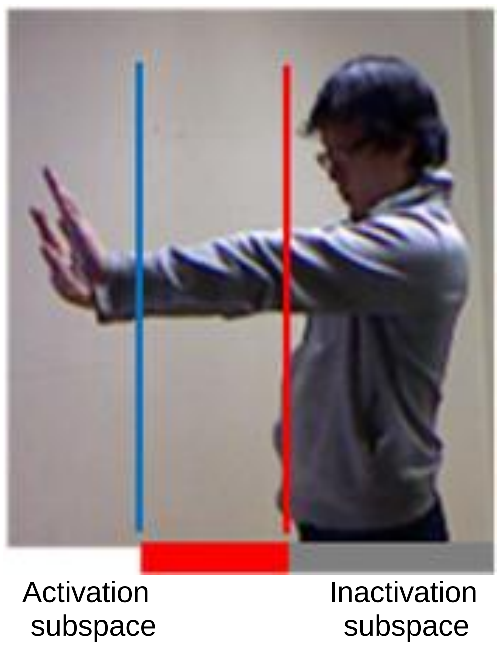


Figure 2. Dividing Spaces into Two Subspaces for Activation and Inactivation of Gestures

As shown in Figure 2, gestures made in the inactivation subspace are ignored, and only gestures done in the activation subspace will be used to control a virtual camera in 3D space. In order to more accurately recognize gestures, we recheck out the subspace in which the gestures are performed, and then the standby time during three seconds is given. A virtual camera will be moved according to the recognition results of continuous gestures in the valid subspace. The procedure is well illustrated in Figure 3.

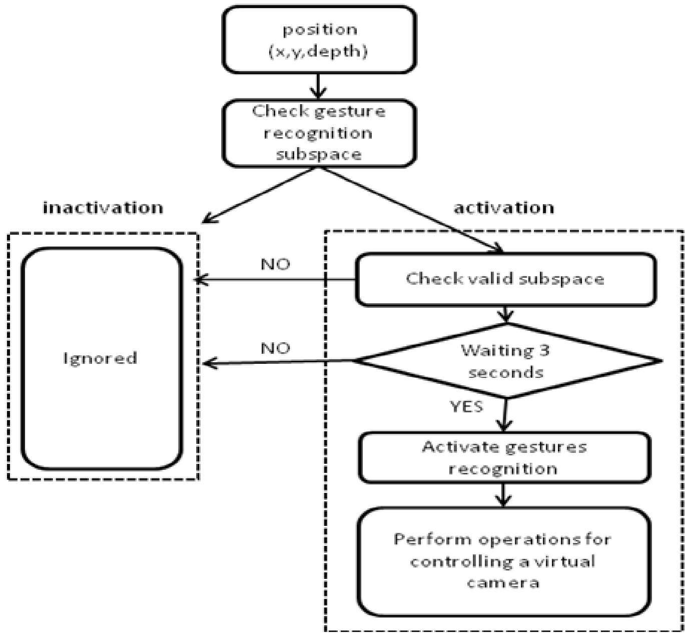


Figure 3. The Procedure to Recognize Users' Gestures

Two, left and right, hands will be used to control a virtual camera in the proposed system. Twenty four kinds of gestures are created if four movement vectors, those are, upper, lower, left and right vectors, of two hands are combined. When two hands are independently moved into any place each other, 16 (=4x4) gestures can be created since the movement vectors are simultaneously achieved. On the other hand, when one hand is fixed and another is moved, 8(=2x4) gestures are created. In this system, only six gestures of twenty four will be defined in Table 1.

