Journal of Korea Multimedia Society Vol. 12, No. 12, DECEMBER 2009(pp. 1761-1768)

A Study on Online Real-Time Strategy Game by using Hand Tracking in Augmented Reality

Gwangha Jeon[†], Jongseok Um[††]

ABSTRACT

In this paper, we implemented online real time strategy game using hand as the mouse in augmented reality. Also, we introduced the algorithm for detecting hand direction, finding fingertip of the index finger and counting the number of fingers for interaction between users and the virtual objects. The proposed method increases the reality of the game by combining the real world and the virtual objects. Retinex algorithm is used to remove the effect of illumination change. The implementation of the virtual reality in the online environment enables to extend the applicability of the proposed method to the areas such as online education, remote medical treatment, and mobile interactive games.

Key words: Virtual reality, Retinex, hand tracking, online strategic game

1. INTRODUCTION

Augmented Reality (AR) has been received great attention since it supplies supplementary information to the real world and increases the degree of the reality. Even though equipments like HMD maximize the absorption in virtual reality, those devices cause headaches and cyber stress. AR resolves such drawbacks with keeping the same level of absorption in virtual reality.

ARToolkit has been used for ten years which compute the relative position and direction of the marker in real time compare with camera. Recently, ARToolkit plus was developed by extracting core algorithms from ARToolkit to implement AR environment to the mobile instruments [1]. Here, we use ARToolkit plus, OpenCV, and OpenGL to implement online real time strategy game which is easy to transplant to the mobile instruments.

We show the battle scene of the proposed online real time strategy game in Fig. 1. Virtual objects in AR are appeared in common to the users connected through the network. All users in the network can communicate with virtual objects. This property enables to apply the proposed algorithm to the other areas such as online education, remote medical examination and treatment and mobile interactive game. To increase the interaction between users and virtual objects, we used the hand tracking system. Typical problem occurs in hand tracking is the effect of illumination change. We used Retinex algorithm to remove illumination effects. Also detecting the distance and direction between camera and hand usually causes trouble in usual hand tracking method. We proposed the algorithm to detect the fingertip of the index finger to find the direction of the hand. To operate virtual menu by the finger movement, we developed the detection method of the finger movement.

※ Corresponding Author : Jongseok Um, Address : (136-792) 389 Samsun-dong, Sungbuk-gu, Seoul, Korea, TEL : +82-2-760-4133, FAX : +82-2-760-4488, E-mail : jsum@hansung.ac.kr

Receipt date : Nov. 30, 2009, Revision date : Dec. 14, 2009
Approval date : Dec. 21, 2009

[†] Department of Multimedia Engineering, Hansung University, Seoul, Korea
(E-mail : junkwangha@naver.com)
[††] Department of Multimedia Engineering, Hansung University, Seoul, Korea

※ This Research was financially supported by Hansung University in the year of 2008.

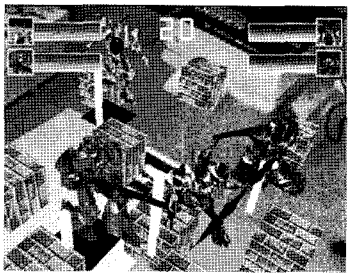


Fig. 1. Online Real Time Strategy Game Demo.

This paper composed of as follows. Section 2 introduces the SLAM and PTAM as a related and section 3 shows overall proposed system structure. Section 4 shows the implementation of the online real time strategy game and the explanation of the algorithms. Finally at section 5, conclusions and further research area are appeared.

2. RELATED STUDY

The implementation method for the AR system has two approaches: marker based approach and NFT (Natural Feature Tracking) based approach. Each approach has its own advantages and disadvantages on inserting virtual object into the scene. Marker based approach is easy to implement and superior to use in the real time process, however marker decreases the reality of the AR system. NFT uses SIFT (Scale-invariant feature transform) algorithm which extracts point features invariant to the rotation and size [2]. Such point features are used to SLAM (Simultaneous Localization and Mapping) to create a virtual workspace.

SLAM algorithm create map and trace the position of the object without prior knowledge on location. At first, SLAM has been studied in the robotic exploration. Recently, research is extended to use camera as an eye of the robot. In [3], single camera is used to construct 3D workspace by tracing depth information. However, SLAM takes a long time for finding point features, estimating camera pose and creating the map simultaneously.

