

# A Study on Multiple Objects Detection, Loading and Control in Video for Augmented Reality

Sungmo Jung, Jae-gu Song, Seoksoo Kim

Department of Multimedia

Hannam University

133 Ojeong-dong, Daedeok-gu, Daejeon-city

Korea

sungmoj@gmail.com, bhas9@paran.com, sskim0123@naver.com

*Abstract:* - Since the researches on augmented reality have recently received attention, M2M market started to be activated and there are numerous efforts to apply this to the real life in all sectors of society. However, with the existing marker-based augmented reality technology, a designated object only can be loaded from one marker and one marker has to be added to load the same object additionally. To solve this problem, the relevant marker should be extracted and printed in screen so that loading of the multiple objects is enabled. However, since the distance between markers will not be measured in the process of detecting and copying markers, the markers can be overlapped and thus the objects would not be augmented. To solve this problem, a circle having the longest radius needs to be created from a focal point of a marker to be copied, so that no object is copied within the confines of the circle.

Therefore, in this paper, multiple objects detection and loading using PPHT has been developed. And markers overlapping control according to multiple objects control has been studied using Bresenham and Mean Shift algorithm.

*Key-Words:* - Multiple Objects Detection, Multiple Objects Loading, Multiple Objects Control, Computer Vision, PPHT, Bresenham Algorithm, Mean Shift Algorithm, Augmented Reality.

## 1 Introduction

Augmented reality(AR) [1] refers to a virtual world which can provide a sense of reality in interaction with users, where images of real world and virtual graphic objects are generated by computer, and it is almost a lower level of virtual reality rather than a virtual reality realizing complete immersion environment. Core technologies to establish augmented reality system include marker detection to locate an object, matching, tracking, camera calibration to synthesize virtual objects into actual images and calculate camera variables, display and 3D modeling technology [2]. That is, it is a technology field that provides the image with information such as characters and graphic objects in real life to enhance users' understanding of the situation of moving objects in 3D space [3].

At first, AR system had been developed using HMD (Head Mounted Display) [4] by Ivan Sutherland in 1968, which could only display some very simple wire frames in real time, due to the limitations of the computer performance at the time [5]. After that, in 1992, Tom Caudell developed the display technology using HMD in earnest and began to use the term, AR, and since then, consistent

research has been conducted. In modern times, as the performance of easy-to-carry computing terminal and display environment has been developed, the AR technology has been rapidly developed, being used in many areas including medical care, education, game, military and industry etc., and researches on AR technology are actively being conducted with consistently increasing demand. For the realization of AR, real life, what point of the image captured by camera the object should be augmented needs to be decided and augmentation method is classified into marker recognition method and markerless recognition method. Since the markerless recognition method requires feature points which are extracted from the image captured by camera, much time and labor are needed for image processing [6]. Besides many researches need to be done to increase extraction rate and the researches are actively on progress in the country. The marker recognition method is a method of augmenting an object in a point where the computer recognizes the tag interface serving as a pointer connecting real world and virtual world, which is called a marker. This method, due to high

recognition rate and easy use, is widely used in the realization of AR [7].

The existing marker-based AR technology only can load an object designated in one marker and actually has to add one marker to load the same object additionally. To solve this problem, the relevant marker should be extracted and printed in screen so that loading of the multiple objects is enabled.

However, this marker-based augmented reality uses PPHT (Progressive Probabilistic Hough Transform)[8]–based multiple objects loading technology to detect and copy the same marker, and in case only this method is used, markers can be overlapped and therefore the object will not be augmented.

For that reason, in this paper, AR multiple objects loading technology has been developed. And utilizing Bresenham algorithm for augmented reality multiple objects loading, an area was designated to prevent markers overlapping based on a focal point of the marker and markers overlapping control was made possible.

## 2 Related Researches

### 2.1 Hough Transform

Hough transform is a method of detecting straight lines, circles or other simple shapes out of the image [9]. The Hough transform is based on the fact that there are countless straight lines passing through any point in the binary image [10].

If a line has a y-intercept of  $b$  and a slope of  $a$  ( $y = ax + b$ ), a point on the  $(x, y)$  coordinate plane is expressed as a straight line on the  $(a, b)$  coordinate plane. In case no zero pixels in the input image are all expressed as straight lines on the  $(a, b)$  coordinate plane image and the pixel values located where the lines pass through are accumulated, a line on the  $(x, y)$  coordinate plane is shown to have a local maximum value on the  $(a, b)$  coordinate plane. Since all loci of the lines expressed by each point are summed, the  $(a, b)$  coordinate plane is commonly called an accumulator plane [11].

However, the method of expressing a straight line with a slope and a y-intercept is not appropriate to express all straight lines on the  $(x, y)$  coordinate plane. This is because to express the common straight lines of which slopes range from  $-\infty$  to  $+\infty$  in the binary image is difficult.

