We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

4

A Robust Hand Recognition
In Varying Illumination

Yoo-Joo Choi[1], Je-Sung Lee[2] and We-Duke Cho[3]
[1]Seoul University of Venture and Information, [2]Korean German Institute of Technology,
[3]Ajou University
South Korea

1. Introduction

As ubiquitous computing provide up-graded smart environments where humans desire to create various types of interaction for many kinds of media and information, the research in the area of Human-Computer Interaction (HCI) is being emphasized to satisfy a more convenient user interface. In particular, the gesture interaction technique has been one of the important research areas under ubiquitous computing environment since it can only utilize widespread consumer video cameras and computer vision techniques without the aid of any other devices to grasp human movements and intentions(Park, 2004; Jung, 2007). Among the gesture interaction techniques, recognition of hand poses and gestures has especially received attention due to great potential to build various and user-centric computer interfaces. The applicability of hand pose recognition is very high in applications where system users can not use existing interface devices such as a keyboard and a mouse since they are required to wear heavy protective gloves for industrial processes. Various types of gesture interfaces have been also presented in three-dimensional games based on virtual reality and these interfaces have enhanced an interest level and creativity within these environments for the users. Humans can distinguish hand poses very quickly through their complex optical systems, while it is very difficult for a computer system to rapidly and accurately understand hand poses. Therefore, many researchers have tried to simulate the human optical system, which can extract objects of interest from complex scenes and understand the context among objects. One of the major factors that disturb automatic gesture recognition is illumination change. The sudden illumination changes lead to the misunderstanding of background and foreground regions.

We propose a robust hand recognition technique that can stably extract hand contours even under sudden illumination changes. Figure 1 shows the flowchart for our proposed method. The proposed method acquires the background images for a restricted duration and calculates the mean and standard deviation for the hue and hue-gradient of each pixel within the captured background images. That is, a background model for each pixel is built. The hue and hue-gradient of the input images captured in real-time are calculated and compared to those of the background images. The foreground objects are extracted based on the difference magnitude between those of the input image and the background image. To accurately extract the tight object region of interest, we calculate the eigen value and eigen

vector for the initially extracted object region and extract the object-oriented bounding box(OBB) on the optimized hand region based on the two eigen vectors. Then, the OBB region is divided into 16 sub-regions and the hand region profile is produced based on the histogram created from the number of edges for each sub-region. The profiles of nine hand poses are trained and each hand pose is recognized using a multi-class SVM algorithm.
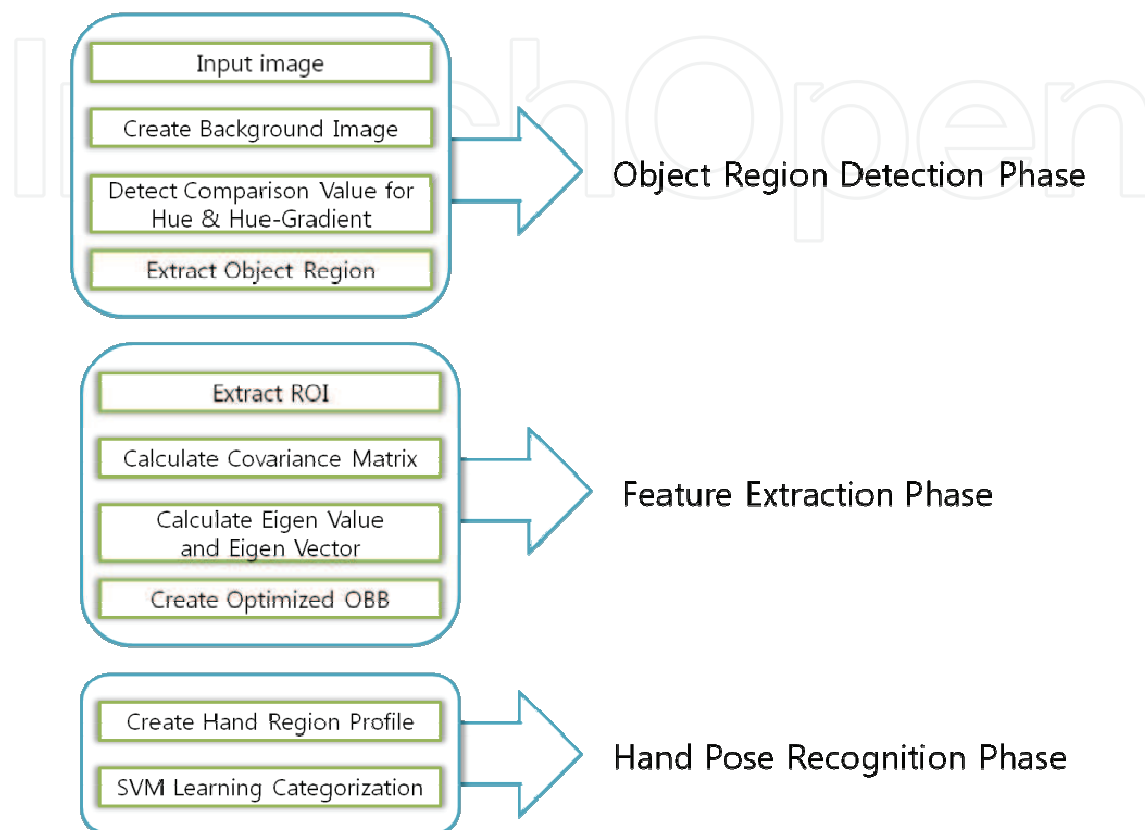

Figure 1. System flowchart

2. Previous Work

Current research of hand pose recognition can be classified into hardware sensor based methods that mostly utilize datagloves and image processing based methods that utilize two-dimensional pattern information or three-dimensional models. Since methods that use datagloves obtain three-dimensional data through sensors directly attached to the gloves in real-time, these methods make hand gesture analysis easy. However, more or less expensive equipments are required and there are problems in implementing natural interfaces since the cables are necessary to connect the equipments to the computing system. As image processing based methods, restricted approaches that uses red or green gloves to skip the complex pre-processing of the hand regions extraction have been presented (KAIST,2004). Other approaches have attempted to recognize hand movements based only on images captured from video cameras under normal lighting conditions without any specialized equipments (Park et al., 2002; Jang et al., 2004; Han, 2003; Jang et al., 2006; Tanibata et al., 2002; Licsar et al., 2005). To support general and natural interfaces, an interest in the hand pose recognition without the use of any specialized equipments or gloves have increased.