Table 1. Gesture Types and their Explanation

Gesture Types	Explanation
G1(left rotation of a camera)	Actions of moving the right arm to the right direction, while holding the left arm.
G2(right rotation of a camera)	Actions of moving the left arm to the left direction, while holding the right arm.
G3(up rotation of a camera	Actions of moving both the left and right arms to the up direction.
G4(down rotation of a camera)	Actions of moving both the left and right arms to the down direction.
G5(zoom-in of a camera)	Actions of outstretching both left and right arms.
G6(zoom out of a camera)	Actions of pursing both left and right arms

Initial state	gesture operations	Left hand	Right hand	Functions
		←	-	**Left (Rotation)** Move camera in positive direction of y- axis based on 3D model.
		-	→	**Right (Rotation)** Move camera in negative direction of y- axis based on 3D model.
		↑ -	- ↑	**Up (Rotation)** Move camera in positive direction of x- axis based on 3D model.
		↓ -	- ↓	**Down (Rotation)** Move camera in positive direction of x- axis based on 3D model.
		←	→	**Zoom In(Scaling)** Decrease distance between a camera Position and a center of 3D model
		→	←	**Zoom Out (Scaling)** Increase distance between a camera Position and a center of 3D model

Figure 4. Gestures Actions for Controlling a Virtual Camera

To simulate six operations for controlling a virtual camera using gestures, we assume that the center position P=(Pcx, Pcy, Pcz) of the camera, which is always looking to the origin O=(Ox,Oy,Oz) of the 3D virtual space with fixed up vector, can be placed at any location of the 3D virtual space. In this situation, P will be transformed according to actions of the suggested gestures. To simulate naturally the movement of the camera, first of all, we have to consider the movement of gestures. Assume that the current position of a hand is located at Gc as shown in Figure 5(a). When the movement position of the hand is escaped out of the designated circle with its center Gc and its radius Gr, the movement will be detected and one of four locations A, B, C, and D, which are defined according to the relative position from Gc as shown in Figure 5(a), will be returned. Six gestures can be recognized as illustrated in Table 2 according to the detected regions of moving two hands.

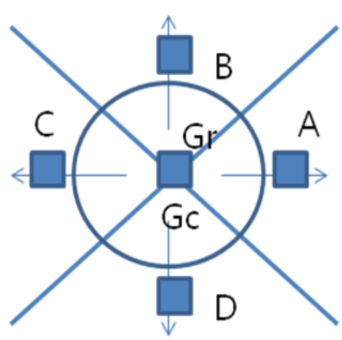


Figure 5. Relative Movement Regions of a Hand

The recognized gestures will be executed with the degree of rotation and scaling. Euclidean distance between Gc and the movement position of the gesture will be used as the degree. Three kinds of the degrees can be defined for two given points, O and P, as shown in Figure 6, where O and P are the center of the 3D virtual space and that of the camera, respectively. Two, rxv and ryv, of them are defined to be angles between O and P with respect to x-axis and y-axis, respectively. The last one, rsv, is defined to be Euclidean distance between O and P.

Table 2. Recognition of Six Gesture Types

Gesture Types	Relative moving region of left hand	Relative moving region of right hand
G1	C	-
G2	-	A
G3	B	B
G4	D	D
G5	C	A
G6	A	C

To simulate smoothly the movement of the camera with the three values rxv, ryv and rsv, the center position P=(Pcx, Pcy, Pcz) of the camera is calculated by using Equation (2) with three specific values, rx, ry and rs, each of which is either increased or decreased step by step in a specific unit from zero to rxv, ryv and rsv, respectively.

$$Pcx=rs*\sin(ry)*\sin(rx), \quad Pcy=rs*\cos(ry)*Oy, \quad Pcz=-rs*\cos(rx)*\sin(ry) \qquad (2)$$

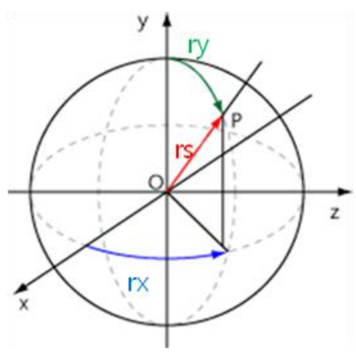


Figure 6. The Movement Amount Required for Executing Gestures

3. Experimental Results

Our experimental system was developed using Unity3D, Kinect, Microsoft C# on Windows 7 (32bit) PC. The experiments were conducted to test both the recognition and execution of the proposed six gestures. The 3D virtual space consists of a book, desks, chairs, and a blackboard in 3D virtual class room. Figure 7 shows the initial rendering result of a 3D virtual space. The initial two hands for controlling gestures are shown as 3D red objects, including markers in four directions.