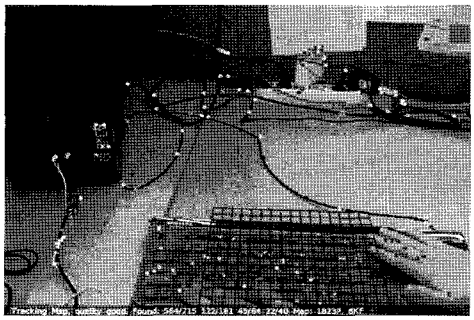


Fig. 2. PTAM map tracking operation of the system.

PTAM (Parallel Tracking and Mapping) resolves this problem by splitting tracking and mapping into two separate tasks, processed with parallel threads: one thread deals with the task of robustly tracking erratic hand-held motion, while the other produces a 3D map of point features from previously observed video frames [4]. However, the camera movement should be slow enough to update the map continuously. If camera moves fast, it causes to change the overall map. Even parallel processing improves the performance, marker based approach is superior to PTAM in its performance. In this paper, we use marker based approach. Fig. 2 shows a typical map tracking operation of the system by using PTAM.

3. SYSTEM FLOWCHART

Fig. 3 shows overall system flowchart. Camera image is used to calculate camera pose (orientation and position) and hand tracking process. ARToolkit plus library estimate camera pose base on the standard camera pose which is enrolled at the initialization step Ar_init. At AR_init, we initialize the size of the marker, camera pose and memory allocation. And then, we take binarization at the Threshold step and image segmentation at the Labeling process. At the DetectMarker step, camera pose is estimate and camera coordinates transforming matrix is calculated at Calc_trans_mat step. At Graphic System, virtual characters and virtual menu are inserted to the scene. Virtual

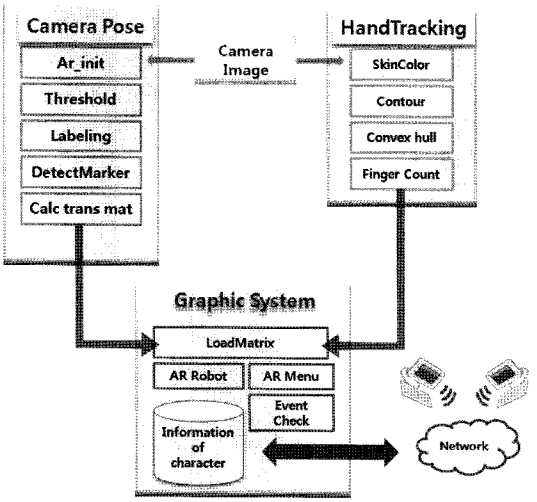


Fig. 3. System Flowchart.

menu is operated by the position and number of the fingers detected through HandTracking. As event occurs, event command and information on virtual character are transmitted to the other computers through network. We synchronize with other computer connected through network by sending various data on virtual characters when events occur.

4. IMPLEMENTATION

Fig. 4 shows the overall operating environment. A webcam is connected to each desktop computer. Each system can share the same marker at the same place or use the same type marker at different place for creating 3D virtual characters. Each system creates on its own type 3D virtual characters, which are displayed to each other's system. Characters are operated by choosing menu through hand tracking.

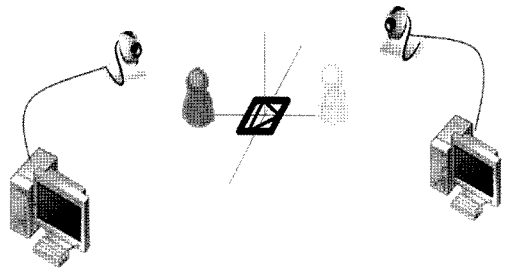


Fig. 4. AR network system.

4.1 Characters

Characters in the game are usually created by using 3D tools such as 3D Max, Maya or MilkShape. Various file formats such as 3DS, X, MD2 and ASE are used for exporting files. We use MD2 file format for 3D characters whose format is introduced by iD Software. Since MD2 format has high compression rate and easy to use, it is effective to create game character. Fig. 5 shows the virtual characters having MD2 format imported by using OpenGL.

We use two characters which are opened to the public use. Character is displayed according to the type of the marker. If we register additional markers, then we can bring new characters. If we use certain amount of markers, then we can assign specific function on each marker. This property can be applied to the development of card game [5,6]. Each character has its own features like close range attacker or long range attacker. Also, we assign features to the character such as moving distance, attack range and weapon for strategic role playing game.