A variety of hand features have been applied for hand pose recognition. In the research proposed in (Park et al., 2002), the direction and size of the hand region was considered to determine the area of interest window and the improved CAMSHIFT algorithm was used to track the coordinates of the hand tip. The focal point of their research was in recognizing the pointing direction by extracting the coordinates of the hand tip from a single hand pose instead of various hand poses. In (Jang et al., 2004)'s research, they defined three main features for the hand pose. First, the normalized hand region on the direction and size was calculated and then from the center point of the normalized hand region, straight lines were beamed on the rotation at the regular angle. The distance on the straight line beamed from the center of the region to the outer boundary of the hand was used as the feature. The ratio of the length of the minor axis to that of major axis was also used as the hand feature. In (Han, 2003)'s research on hand pose recognition, they first estimated the entropy on the frame difference between adjacent frame images and through the distribution on the skin color, then they extracted only the hand region from the frame image. Nest, the contour of the hand was detected using a chain-code algorithm and an improved centroidal profile method was applied to recognize the hand pose on a speed of 15 frames per second. In (Jang et al., 2006)'s research, they proposed a method that divides the hand pose features into structural angles and hand outlines. They defined the relation between these hand features by using enhanced learning. Through their research, they proved the appropriateness of their proposed method by applying their method to a hand pose recognition system that uses not one but three cameras. As shown above, although many methods have been proposed for hand pose recognition, most of the research is carried out in limited situational environments with no change in lighting or the background. Thus, environment-dependent research results have been produced(Park et al., 2002; Jang et al., 2004; Han, 2003; Jang et al., 2006; Tanibata et al., 2002; Licsar et al., 2005). In contrast to the research, the application areas for hand pose recognition techniques call for environments that can handle illumination changes. Thus there is an increased demand for research on robust hand pose recognition under illumination changes. In addition, a variety of hand features that characterize the hand pose have been proposed in order to improve the success rate of the recognition, however, these features are quite complex making it difficult for real-time processing. This calls for better research in defining hand features that can efficiently and accurately represent hand poses.

3. Hand Region Extraction

3.1 Hue and Hue-Gradient Distribution of the Hand Region

The RGB color model is an additive color model that adds the three additive primaries - R(Red), G(Green), B(Blue) to create a desired color. RGB color model is sensitive to light and shadow making it difficult to extract object outlines during image processing(Kang, 2003). On the contrary, HSI color model is a user-oriented color model that is based on how humans perceive light. This model is composed of H(Hue), S(Saturation), and I(Intensity). Figure 2. Shows the HSI color model is depicted as a basic triangle and color space. Hue is expressed as an angle ranging from 0° to 360°and saturation is the radius ranging from 0 to 1. Intensity is the z axis with 0 being black and 1 being white(Kang, 2003).
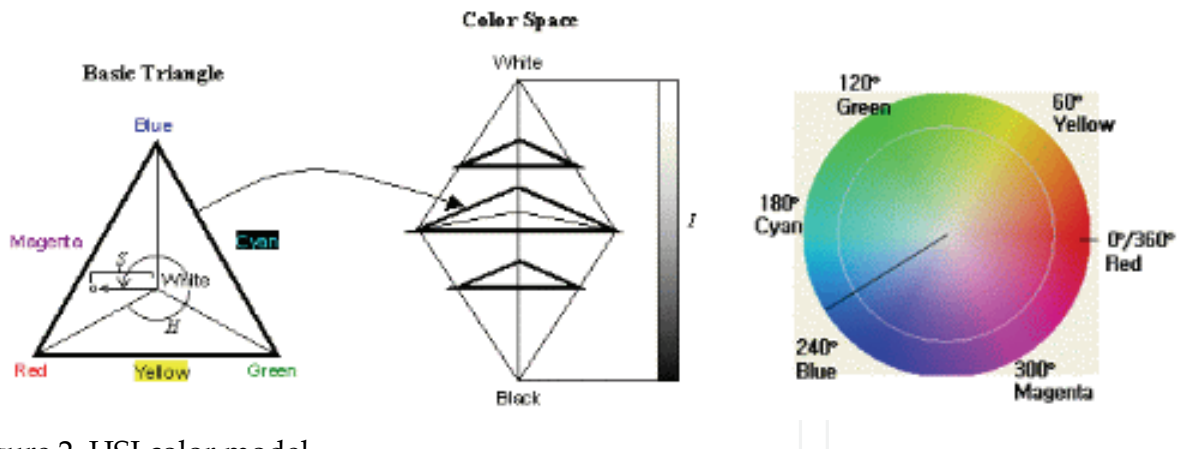
Figure 2. HSI color model

The hue component in the HSI color model represents the object's inherent color value independently from the brightness or saturation of an object. This model can be converted from the RGB model through Equation (1)(Kang, 2003). Figure 3 shows the hue and saturation difference between the hand region and the background region.

$$H = \begin{cases} \theta & if & B \leq G \\ 360 - \theta & if & B \succ G \end{cases}$$

$$\theta = \cos^{-1}\left(\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right)$$