Therefore, in practical realization, it is expressed as a point,  $(\rho, \theta)$  on the polar coordinate system by means of other method. A line expressed as  $(\rho, \theta)$  on the polar coordinate system means a straight line perpendicular to a line passing through this point and the origin, which is expressed in the following numerical formula (1):

$$\rho = x \cos \theta + y \sin \theta \quad (1)$$

A point  $(x_0, y_0)$  (a) on the image coordinate system becomes a point (b) where many straight lines expressed as  $(\rho, \theta)$  on the polar coordinate system intersect and it can be expressed as a line (c) on the  $(\rho, \theta)$  plane as follows;

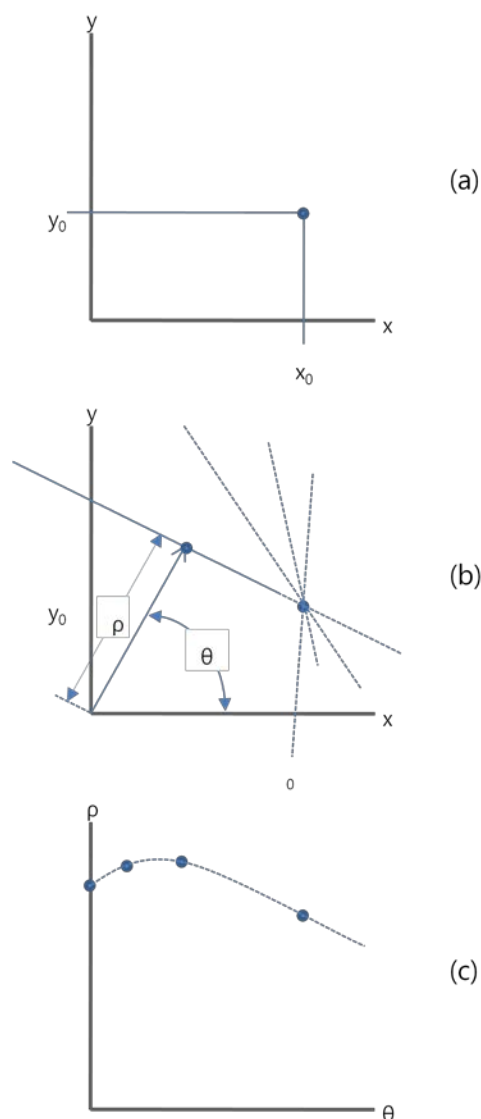


Fig. 1 Common Accumulator Plane expressed on the Polar Coordinate System

In this paper, reflecting local maximum values on  $(\rho, \theta)$  coordinate plane, the marker recognition method was expanded to include not only a simple straight line but other general shapes.

## 2.2 Bresenham Algorithm

Most of the various line algorithms require floating point division operations having low performance. However, the rasterization which was proposed by Bresenham was a superhigh speed algorithm rasterizes straight line/circle/ellipse only with addition/subtraction operations of pure integers [12].

The fundamentals of the Bresenham algorithm [13] is accumulated 'error terms'. That is, the computer screen corresponds to a great 2-dimensional array of colored squares. An act of drawing a straight line (or another fig.) at this point is a process of 'approximating' a continuous coordinate system of real numbers into a discontinuous coordinate system. Therefore, a dot usually provides an error between discontinuous coordinate systems printed in the original coordinate system of the line and the screen, and the key concept of Bresenham algorithm is to select a dot which best minimizes this error by drawing the straight lines (or another fig.s).

The process of drawing lines using Bresenham algorithm is as follows;

- 1) Array  $p1$  and  $p2$ , the two dots representing the line in the order of coordinate axes. At this point, if the slope of the line is less than 1, array in increasing order of coordinate  $x$  and if greater than 1, array in increasing order of coordinate  $y$ . (Here assumed to be arrayed to increasing order of  $x$ )
- 2) Begin with the first dot out of the dots arrayed.
- 3) Make a dot at the present location.
- 4) Provide setting the next location. Then, increase the location of the present pixel by one to increasing order of coordinate  $x$ .
- 5) Calculate the error value at the next location. Here, the error term is the addition of the differences between  $y$  coordinate values of  $p1$  and  $p2$ .
- 6) Compare the error terms and examine if the error portion is greater than one pixel. That is, after comparing the error terms up to now and the difference between  $x$  coordinate values of  $p1$  and  $p2$ , increase the coordinate value by one to increasing order of coordinate  $y$  if the error term is greater than the difference.
- 7) Repeat 3) to 6) until the last coordinate is dotted.

For drawing of a quadrangle, the process of drawing four lines using Bresenham algorithm is repeated.

The process of drawing a circle represented by the equation,  $x^2 + y^2 = r^2$ , the fundamental of algorithm to be used in this paper is as follows;

- 1) Begin with a fixed point on the top of the circle. Here, draw a quarter circle clockwise and repeat this circle four times.
- 2) Make a dot in the present coordinate.
- 3) Increase the coordinate by one to increasing order of coordinate  $x$ .
- 4) Then decide  $y$  coordinate. Decide one out of  $y$  or  $y-1$  for  $y$  coordinate. If  $x^2 + (y-1)^2 < x^2 + y^2 < r^2$  is valid,  $y$  becomes the next coordinate and if  $r^2 < x^2 + (y-1)^2 < x^2 + y^2$  is valid,  $y-1$  becomes the one. In other cases except for these, the error can be ignored with any dot between the two.
- 5) Repeat 2) to 4) until  $x=y$ .
- 6) Begin with the fixed point on the right side of the circle.
- 7) Make a dot in the present coordinate.
- 8) Increase the coordinate by one to increasing order of coordinate  $y$ .
- 9) Then decide  $x$  coordinate. Decide one out of  $x$  or  $x-1$  for  $x$  coordinate. If  $(x-1)^2 + y^2 < x^2 + y^2 < r^2$  is valid,  $x$  becomes the next coordinate and if  $r^2 < (x-1)^2 + y^2 < x^2 + y^2$  is valid,  $x-1$  becomes the one. In other cases except for these, the error can be ignored with any dot between the two.
- 10) Repeat 7) to 9) until  $x=y$ .