Users have performed one of six kinds of gestures while seeing the rendering scenes for controlling a virtual camera in the 3D virtual space. After an initial stage, Gr has to be specified to detect the movement of hand gestures. This paper gives the value of Gr to 35 to remove shading of the camera after repeating several times. Figure 8 (a) shows the results executing gestures to give left rotations for center position of the camera. Two figures in Figure 8 (a) show that the left hand model is moved to left direction. In the similar method, Figures 8 (b), (c), (d), (e) and (f) show the rendering scenes while applying five gestures, for controlling right rotation, up rotation, down rotation, zooming in and zooming out of the camera, respectively. Each of all gestures is performed more than 500 times to test gesture recognition. Experimental results show that the proposed method can get gesture recognition ratio of more than 75%.

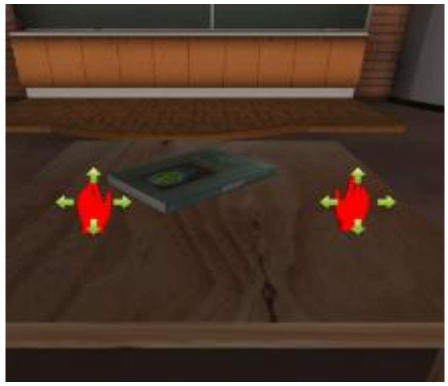
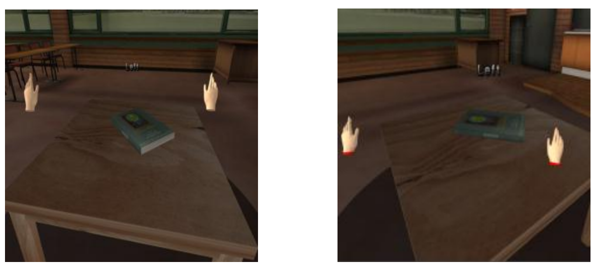
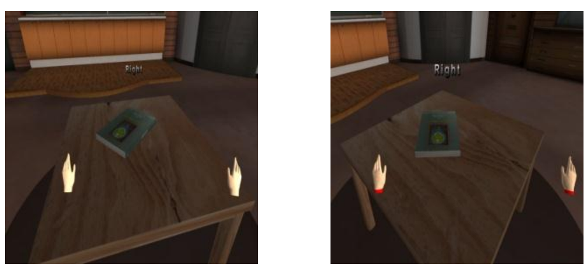


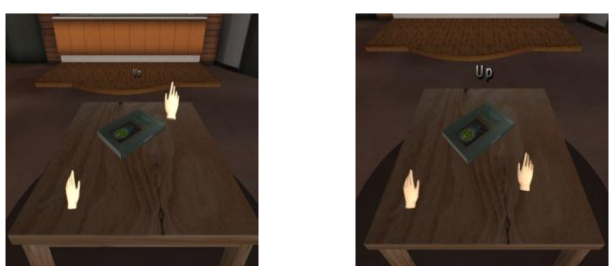
Figure 7. A Rendering Result of 3D Virtual Space Including Two Hands at Initial State for Gesture Recognition



(a) left rotation



(b) right rotation



(c) up rotation

(d) down rotation

(e) zooming in

(f) zooming out

Figure 8. Experimental Results after Applying our Proposed Gestures into a Virtual Camera

4. Conclusion

Our experimental results show that gesture user interface is very convenient to control a virtual camera in the 3D space. This study is still ongoing, and the gesture user interface researched until now will be improved in two directions. The first one is to enhance the ratio of gestures recognition, and the second one is to expand the transformation capabilities of the objects in 3D space.

Acknowledgements

This research was financially supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation, and by the ICT standardization program of MKE (The Ministry of Knowledge Economy).