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)]$$

$$I = \frac{1}{3}(R+G+B)$$

(1)
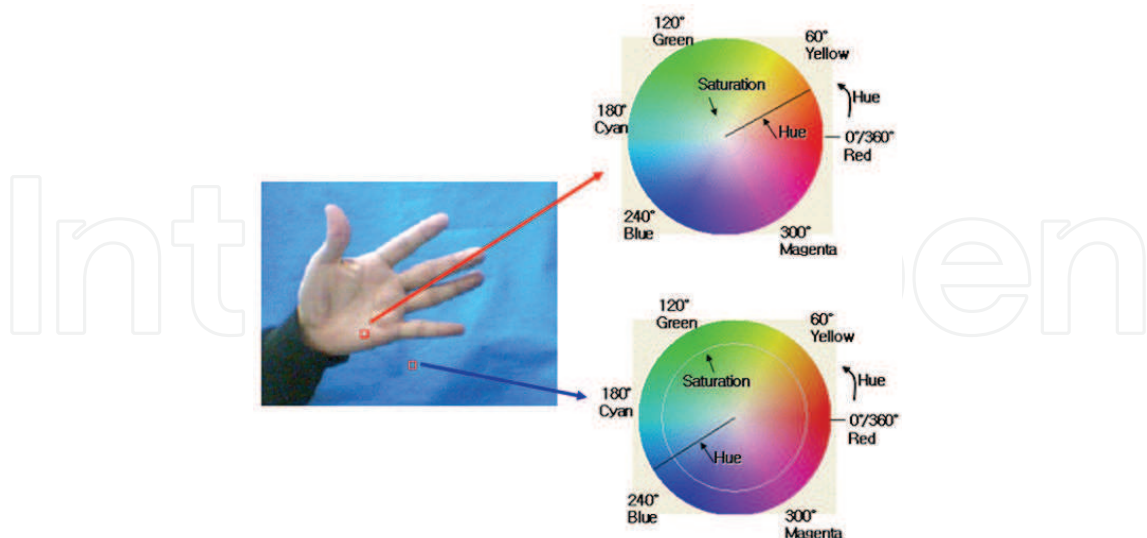
Figure 3. Hue and hue-gradient in hand region and background region

In HSI color model, components related to the color are represented by hue and saturation, and brightness is represented by intensity, which makes it easier to separate a region of

interest regardless of illumination change. HSI color models are widely used in the field of interpretation of images from cameras because when there is a change in the color, all three RGB components in the RGB model are changed, whereas in the HSI model, only the hue are mainly changed. Table 1 shows the histograms for the hue image and saturation image under changes in the illumination. When there is a sudden change in the lighting condition, we can see that the histogram of the saturation is completely changed. On the other hand, although there is some change in the histogram, the distribution characteristic of the histogram of the hue is relatively preserved.
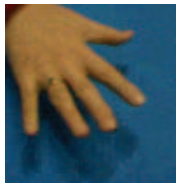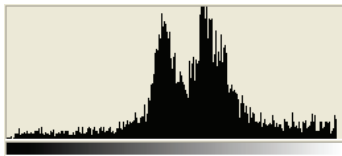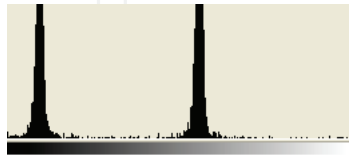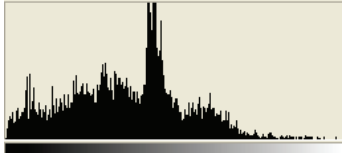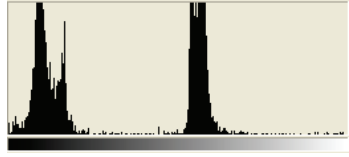
Original Image	Saturation Histogram	Hue Histogram

Table 1. Change of saturation and hue in response to illumination change

3.2 Background Subtraction Based On Hue and Hue-Gradient

Background subtraction is an object detection method that first obtains the average background image from frame images inputted for a set period of time, and then carries out a pixel comparison between the average background image and the incoming input image (Haritaoglu, 1998). This method is summarized as follows.

$$\text{if } (|I_n(x) - B_n(x)| > T_n(x))$$

$$x \text{ is a foreground pixel}$$

$$\text{else} \tag{2}$$

$$x \text{ is not a foreground pixel}$$

$B_n(x)$ is the background image pixel. In other words, if the difference between the background image pixel and the current image pixel $I_n(x)$ is bigger than the threshold $T_n(x)$, then the background pixel is regarded as the foreground pixel. In contrast to the frame subtraction methods that can only detect areas of movement in a short period of time during a scene change, background subtraction methods can detect the overall region of the moving objects. Thus object detection can be efficiently carried out if the background image well represents the current environmental condition without much change. However, if there are changes to the environment such as illumination changes and the background image does not accurately represent the actual environment anymore, the objects cannot be successfully detected using the background subtraction method. Therefore, the accuracy of the background subtraction method is highly dependent on how accurate the background model represents the current background. For accurate object detection, the background image needs to continuously learn the changing environment over time.