The algorithm of drawing an ellipse to be used in this paper is similar to the case of a circle, and the algorithm was applied by changing the equation and calculating the point turning over the direction of axis.

## 2.3 Mean Shift Algorithm

Mean shift algorithm [14] is a nonparametric statistical method for detecting a major mode among sample points distributed. The multivariate kernel density estimator under the condition of  $R^d$  ( $d$  dimensional space) and  $\{x_i\}_{i=1, \dots, n}$  ( $n$  number of data set) is as follows.

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (2)$$

Here,  $K(x)$  refers to the kernel and  $h$ , a radius of a window. The mode that density  $f(x)$  becomes a

local maximum is located where gradient  $\nabla f(x)$  is 0. This is expressed as follows.

$$g(x) = -k'(x) \quad (3)$$

Here,  $k(x)$  is a profile of kernel  $K(x)$ . If  $g(x)$  is a profile of kernel  $G(x)$ ,  $G$  is called a shadow of kernel  $K$  [15].

Detecting a mode of density means estimating density to find out a convergent point of the mean shift. Thus, kernel  $G$  is used in order to find out a convergent point where the difference between  $x$  and a sample mean is 0. The following formula shows this.

$$m_{h,G(x)} = \frac{\sum_{i=1}^n x_i g(\|\frac{x-x_i}{h}\|^2)}{\sum_{i=1}^n g(\|\frac{x-x_i}{h}\|^2)} - x \quad (4)$$

Mean shift  $m_{h,G(x)}$  is proportional to a gradient of normalized density and always changes in the direction that a density function shows the greatest increase.

Formula (4) can be expressed as Formula (5) by using shadow kernel  $G$  to express a series of locations of the sample means as  $\{y_j\}_{j=1,2,\dots}$ .

$$y_{j+1} = \frac{\sum_{i=1}^n x_i g(\|\frac{x-x_i}{h}\|^2)}{\sum_{i=1}^n g(\|\frac{x-x_i}{h}\|^2)} \quad (5)$$

Here,  $y_{j+1}$  refers to the weighted mean at  $y_i$ , calculated by kernel  $G$ , and  $y_1$ , initial center of kernel position. The mean shift algorithm repeats the following process so as to find out the potential centers of the cluster [16].

- 1) Calculation of mean shift vector  $m_{h,g(x)}$
- 2) Renewal of  $y_j$ , position of the current anchor point

In this paper, based on the concept of the mean shift algorithm as well as "Motion segmentation and analysis in video sequences [17]" and "A graph-segment-based unsupervised classification for multispectral remote sensing images [18]", numerous micro-domains such as noises are removed from a final outcome image by effectively combining image domains, taking into account that objects should be augmented by markers when loading/controlling multi objects. Also, the study suggests that a size of a domain be limited for minimum merger of adjacent domains.

### 3 Multiple objects detection and loading

#### 3.1 Marker detection using PPHT

In this paper, 8 bit input image is processed into binary image, no zero pixels are handled all equally and an  $N \times 1$  sized matrix is stored in the memory storage as a pointer of the space where the result of Hough transform is to be stored. The most important thing in case of detecting a marker using PPHT, is to set a critical value to estimate it as a straight line on the accumulator plane. This paper set the critical value to optimize this in experimental condition and derived the result image.

The methods of implementing the edge tracing algorithm to achieve contour information of the marker's contour out of the result image derived, and extracting edge points from the contour information existing within the marker region are as followings;

First, if all edge points inside or outside of the marker region appear, one point out of tag region's contour information is selected and an edge point locating farthest from the point is extracted by means of Pythagoras' theorem as shown in Fig. 2(a). Next, an edge point lying farthest from the extracted point is investigated as in Fig. 2(b), and then an edge point locating farthest from the obtained edge point is extracted again as in Fig. 2(c). As illustrated in Fig. 2(d), a point which maximizes the area of the square becomes the rest one point. Based on the four points extracted, the marker ROI (region of interest) in marker tag can be extracted as shown in Fig. 2(e).

*RectArea*

$$\begin{aligned} &= \frac{1}{2}(x_1y_2 - x_2y_1 + x_2y_3 - x_3y_2 + \dots + x_ny_1 - x_1y_n) \\ &= \frac{1}{2}x_1(y_2 - y_n) + x_2(y_3 - y_1) + \dots + x_n(y_1 - y_{n-1}) \quad (6) \end{aligned}$$