Here is the process of character registration:

1. Locate marker into map display.
2. Move marker where you want, then click mouse button.
3. Character is registered, so remove marker.
4. Repeat from Step 1.

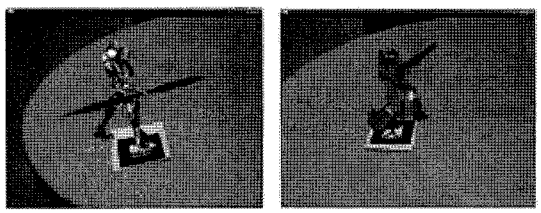


Fig. 5. 3D Virtual Characters.

4.2 Virtual Menu

Fig. 6 shows the virtual menu to operate characters. We use picking process to choose unit and menu entry with HandTracking event in the 3D space. Picking process is operated using CCW algorithm in AR environment. The CCW algorithm simply checks the location to which menu entry belongs. Virtual menu has entries such as move, attack, item, turn over and cancel. Once character is picked, then menu appears on the screen. Fig. 7 shows the result of the picking process.

4.3 Moving and Attack of Virtual Object

In the strategy game, characters move grid by grid on the grid type map. Since this map looks like 2D array, we can apply the shortest path search algorithm such as A*, breadth-first search (BFS) or depth-first search(DFS). To move character, we have to manifest the possible range of movement of the character In graph theory, breadth-first search (BFS) is a graph search algorithm that begins at the root node and explores all the neighboring nodes. BFA is an appropriate method for movement of the character in this case

Fig. 6: Virtual Menu.

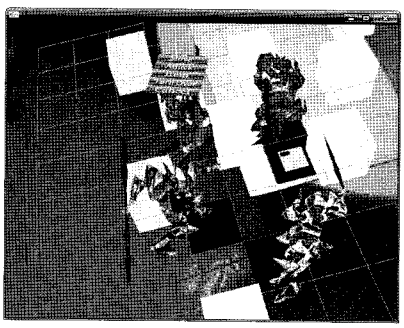


Fig. 7. 3D Virtual character and Virtual menu.

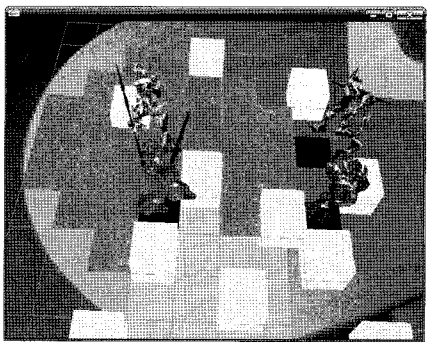


Fig. 8. Range of possible movement and path search.

which shows equal distance map centered at the reference point. In Fig. 8, blue area shows the range of the possible movement of the upper left character. Based on BFS, we can search the shortest path to the destination.

4.4 Control of Virtual Object by Hand Tracking

In this section, we show how users interact with virtual characters using hand tracking. Here, we introduce Retinex algorithm to remove the effect of illumination change. Fig. 9 shows overall processes of the event handling by using hand tracking. After getting image from webcam, we detect skin color, remove noise, perform image segmentation,

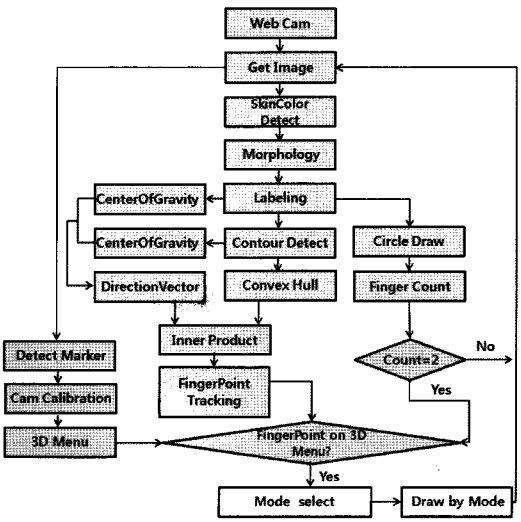


Fig. 9. Event handling flowchart with hand tracking.

detect fingertip, and find the number of fingers. If hand moves with one finger, it means mouse movement. If the number of fingers is two, it means click the entry of menu where index fingertip points.