The hue component represents the color of the image itself and minimizes the illumination effect. The hue-gradient image maintains the background image's features and on the other hand eliminates the illumination changes and shadow effects. This section introduces a background subtraction method based on hue and hue-gradient. First, to acquire the background model image in the HSI color space, a background model that has been trained for a certain period of time is built with hue and hue-gradient of the background region. Through this model, the object is extracted by first calculating the hue and hue-gradient values when the object is inputted and then separating the foreground region from the input image by comparing the threshold defined based on the background model (Choi, 2007).
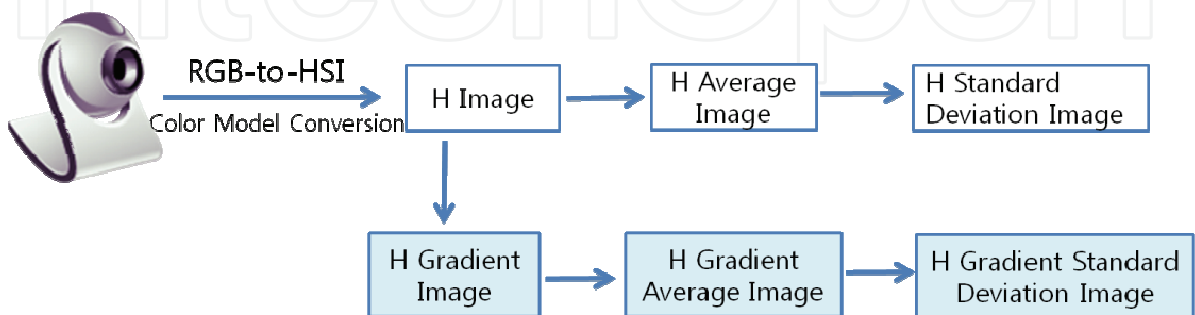
Figure 4. The procedure of background image creation

During the background learning stage, learning is carried out on both the hue component and hue-gradient value for each pixel. To measure the color change caused by general illumination changes, the background image is first acquired for a time period of T_i and then the mean and variance values for each pixel's hue is calculated.

Likewise, a sequence of background images inputted for a time period of T_i is used to calculate the hue-gradient background image. The individual pixels are used with the adjacent pixels to calculate the gradient size of the hue component value. Another averaged background image is then built with the hue-gradient component value.

The equation to calculate the hue gradient size (∇H) of each pixel in the background image is shown in Equation (3).

$$\nabla H = \sqrt{\left(H_{(x+1,y)} - H_{(x,y)}\right)^2 + \left(H_{(x,y+1)} - H_{(x,y)}\right)^2} \tag{3}$$

The equation to gradually calculate the mean and variance of the hue component for each pixel for time t is shown in Equation (4).

Mean Update :

$$\mu\big(H_i(0)\big) = H_i(0) \qquad\qquad : t = 0$$

$$\mu\big(H_i(t)\big) = (1 - \alpha)\mu\big(H_i(t-1)\big) - \alpha H_i(t) \qquad\qquad : t \geq 1$$

Variance Update :   (4)

$$\sigma^2\big(H_i(0)\big) = \big(H_i(1) - \mu(H_i(0))\big)^2 \qquad\qquad : t = 1$$

$$\sigma^2\big(H_i(t)\big) = (1 - \alpha)\sigma^2\big(H_i(t-1)\big) + \alpha(H_i(t) - \mu(H_i(t)))^2 \qquad : t \geq 2$$

The equation to gradually calculate the hue-gradient component's mean and variance for each pixel for time t is shown in Equation (5). $\mu(\nabla H_i(0))$ is the hue-gradient mean's initial value and $\mu(\nabla H_i(t))$ is the mean at time t. $\sigma^2(\nabla H_i(0))$ is the hue-gradient variance's initial value and $\sigma^2(\nabla H_i(t))$ is the variance of hue-gradient at time t.

Mean Update :

$$\mu\big(\nabla H_i(0)\big) = \nabla H_i(0) \qquad\qquad : t = 0$$

$$\mu\big(\nabla H_i(t)\big) = (1 - \alpha)\mu\big(\nabla H_i(t - 1)\big) + \alpha\nabla H_i(t) \qquad : t \geq 1$$

Variance Update :
$$\tag{5}$$

$$\sigma^2\big(\nabla H_i(0)\big) = \big(\nabla H_i(1) - \mu(H_i(0))\big)^2 \qquad\qquad : t = 1$$

$$\sigma^2\big(\nabla H_i(t)\big) = (1 - \alpha)\sigma^2\big(\nabla H_i(t - 1)\big) + \alpha\big(\nabla H_i(t) - \mu(H_i(t))\big)^2 \qquad : t \geq 2$$

The inputted images which include the object are converted from the RGB color space to the HSI color space. If difference between the converted hue value of each pixel and that of background image is greater than the variance of background hue image, it is regarded as the object candidate region. Likewise, if difference between the hue-gradient of each pixel of the input image and that of the background image is greater than the variance of background hue-gradient image, it is regarded as the object candidate region. The region that satisfies with both conditions is extracted as the object region. The procedure of the proposed object extraction is shown in Figure 5.
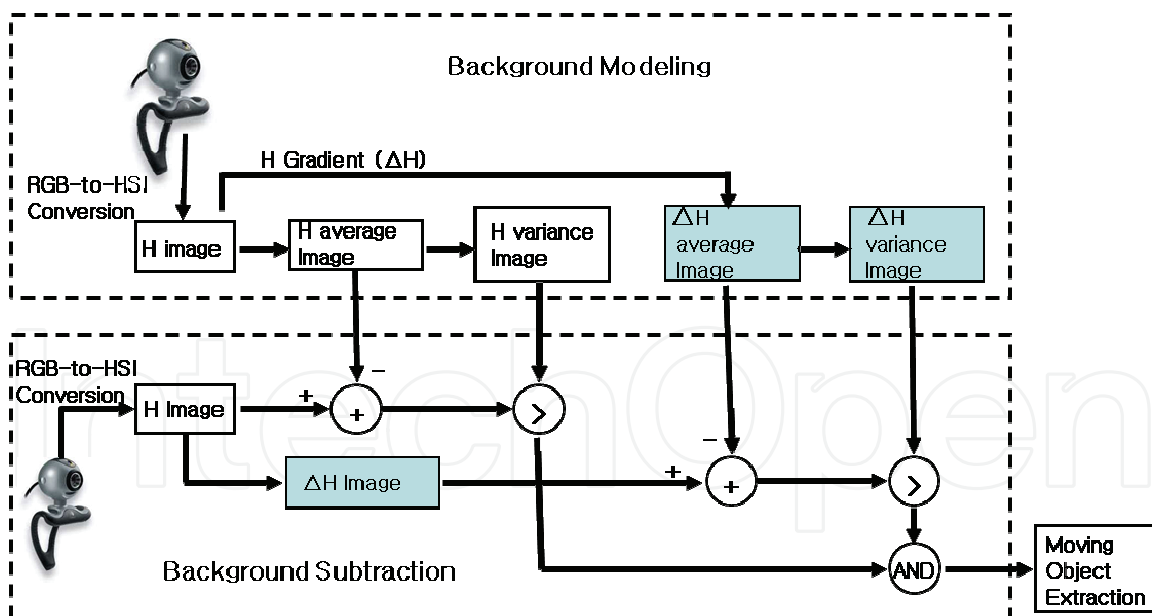
Figure 5. The Procedure of the proposed object extraction