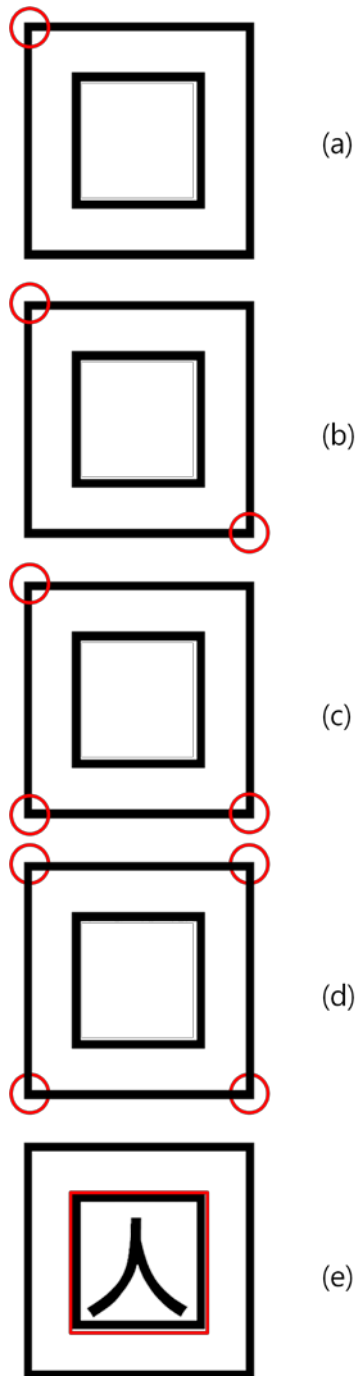


Fig. 2 Edge Points and Detection of Tag ROI

PPHT (Progressive Probabilistic Hough Transform)[19] which is to be finally used for marker detection can calculate the beginning and the end of each line as well as the direction of line. This method only considers some voluntarily selected edge pixels, not increasing values for all edge pixels on the accumulator plane. In case a value in a specific location appears to be higher than the critical value, no more of value increase is allowed, and the edge pixels displayed by this point into a straight line will be ignored in the further

calculation. Besides, since this method can reduce the calculation time drastically, it is highly effective in real time marker detection.

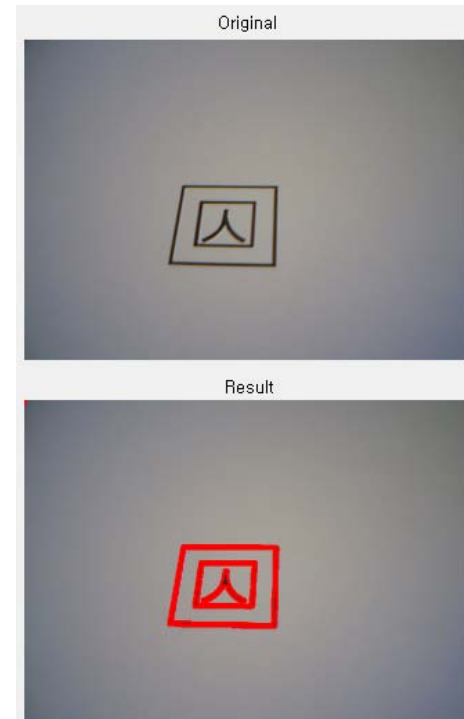


Fig. 3 Marker Detection using PPHT

The above fig. shows the tag detected using PPHT, from the original image captured from Webcam. For easy distinction, the detected parts are marked in bold.

### 3.2 Video output of replicated marker

This research suggests a template matching method that makes warping into the marker registered in a square form from edge points which are extracted within the image, and performs matching.

First of all, when the four edge points outside the marker are designated as  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ ,  $(x_4, y_4)$  respectively, a point where the straight line passing through  $(x_1, y_1)$  and  $(x_3, y_3)$ , and the line through  $(x_2, y_2)$  and  $(x_4, y_4)$  intersect becomes the center of the marker. The formula to calculate the center of the marker using the equation of straight line is as below;

$$C_y = \left( \frac{y_3 - y_1}{x_3 - x_1} \right) (C_x - x_1) + y_1 \quad (7)$$

$$C_y = \left( \frac{y_4 - y_2}{x_4 - x_2} \right) (C_x - x_2) + y_2 \quad (8)$$

$C_x$  is obtained by calculating formulas (7) - (8) and then  $C_y$  is obtained by substituting this for formula (7). If the height of the tag of a specific

standard and the height of the extracted marker tag set to  $h_r$  and  $h_d$  respectively, the *Sdiff* (*Scale difference ratio*) according to the change of plane for the extraction can be calculated as the following formula (9).

$$S_{diff} = \frac{h_d}{h_r} \cdot 100 \quad (9)$$

If the center of the marker which is finally calculated is estimated as the origin, the size error correction ( $T_{total}$ ) can be expressed as the formula (10), and  $D_T$  here refers to transform matrix.

$$T_{total} = \begin{bmatrix} \frac{1}{S_{diff}} & 0 & 0 \\ 0 & \frac{1}{S_{diff}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot D_T \quad (10)$$

The fig. below shows that a marker tag extracted from the image is copied and moved to the desired location.

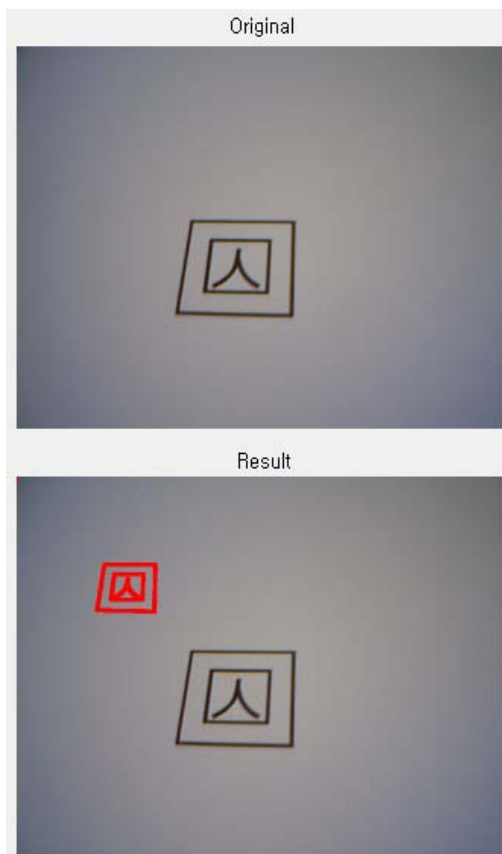


Fig. 4 Copying and Moving of the Detected Marker

The binarized image of the marker marked for the function of a marker is as shown below.