4.4.1 Remove illumination effect with Retinex

Images obtained from webcam vary according to the illumination condition, which changes the color of the images. We use skin color to track hand so that removing the illumination effect is important to detect hand. We use Retinex algorithm which remove illumination effect by splitting color of image into two parts: color from its own reflection (Retinex output) and illumination [7].

Equation (1) shows the mathematical form of single-scale Retinex (SSR),

$$R(x,y) = \log\big(I(x,y)\big) - \log\big(F(x,y) * I(x,y)\big) \quad (1)$$

where $R(x,y)$ is the Retinex output, $I(x,y)$ is the color of the input image and "$*$" symbol represent convolution, and $F(x,y)$ is the surround filter (usually Gaussian) defined by

$$F(x,y) = K \exp(-(x^2 + y^2)/c^2).$$

Here, K is the normalizing constant and c is a scaling factor. In this paper, we use multiscale Retinex (MSR) algorithm which shows the flowchart at Fig. 10. MSR uses several scaling factors ($c1$, $c2$, $c3$) and get the MSR result by summing SSR results with weights ($w1$, $w2$, $w3$) as in Fig. 10. The result of the MSR output is in Fig. 11. It shows the restored color image compensated from

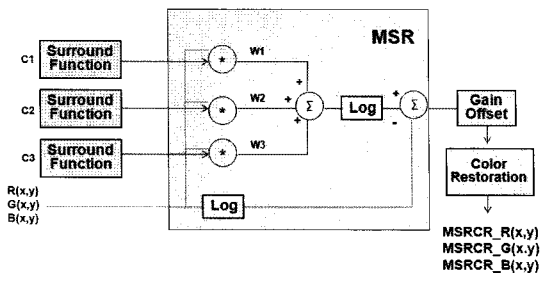
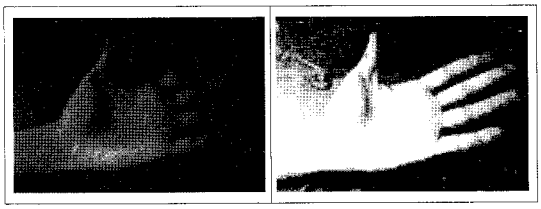


Fig. 10. Multi Scale Retinex flowchart.



(a) (b)

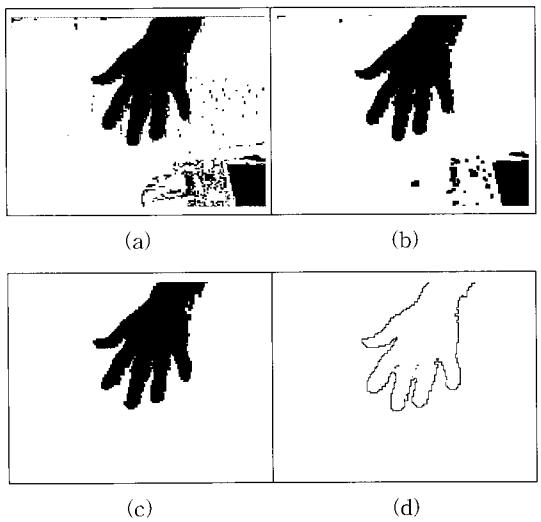
Fig. 11. Result of Multiscale Retinex (a) Input image (b) Retinex output.

the input image which lost its color due to the darkness.

We use the skin color of the hand to detect hand rather than the shape of the hand since the shape of the hand is changed frequently. After obtaining compensated Retinex output, we transform the Retinex output to HSI color model and YCbCr color model to extract skin color. Usually, the thresholds used to HSI model and YCbCr model were obtained based on the experimental results [8]. We follow the same method to get the thresholds.

4.4.2 Preprocessing the Image

After finding the areas having skin color, we binarize hand area as black. Fig. 12.(a) shows the binarized image of the hand area. However, those



(a) (b)

(c) (d)

Fig. 12. Result of Preprocessing. (a) Binary image of skin color area (b) Closing operation (c) Image segmentation (d) Thinning.