The comparison to extract the object region is shown in Equation (6).

$$R_i(x) = \begin{cases} 1, & \text{if } \big|H_i(x) - H_{b_i}(x)\big| > \omega_1 \sigma\big(H_{b_i}(x)\big) \text{ and} \\ & \big|\nabla H_i(x) - \nabla H_{h_i}(x)\big| > \omega_2 \sigma\big(\nabla H_{h_i}(x)\big), \\ 0, & \text{otherwise} \end{cases} \tag{`6}$$

$H_i(x)$ and $\nabla H_i(x)$ represent the hue and hue-gradient value of pixel i of the current input image, respectively. $H_{b_i}(x)$, $\nabla H_{b_i}(x)$, $\sigma\left(H_{b_i}(x)\right)$, and $\sigma\left(\nabla H_{b_i}(x)\right)$ represent the means of the hue and hue-gradient, variances of hue and hue-gradient of the background model, respectively. ω_1 and ω_2 each represent the weight value for the threshold region.

4. Hand Pose Recognition

4.1 Hand Pose Feature Extraction

This section presents a hand pose recognition method that recognizes the 1-9 hand sign of the sign language regardless of its direction and size. Figure 6 shows the 1-9 hand sign. 18 scalar feature values, that is, two normalized eigen values of the OBB for the detected hand region, the number of hand edge points in 16 subregions that are defined in the detected OBB, are used as features of a hand pose.
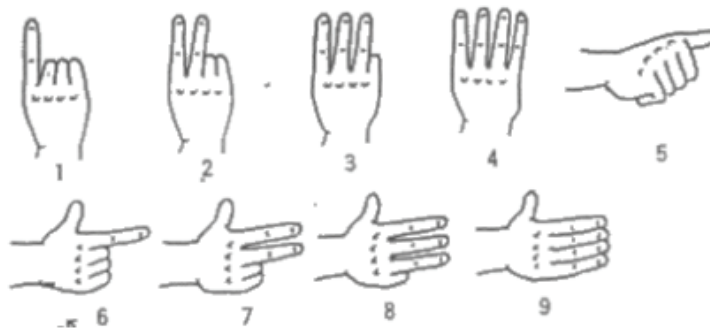
Figure 6. Numbers 1 to 9 using hand signs

4.1.1 Extraction of the eigen value and eigen vector

The detected object has different directions and sizes making image recognition difficult. To increase the success rate of the recognition of hand poses regardless of hand's direction and size, the object-oriented bounding box(OBB) is calculated by extracting the two eigen vectors for the detected hand object. The eigen vetors are preserved even though the object is rotated or scaled. Eigen value and eigen vector always form a pair (Kreyszig, 1999a). The eigen value and eigen vector are derived from the covariance matrix that expresses the specific object's fluctuating value.

Covariance matrix is the statistical criteria, which represents the changing aspect of each fluctuating value when two or more fluctuating data is given. The covariance matrix is calculated using the following equation when the samples' random data is bivariate x_i and y_i(Kreyszig, 1999b).

$$\text{If} \quad \bar{x} = \frac{1}{n}\sum_{i=0}^{n} x_i \;,\; \bar{y} = \frac{1}{n}\sum_{i=0}^{n} y_i$$

$$\text{cov}_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{7}$$

The covariance matrix C is

$$c = \begin{pmatrix} \text{cov}_{xx} & \text{cov}_{xy} \\ \text{cov}_{yx} & \text{cov}_{yy} \end{pmatrix} \tag{8}$$

The covariance cov_{xx} is equal to the variance of x, var_x and the cov_{yy} value is equal to the variance of y, var_y. The covariance matrix is always symmetric and has the same number of eigen vectors as the dimension of the covariance matrix. Eigen vectors are always perpendicular to each other as shown in Figure 7.
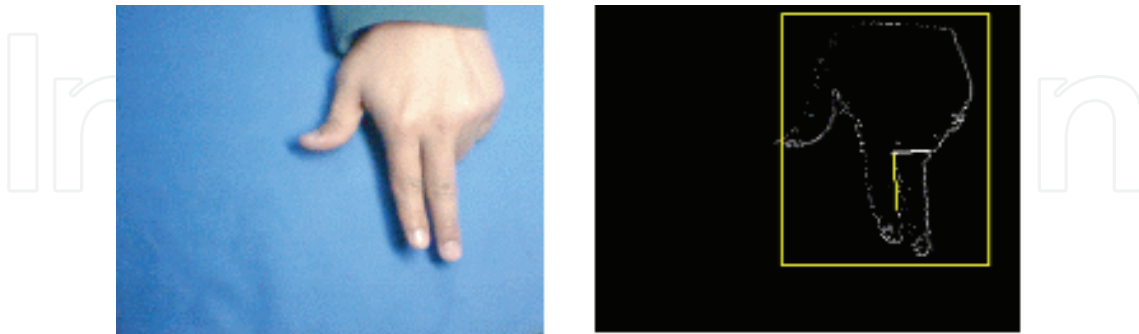