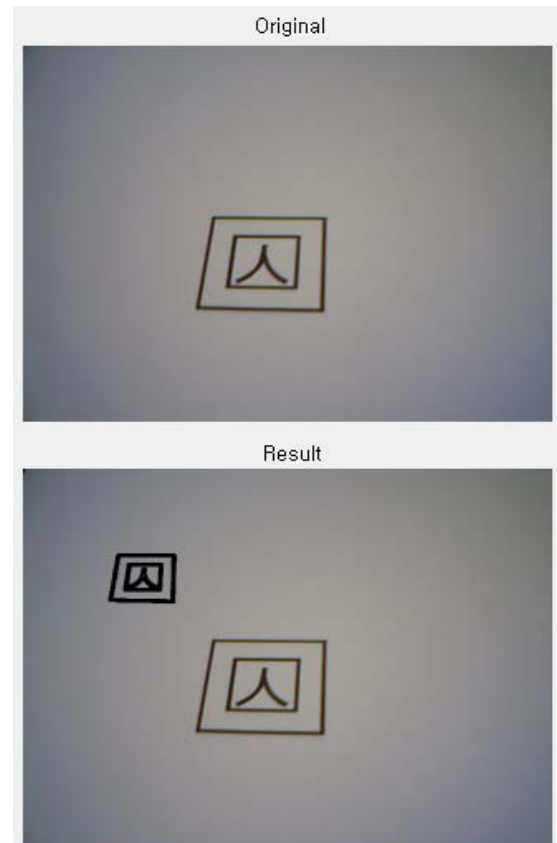


Fig. 5 Binarization of the Marker Copied and Moved

#### 4 Deduction of markers overlapping problem

This paper is based on the assumption that 8-bit input image is processed in binary image data, non-zero pixels are all treated as the same and  $N \cdot I$  matrix is stored in a memory storage for the pointer of a space where copied markers will be stored. Fig. 6(a) shows the process of detecting the marker using PPHT and copying one marker. Two markers are copied through multi markers copying as shown in Fig. 6(b). In this way, n markers can be copied and therefore objects can be augmented, but if markers are overlapped as in Fig. 6(c), the object cannot be augmented.

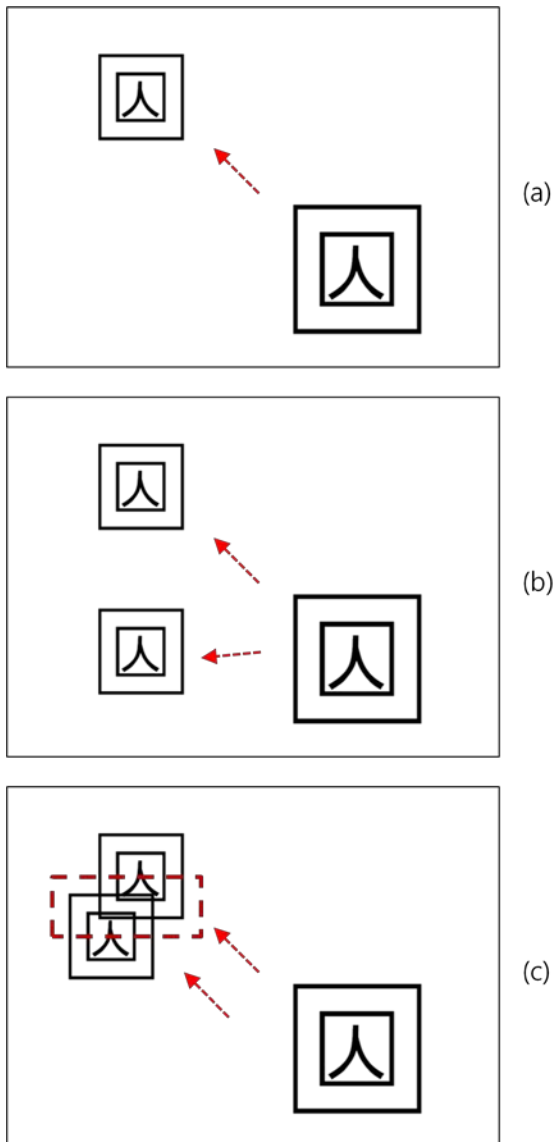


Fig. 6 Deduction of markers overlapping problem

Therefore in this paper, to control markers overlapping problem, the marker area was set up with Bresenham algorithm applied and the problem was solved by designating non-overlapping area based on the marker's focal point.

## 5 How to create marker area by applying

### 5.1 Calculation of the marker's focal point through detection of edge points

For the calculation of the marker's focal point, the contour information of the marker area in the image was acquired by conducting the edge tracing algorithm and the edge points were extracted from the contour information inside the marker area.

The process of extracting the marker's focal point with the above formula is as following fig..