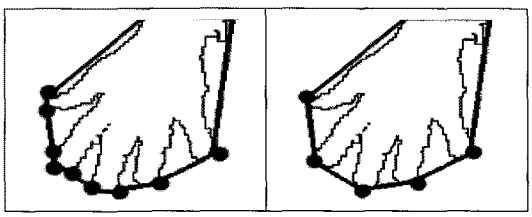
areas include hand area, noise artifacts and other objects, too. We need to remove noise artifacts and other objects. First, we apply morphological image process, closing operation, which applies dilation and erosion consecutively. Fig. 12(b) shows the result of the closing operation. After taking closing operation, we take labeling procedure with Grassfire algorithm and then segment the largest area as the hand area. Fig. 12(c) shows the result of the segmentation. Finally, we detect the boundary of the hand. Fig. 12(d) shows the result of the boundary detection.

4.4.3 Detecting Fingertips

We need the fingertips to find the direction of the hand and mouse point in the hand when we use the hand as the mouse. Graham scan algorithm is applied to compute the convex hull of the hand. Fig. 13(a) shows the convex hull and vertices on it. The number of vertices is too many for indicating fingertip of each finger. Vertices existing within a certain distance are removed according to the distance to the center of mass. Comparing these distances, we remove the vertex which has shorter distance. Fig. 13(b) shows the result of removing redundant vertices at the fingertip.

4.4.4 Detecting the Direction of the Hand

To use the hand as a mouse, we need mouse point and click function. We will use the fingertip of the index finger as the mouse point. Also, we use the angle between the thumb and index finger for checking click event. If hand is moving with the angle zero, it indicates that mouse is moving. When hand is stopped at one point and angle becomes large and then goes back to zero, then it indicates that mouse is clicked at the point.



(a) (b)

Fig. 13. Convex hull (a) Before removing vertices (b) After removing vertices.

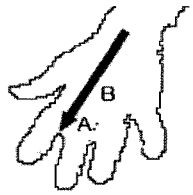


Fig. 14. Direction of the hand.

To search for the direction of the hand, we calculate two centers of the mass. One is the center of the hand and the other is the center of the convex hull of the hand. Since the convex hull is a polygon whose vertices are the fingertips of the fingers, the center of the convex hull is closer to the fingertip than the center of the hand. In Fig. 14, B is the center of the hand and A is the center of the convex hull. Clearly, A is closer to the fingertip than B. Connecting points from B to A gives the direction of the hand.

4.4.5 Creating Mouse Point

In Fig. 15, the hand shape is given when hand is used as a mouse. Vertex C is use as the mouse point, which can be obtained as follows. Vector BA is the direction of the hand. We calculate $\cos\theta$ between vector BA and vector B to fingertips of convex hull. As the angle between BA and vector from B to vertices decreases, the value of $\cos\theta$ increases. Since the direction of the hand and the direction from B to vertex at index finger is similar, $\cos\theta$ becomes near to 1. We can find point C by finding maximum value of $\cos\theta$ between vector

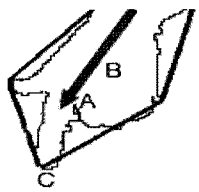


Fig. 15. Hand shape with mouse point

BA and vector from B to vertices of convex hull. The $\cos\theta$ is calculated with easy by using equation (2).

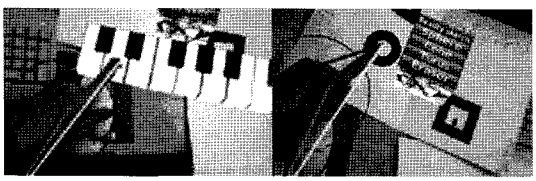
$$\cos\theta = \frac{a_1c_1 + a_2c_2}{\sqrt{a_1^2 + a_2^2}\sqrt{c_1^2 + c_2^2}} \quad (2)$$

Here, (a_1,a_2) is the directional vector of BA and (c_1, c_2) is the directional vector of BC.

4.4.6 Counting the number of fingers

To operate hand as mouse, we use the fingers in Fig. 15. Since the number of fingers changing from two to one indicates that mouse is clicked, it is indispensable to check the number of fingers in Fig. 15. We can see the method how to count the number of fingers in Fig. 16. Simply draw the circle with appropriate radius and count the number of change of the intensity from 255 0. The number of changes is the number of the fingers in the Fig. 16. This enables to operate AR system by picking entries in the menu. In Fig. 17, yellow line is the directional vector from the origin of the mouse point and green line is the directional vector of the center of the convex hull and blue line is the directional vector of the center of the hand. In [9], user wearing HMD draws virtual object with pen at the 3D space. In this paper, we propose the

Fig. 16. Counting the number of fingers



(a) (b)

Fig. 17. Interaction between hand and virtual objects (a) Playing virtual piano (b) Drawing virtual circle.