Figure 7. Example of eigen vectors for one hand pose

4.1.2 Definition of feature values

Figure 8 shows the flowchart for the OBB extraction process for a hand pose. Figure 8 (a) shows the target OBB region for the hand pose and Figure 8 (b) shows mean point A and the major/minor eigen vectors. Figure 8 (c) represents the transform to move the object coordinates to x-y coordinates. That is, the mean point is moved to the origin, and major vector and minor vector are aligned to x axis and y axis, respectively. Figure 8 (d) shows that the origin is translated to the center of the window.

To align the major and minor eigen vectors of the hand region to x axis and y axis, the matrix W is defined by eigen vectors, u_1 and u_2, as Equation (9). The contour points can be transformed to the x-y coordinates by using Equation (10).

$$W = [u_1 \ldots u_n]$$ (9)

$$y = W^T x$$ (10)

After the main axes of the object are transformed, a maximally possible OBB is defined based on the variances of x and y, S_x and S_y. The reason for holding a maximally possible OBB region with the variance S_x and S_y in advance is to prevent the size of OBB from being extremely misestimated due to image noise that exist outside the hand region. The OBB is adjusted by finding the maximum contour point from the starting point on both the x and y axis' positive and negative directions within the maximally possible OBB. By defining the maximally possible OBB in advance with the variance and optimizing the OBB based on the contour point within the OBB, the error rate for false ROI extraction due to background noise can be reduced. Figure 8 (e) shows the result of the optimized OBB detection.

18 feature component values are used to recognize the hand signs representing 1 to 9 from the hand images. First, the eigen values from the two directions of the hand region is normalized using the variances of x and y within the OBB. The OBB is then subdivided into 16 regions as shown in Figure 9 and the number of overall contour pixels contained in each region is counted. Two normalized eigen values and the number of contour points for each sub-region are used as the feature information.
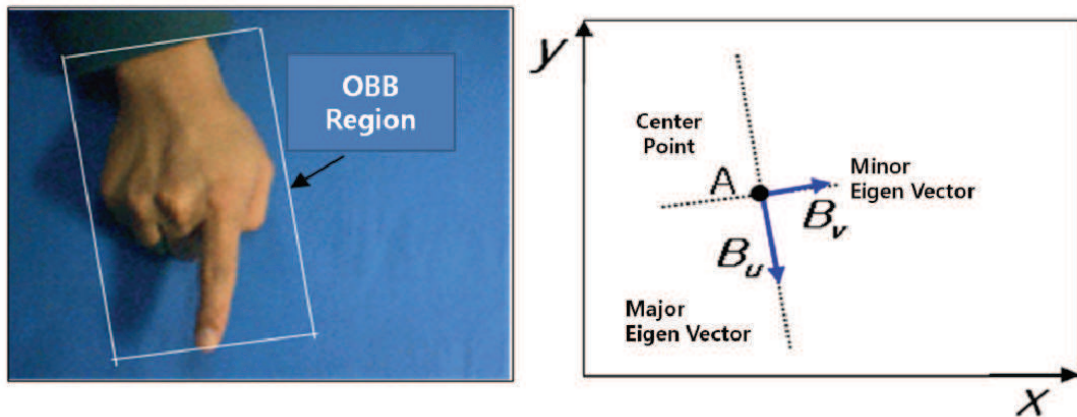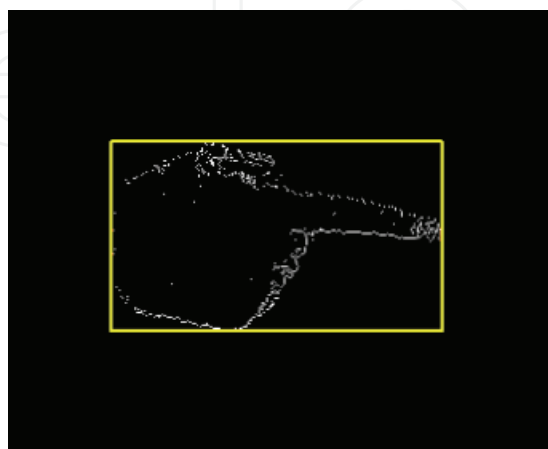
(a) Input image and OBB region of interest

(b) Center point and eigen vectors
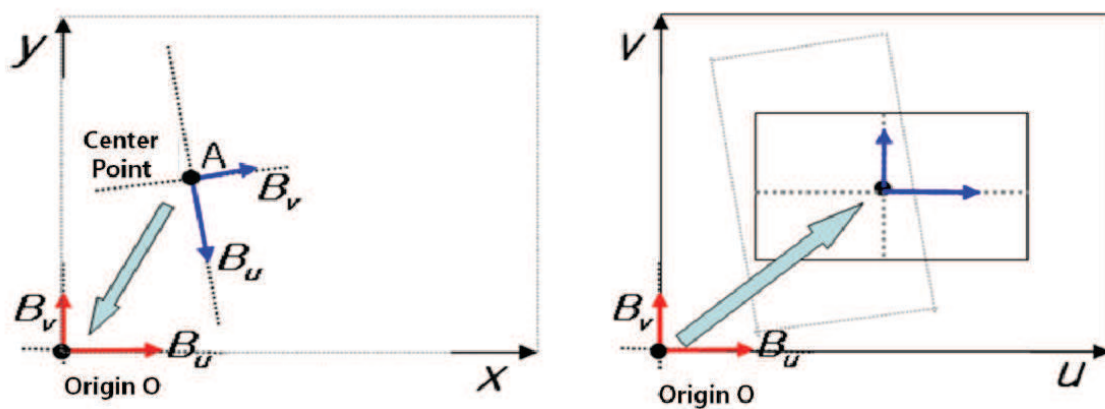
(c) Alignment of eigen vectors