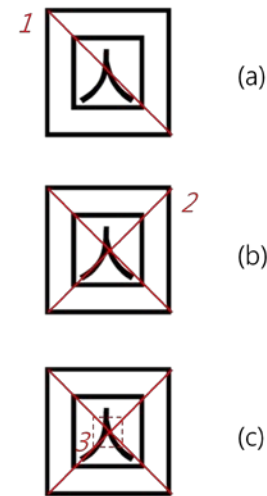


Fig. 7 Extraction of marker's focal point

### 5.2 Creation of non-overlapping area, based on marker's focal point

When calculating a circle's area from the marker's focal point in a square, the area should be divided every 45 degree. However, the marker extracted from the image is not recognized as a 100% circle, an ellipse's area needs to be considered. Since an ellipse cannot be calculated or divided into every 45 degree, when considering a line drawn at a point with a gentle slope, the slope  $\frac{dy}{dx}$  should be regarded as 1 or -1.

In elliptic equation,  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ , a point with the slope of -1 is  $x = \frac{a^2}{\sqrt{a^2 + b^2}}$ ,  $y = \frac{b^2}{\sqrt{a^2 + b^2}}$ .

Therefore,  $x$  should be an independent variable to this point and  $y$  should be one for the remained part. Other quadrants are drawn using its symmetry properties.

If the point is  $(x, y)$ , the next point possible is  $(x+1, y)$  or  $(x+1, y-1)$ , and it is determined by seeing whether the middle point of these points is fallen inside or outside of the ellipse.

The following fig. describes the process of creating an ellipse with the discriminant,  $d = F(x + 1, \frac{y-1}{2})$ , for  $F(x, y) = b^2 \cdot x^2 + a^2 \cdot y^2 - a^2 \cdot b^2$ .



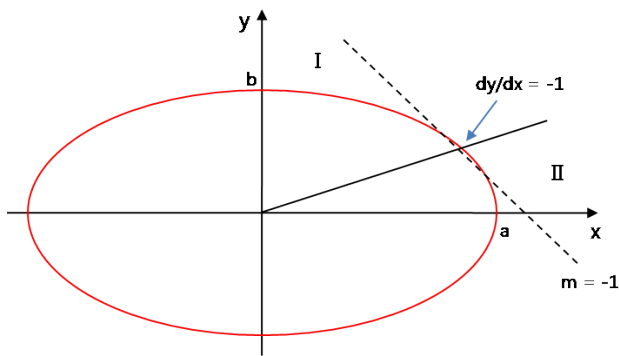


Fig. 4. Basic structure to create an ellipse

If  $d$  value is less than 0,  $(x+1, y)$  and greater than 0,  $(x+1, y-1)$  is selected. For the update of the discriminant to be calculated on the marker's focal point coordinate, if  $d_{old}$  is less than 0, it is  $d_{new} = F(x+2, y) = d_{old} + b^2 \cdot (2 \cdot (x+1) + 1)$  and if  $d_{old}$  is greater than 0, it is  $d_{new} = F(x+2, \frac{y-3}{2}) = d_{old} + b^2 \cdot (2 \cdot (x+1) + 1) - a^2(2 \cdot (y-1))$ .

As the independent variables of  $x$  are between 0 and -1 of the slope, for an interval of independent variables in  $y$ , the roles are just changed in the numerical formulas calculating independent variables in  $x$ . The following fig. shows the process of creating an ellipse's area based on the marker's focal point.

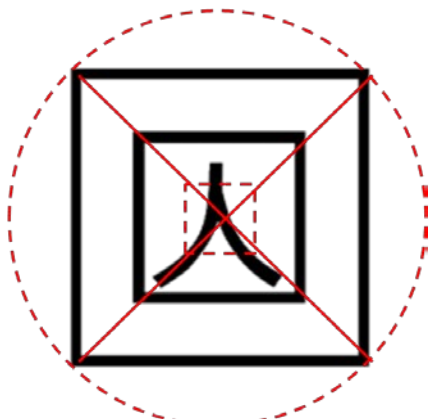


Fig. 8 Creation of marker's focal point-based area

In case two markers are copied after the area created is designated as non-overlapping area, the applicable area is not interfered as in Fig. 9(a). Besides, if more than two markers are to be copied, as the number of markers printed in one screen is automatically limited, the markers overlapping problem can be solved and the number of copying operations can be regulated according to the scale of the marker.

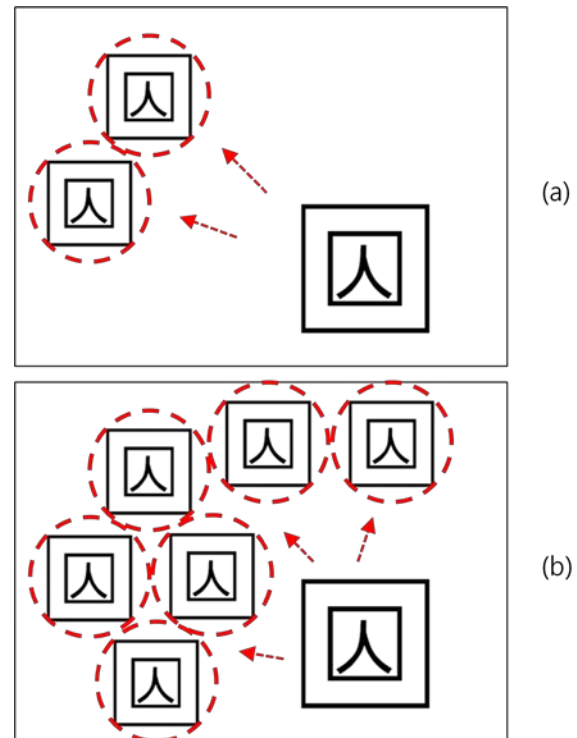


Fig. 9 Overlapping control during marker creation, by designating the created areas as non-overlapping areas

### 5.3 Removal of Micro-Domains Using the Mean Shift Algorithm

Image segmentation using the mean shift algorithm can produce the best result when sizes of a space window and a keypoint window are properly selected. However, if the sizes of the two windows are small, there can be excessive micro-domains, resulting in excessive segmentation also.