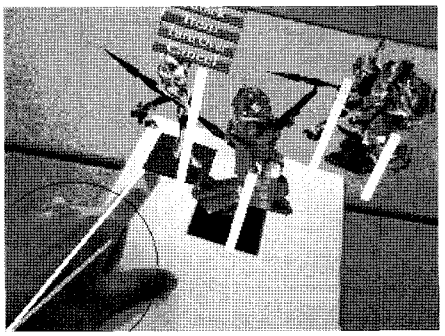


Fig. 18. Final game scene

usage of the hand to play virtual piano as in Fig. 17(a) and to draw virtual circle as in Fig. 17(b). Thus, the interaction between hand and virtual objects can be applied to many areas.

4.6 Implementing Strategy Game

We see the final game scene in Fig. 18. Vertical yellow lines display the remaining power of the virtual characters. If character is clicked by the mouse point of the hand, the position is turned into red and the character located at the position is picked and operated by choosing the menu entry.

5. CONCLUSIONS

In this paper, we implemented online real time strategy game at the desktop environment. Especially, we introduced the hand tracking method for interaction between user and the virtual object and virtual menu. This enables to extend the applicability of the proposed method to the mobile phone or handheld instrument [10,11]. This extension enables us to play online real time strategy game in AR system with other instrument through wireless communication. The proposed method can applicable to many areas such as online education, remote medical treatment and mobile interactive game.

We have two areas for further research. One is the introducing voice interface for controlling virtual objects and the other is the extension of the

application to the handheld instruments.

REFERENCES

[1] ARToolKit, http://www.hitl.washington.edu/artoolkit

[2] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, 125-134, 2008.

[3] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.29, No.6, June 2007.

[4] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," *Proc. International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, 2007.

[5] JinKook Kim and Jongwon Lee, "Board Game Framework in AR," *Journal of Information Science and Engineering (Korean)*, Vol.26, No.3, pp. 40-45, 2008. 3.

[6] H. Jones and M. Snyder, "Supervisory control of Multiple Robots Based on a Real-Time Strategy Game Interaction Paradigm.," *IEEE International Conference on Systems, Man, and Cybernetics*, Vol.1, pp. 383-388, 2001.

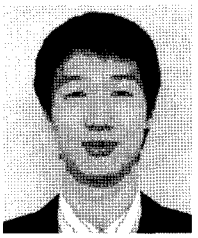
[7] Kil, Hyejin and Yi, Juneho, "Real-Time Skin Co;or Segmentation Using a Retinex Algorithm and Principle Component Analysis," *Journal of Sungkyunkwan University, Science and Tecnology*, Vol.53, No.2, pp. 19-34, 2002.

[8] F. Gasparini and R. Schettini, "Skin segmentation using multiple thresholding," *Proceedings of the SPIE - The international Society for Optical Engineering*, Vol.6061, pp. 60610F, 2006.

[9] G. Raphael, B. Laurence, G. Jean-Dominique, and D. Schmalstieg, "Interactive Mediated Reality," *Proceedings of the Sixth Australasian conference on User interface*, Vol.40, pp. 21-29, 2005.

[10] Squire, Kurt D. Jan, and Mingfong, "Mad City Mystery: Developing Scientific Argumentation Skills with a Place-based Augmented Reality Game on Handheld Computers," *Journal of science education and technology*, Vol.16 No.1, pp. 5-29, 2007.

[11] Won Hyung Kang, "A study on Handheld Game System in Augmented Reality using Dynamic Environment," *KAIST, Master thesis*, 2007.



Gwangha Jeon

2009 Samsung Electronic Software Membership

2010 Dept. of Multimedia Engineering, Hansung University (B.S)

Interesting Area : Computer Vision, Augmented Reality, Game programming.

Jong seok Um

1982 Dept. of Applied Statistics, Yonsei University (B.S.)

1984 Dept. of Applied Statistics, Yonsei University (M.S.)

1991 Statistics, Ohio State University (Ph.D)

1992~Present Professor, Hansung University, Dept. of Multimedia Engineering

Interesting Area : Computer Vision, Pattern Recognition, Data Mining, Computer Graphics