(d) Translation of origin

(e) optimized OBB
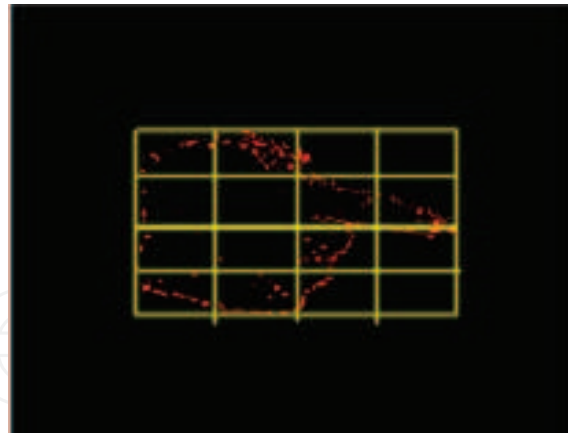
Figure 8. Procedure for extracting the optimized OBB

Figure 9. Sixteen sub-regions of an OBB

4.2 Recognition using SVM

SVM (Support Vector Machine) is a method that was developed by Vladimir Vapnik and his research team at AT&T Bell Research Lab. It is an object identification method that is widely used from the field of data mining to pattern recognition application areas such as face recognition(Han, 2005; Cristianini, 2000; Weida, 2002). A linear SVM is a Statistical Learning Theory(SLT) which classifies the arbitrary data based on the decision function being defined by PDF(Probability Density Function). PDF is obtained by the learning process of the training data and the categorical information.
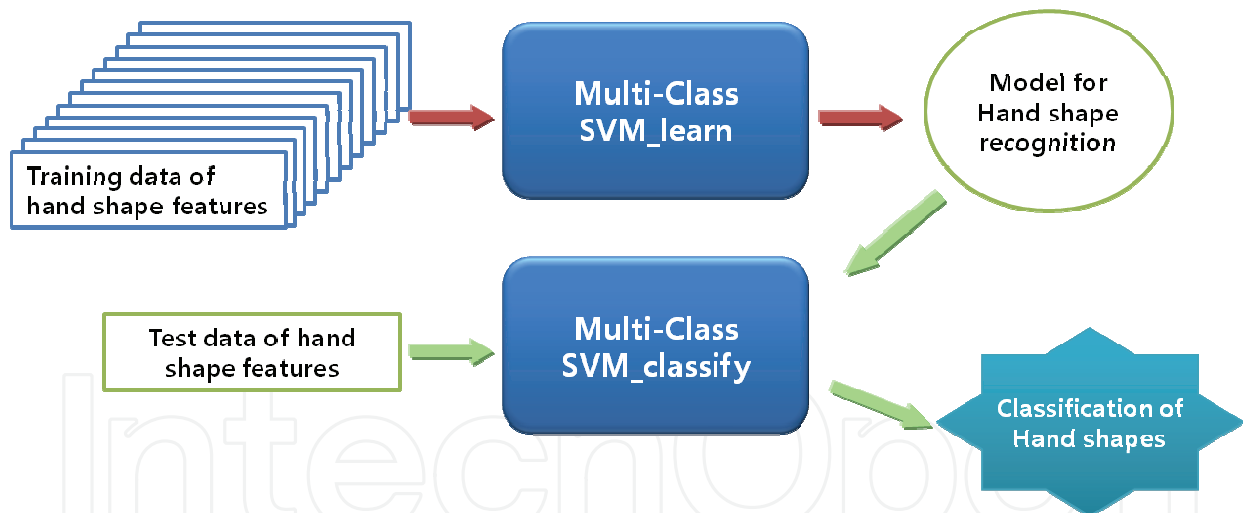
Figure 10. Procedure of hand shape recognition using SVM

SVM is an alternative learning method to polynomial, radial basis function and multi-layer perceptron. SVM can abstract patterns into higher level feature space and make globally optimal classification to be possible. Most of traditional pattern recognition methods such as neural network and statistical methods are based on empirical risk minimization(ERM) for optimal data execution, while SVM is based on the structural risk minimization(SRM), which minimizes the probability to misclassify unknown data set(Han, 2005). Originally the SVM was conceived for only dual class, but it has been expanded to multi-class classification responding to need. Expanding the SVM from a dual class to multi-class involves combining the dual class SVMs. In this section, nine hand poses are categorized using the multi-class SVM(Crammer, 2001). The

multi-class SVM technique is used on the feature values extracted from the hand poses in section 4.1 to produce the learning model. The hand pose is extracted from the real-time input images and the feature values are extracted to classify the hand pose based on the hand learning model. Figure 10 shows the process for hand pose learning and recognition using the SVM.

The training data created for the SVM learning algorithm is sorted as the class name, feature classification number, and feature value. Learning is executed in off-line by the multi-class SVM learning function. The class name is the hand sign number from the hand signs 1 to 9's classification and each class defines 18 feature values which are sorted as the feature classification number and the extracted feature value. The feature values contain the two normalize eigen values and the overall number of pixels for OBB's 16 sub-regions.

5. Experimental Results

5.1 Hand Region Detection Under Illumination Change

The testing environment set up for this experiment was implemented with Visual C++ software on a Windows XP with a Pentium-IV 3.0 GHz CPU and 1 GB of memory. Logitech Quickcam Chat camera was used to acquire input test images for the hand tracking. The images are 320 x 240 in size captured as a 24bit RGB color model.