References

[1] Microsoft Corp, Kinect for Windows, **(2012)**.
[2] C. Cadoz, "Gesture – music, in Trends in Gestural Control of Music", Paris, France: Ircam-Centre Pompidou, **(2000)**, pp. 71-94.
[3] M. Turk, "Gesture Recognition", Chapter 10, http://ilab.cs.ucsb.edu/projects/turk/ TurkVEChapter.pdf.
[4] J. -L. Nespoulous and A. R Lecours, "Gesture: nature and function", In The Biological Foundations of Gestures: Motor and Semiotic Aspects, Lawrence Erlbaum Assoc., **(1986)**, pp. 49-62.
[5] P. Garg, N. Aggarwal and S. Sofat, "Vision Based Hand Gesture Recognition", World Academy of Science, Engineering and Technology, vol. 49, **(2009)**.
[6] G. Berry, "Small-wall, A Multimodal Human Computer Intelligent Interaction Test Bed with Applications", Dept. of ECE, University of Illinois at Urbana-Champaign, MS thesis, **(1998)**.
[7] B. Stenger P. R. S. Mendonca and R. Cipolla, "Model-Based 3D Tracking of an Articulated Hand", In Proceedings of British Machine Vision Conference, Manchester, UK, September, vol. 1, **(2001)**, pp. 63-72.
[8] C. C. Wang and K. C. Wang, "Hand Posture recognition using Adaboost with SIFT for human robot interaction", Springer Berlin, ISSN 0170-8643, vol. 370, **(2008)**.
[9] A. L. C. Barczak and F. Dadgostar, "Real-time hand tracking using a set of co-operative classifiers based on Haar-like features", Res. Lett. Inf. Math. Sci., vol. 7, **(2005)**, pp. 29-42.
[10] C. S. Bourennane1 and L. Martin, "Comparison of Fourier descriptors and Hu moments for hand posture recognition", In Proceedings of European Signal Processing Conference (EUSIPCO), **(2007)**.
[11] J. S. Jeong, C. Park and K. H. Yoo, "Hand Gesture User Interface for Transforming Objects in 3D Virtual Space", Communication in Computer and Information Science, vol. 262, **(2011)**, pp. 172-178.
[12] MIT Media Lab., DepthJS, http://depthjs.media.mit.edu/.
[13] G. Welch and G. Bishop, "An introduction to the Kalman filter", TR95-041, Dept. of Computer Science, University of North Carolina at Chapel Hill, **(2006)**.

Authors

Jong-Oh Kim

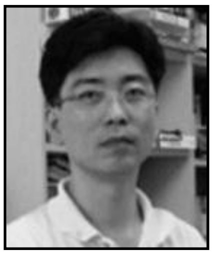
He is a MS candidate of department of digital informatics and convergence at Chungbuk National University, Korea. He received the B.S. in Computer Education from Chungbuk National University, Korea in 2012. His research interests include computer graphics, e-Learning, u-Learning, and multimedia.

Chan Park

He is a Senior Researcher in Korea Internet Software Corporation. He received the B.S., M.S. and PhD. in Computer Education and Information Industrial Engineering from Chungbuk National University, Korea in 2003, 2007 and 2012, respectively. His research interests include computer graphics, e-Learning, u-Learning, and multimedia.

Ji-Seong Jeong

He is a PhD candidate of department of IIE(information industrial engineering) at Chungbuk National University, Korea. He received the B.S. and M.S. in Computer Education and IIE from Chungbuk National University, Korea in 2008 and 2010, respectively. His research interests include computer graphics, e-Learning, u-Learning, and multimedia.



Nakhoon Baek

He is currently an associate professor in the School of Computer Science and Engineering at Kyungpook National University(KNU), Korea. He received his BA, MS, and Ph.D degree in Computer Science from the Korea Advanced Institute of Science and Technology(KAIST) in 1990, 1992, and 1997, respectively. His research interests include real-time and mobile graphics systems.

Kwan-Hee Yoo

He is a professor in the Department of Computer Education and in the Department of Information and Industrial Engineering at Chungbuk National University, Korea. He received his BA in Computer Science from Chonbuk National University in 1985, and his MS and PhD in Computer Science from KAIST (Korea Advanced Institute of Science and Technology) in 1988 and 1995, respectively. His research interests include computer graphics, 3D characters animation, and dental/medical applications.