Hence, in order to overcome such a problem, this study suggests a method that domains in an excessively segmented image are merged using critical values and boundaries of merged domains are preserved.

The processing of an input image is largely divided into two steps: segmentation by the mean shift algorithm and merger of the outcome images.

The first step, segmentation by the mean shift algorithm, is as follows.

- 1) Convert an RGB color domain of an input image into a CIE LUV color domain
- 2) Map each pixel of the converted image by a joint domain of a spatial-range domain
- 3) Classify the mapped pixels into clusters. Classify pixels within a window in a joint domain into a single cluster of those with a similar keypoint.
- 4) Calculate a mean of a cluster
- 5) Calculate the value of the mean shift



- 6) Find the convergent point. If can't, move the center of a window to the mean calculated by 4) and repeat 3)~6). End the segmentation process if the convergent point was found.

Image segmentation using the mean shift algorithm could be done excessively according to sizes of a window. The suggested method converts an RGB color domain into an HIS[20] color domain for merger of domains in an excessively segmented image. This method is frequently used for color segmentation, for it is less affected by non-uniform illumination such as shadow, projection, and reflected light. However, if domains are merged by colors, objects of different adjacent domains but of the same color could be merged, which could obscure boundaries of major domains. In addition, most domains with low brightness are those of shadow and domains of low chroma are dim areas. Therefore, there are color values not perceived in a dark or dim area, which may cause inaccurate merger of domains.

This research, therefore, applies limiters for merger of domains to solve the problem. By applying limits to features existing in the domain of the boundaries, the boundaries are preserved and influences of low brightness and chromas are minimized. Such limits include chroma, brightness, and an edge of an HIS color domain. The edge is calculated by applying the Sobel operation to the brightness of an HIS color domain. If brightness and chroma of a pixel to be processed is lower than that of a critical value, respectively, or if the edge is higher than a critical value, the pixel is not merged.

According to the suggested method, limiting segments of boundaries are applied to a pixel to be processed in the initial labeled domain prior to merger of domains. If any restrictive condition is applicable to a pixel, that pixel is not treated for merger. Also, If no restrictive condition is applicable, adjacent domains of the same color are merged. The following formula shows a restrictive condition ( $FR$ ) while  $P_i$ ,  $P_s$ , and  $P_e$  refers to brightness, chroma, and edge of a pixel to be processed and  $Th_i$ ,  $Th_s$ , and  $Th_e$ , critical value of the brightness, chroma, and edge.

$$FR = \begin{cases} \text{not process, } P_i < Th_i \text{ or } P_s < Th_s \text{ or } P_e > Th_e \\ \text{merging process, otherwise} \end{cases} \quad (11)$$

After merger of domains, there are many micro-domains such as a noise in the outcome image. This study suggests that the area of a domain be limited. Using the histogram method, frequency of a domain label is examined and a domain smaller than a

critical value is merged with others through Euclidean distance. That is, domains are merged so that chrominance distance is minimized. The following formula shows the Euclidean distance when the color of pixel  $c_1$  and  $c_2$  is referred to as  $(R_1, G_1, B_1)$  and  $(R_2, G_2, B_2)$ .

$$d_E(c_1, c_2) = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (12)$$

The following picture is the final outcome.

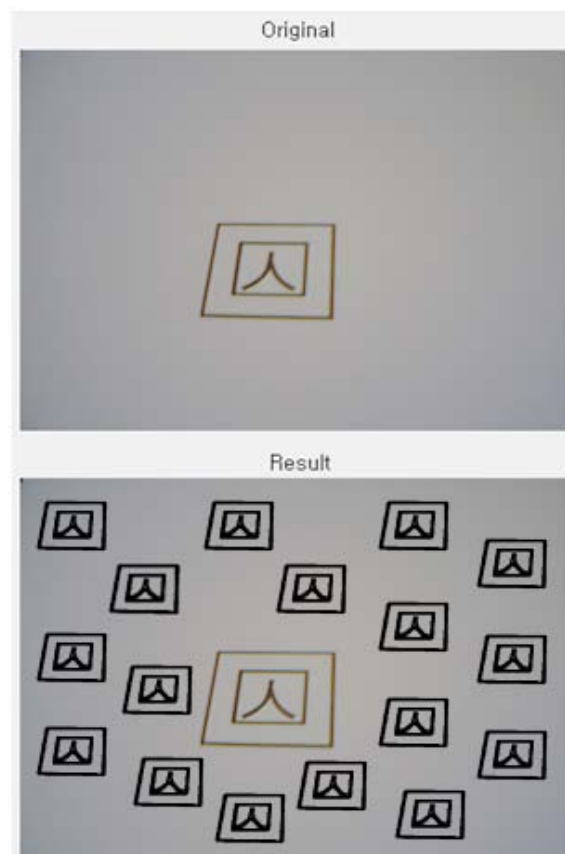


Fig. 10 Multiple objects loading after removing micro-domains

## 6 Conclusion

In this paper, to solve the problem of the existing marker-based AR technology, one marker to one object loading, PPHT-based AR multiple objects loading technology which enables multiple objects loading without another marker added has been developed, by extracting the same marker from the image and copying it at a desirable location. And markers overlapping problem was deducted in PPHT-based augmented reality multiple objects loading technology which realized one marker N objects loading. To solve the problem deducted, the method of setting marker area using Bresenham and Mean Shift algorithm was researched, and the

methods of calculating the marker's focal point through detection of edge points and of creating non-overlapping area based on the marker's focal point were suggested.

This research has the advantage that it can solve the problems that may occur during marker copying and limit the number of markers printed in one screen automatically. And this research can be most widely used since it is applicable to the researches on feature points extraction of not only the marker-based AR but markerless-based AR.

Based on this research, an application that enables real objects loading will be added to the future researches. Besides, researches on an algorithm which reduces more marker detection time and a system that implements marker detection smoothly even in an environment that users cannot predict are required to be carried out. And a plan to increase or decrease the number of markers printed in one screen should be prepared in the further researches through the development of a technology of regulating the scale during the marker copying. Because of a real time, the nature of augmented reality, a research of very fast and correct algorithm is required.

### Corresponding author

Seoksoo Kim(sskim0123@naver.com)

### References:

- [1] Azuma, Ronald T., *A Survey of Augmented Reality*, Teleoperators and Virtual Environments Vol. 6, No. 4, 1997, pp. 355-385.
- [2] Gabor Sziebig, *Achieving total immersion: technology trends behind augmented reality-a survey*, Proceedings of the 9th WSEAS international conference on Simulation, modelling and optimization, 2009, pp.458-463.
- [3] T.P. Caudell and D.W. Mizell, *Augmented reality:an application of heads-up display technology to manual manufacturing processes*, IEEE Comput.Soc. Press, Vol. 2, 1992, pp. 659 - 669.
- [4] I. Sutherland, *A Head-Mounted Three Dimensional Display*, Proceedings of Fall Joint Computer Conference, 1968, pp. 757-764.
- [5] Shin-Ichiro Kaneko, Yasuo Nasu, Shungo Usui, Mitsuhiro Yamano, Kazuhisa Mitobe, *Development of a teleoperation system and an operation assist user interface for a humanoid robot*, Proceedings of the 9th WSEAS International Conference on Systems, 2005, Article No.53.
- [6] A.I. Comport, E. Marchand, and F. Chaumette, *A real-time tracker for markerless augmented reality*, International Symposium on Mixed and Augmented Reality, 2003.
- [7] S. K. Feiner, S. MacIntyre and D. Seligmann, *Knowledge-Based Augmented Reality*, Communications of the ACM, Vol. 30, No. 7, 1993, pp. 53-62.
- [8] Fernandes, L.A.F. and Oliveira, M.M., *Real-time line detection through an improved Hough transform voting scheme*, Pattern Recognition, Elsevier, Volume 41, Issue 1, 2008, pp. 299-314.
- [9] R.O.Duda and P.E.Hart, *Pattern Recognition and Scene Analysis*, New York: Wiley, 1973.
- [10] Costin-Anton Boiangiu, Bogdan Raducanu, *Line detection techniques for automatic content conversion systems*, WSEAS Transactions on Information Science and Applications, Vol.5, Issue 7, 2008, pp.1200-1209.
- [11] Fernandes, L.A.F. and Oliveira, M.M., *Real-time line detection through an improved Hough transform voting scheme*, Pattern Recognition, Elsevier, Volume 41, Issue 1, 2008, pp. 299-314.
- [12] I. Jivet, B. Dragoi, *Performance analysis of direct digital synthesizer architecture with amplitude sequencing*, WSEAS Transactions on Circuits and Systems, Vol.7, Issue 1, 2008, pp.1-6.
- [13] J.E. Bresenham, *A linear algorithm for incremental digital display of circular arcs*, Commun. of the ACM, vol.20,no.2, 1977, pp100-106.
- [14] Dimitrije Bujakovic, Milenko Andric, *One approach to the analysis influence of change background statistical parameters on the capability of tracking objects using "mean shift" procedure*, WSEAS Transaction on Signal Processing, Vol.4, Issue 1, 2008, pp.1-5.
- [15] Y.Cheng, *Mean shift, mode seeking, and clustering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.12, 1995, pp.790-799.
- [16] C.J.Yang, R.Duraiswami, D.Dementhon, and L.Davis, *Mean-Shift Analysis using Auasi-Newton Methods*, University of Maryland, College Park, MD 20742.
- [17] Gui, Vasile, Alexa, Florin, Căleanu. Catalin, Fuiorea. Daniela, *Motion segmentation and analysis in video sequences*, WSEAS Transactions on Circuits and Systems. Vol.6, no.1, 2007, pp.142-148.
- [18] Nana Liu, Jingwen Li, Ning Li, *A graph-segment-based unsupervised classification for*

*multispectral remote sensing images*, WSEAS Transactions on Information Science and Applications, Vol.5, Issue 6, 2008, pp.929-938.

- [19] C. Galambos, J. Kittler, J. Matas, *Progressive Probabilistic Hough Transform for Line Detection*, cvpr vol. 1, 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99) - Volume 1, 1999, pp.1554.
- [20] H.D.Cheng, Y.Sun, *A hierarchical approach to color image segmentation using homogeneity*, IEEE Transaction on Image Processing, Vol.9, No.12, 2000, pp.2071-2082.