The hue based background image was produced by converting the RGB color space into the HSI color space, and then calculating the hue and the hue-gradient's mean and variance. During the next step, the hue and hue-gradient values are extracted from the image pixels captured in real-time. The hue and hue-gradient values of the input image are compared with those of the background images. As a result, the binary image including the contour of the hand is produced. In other words, the contour of the hand is marked as white and the background is marked as black.

	(a) input image	(b)RGB color model	(c) Normalized RGB Color Model	(d) Hue	(e) Hue and H-Gradient
Bright Illumination					
Dark Illumination					

Table 2. Comparison of background subtraction methods based on different color models under sudden illumination changes

For the experiment, we compared the results of the existing methods with that of the proposed method under the exact same illumination from when the initial background was

built. Then we also compared them under drastically different illumination conditions from the initial illumination.

To create the illumination change, one condition used a 80 lm/W normal white fluorescent lighting together with a 18W 63.1 lm/W white desktop lamp directly on the hand region. The other condition eliminated the 18W 63.1 lm/W white desktop lamp and carried out the experiment to extract the hand region without rebuilding the background model for the varying illumination.

Object	Covariance Matrix		Eigen Value		Eigen Vector	
none	0	0	0	0	1	0
	0	0			0	1
1	589.96	216.33	494.33	1079.3	0.91	-0.4
	216.33	983.66			0.4	0.91
2	619.37	293.45	531.11	1595.08	0.96	-0.29
	293.45	1506.82			0.29	0.96
3	566.75	182.57	527.82	1423.06	0.98	-0.21
	182.57	1384.14			0.21	0.98
4	751.79	99.61	738.21	1482.39	0.99	-0.14
	99.61	1468.81			0.14	0.99
5	931.3	5.57	585.24	931.39	0.02	-1
	5.57	585.33			1	0.02
6	913.41	117.5	813.78	1051.99	0.76	-0.65
	117.5	952.35			0.65	0.76
7	831.65	154.6	783.37	1326.71	0.95	-0.3
	154.6	1278.43			0.3	0.95
8	832.16	28.46	830.77	1414.51	1	-0.05
	28.46	1413.11			0.05	1
9	1037.26	-73.61	1022.4	1402.02	0.98	0.2
	-73.61	1387.16			-0.2	0.98

Table 3. Covariance Matrix, Eigen Value and Eigen Vector's Mean on the Hand Signs 1 to 9

As shown in Table 2, when the dark and bright illumination conditions are compared, we can see that applying the proposed method better preserves the hand contour. Table 2. (b) is the binary image produced from the difference between the background image based on the RGB color model and the input image. It is the most sensitive to light change. Table 2. (c) is the result of applying normalized RGB color model as the background model. Here the contour of the hand is roughly preserved but the hand's internal noise as well as shadow noise is more prominent. Table 2. (d) is the binary image based only the HSI color model's hue value. There is an improvement from the RGB color model but it reacts to the shadows

due to the illumination changes. Table 2. (e) shows that even under sudden illumination changes, the region information can be stably extracted using the proposed method.
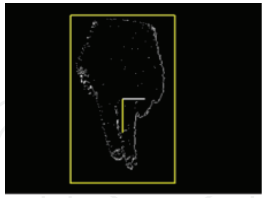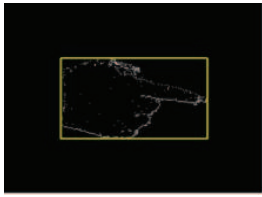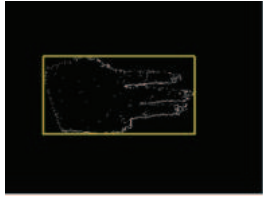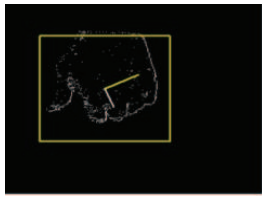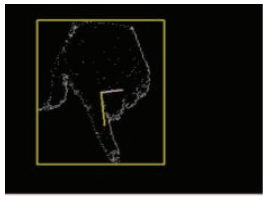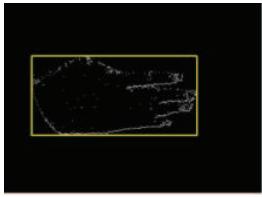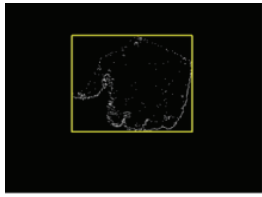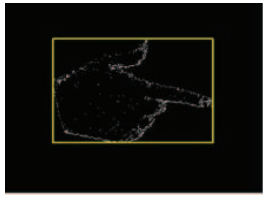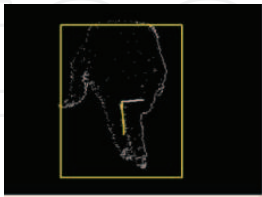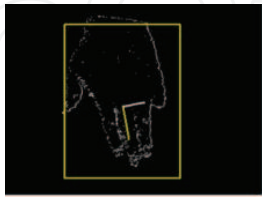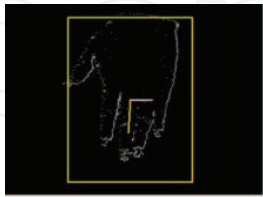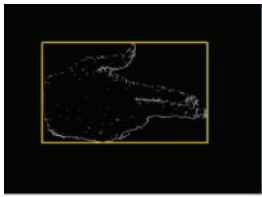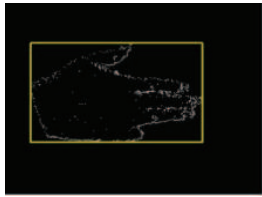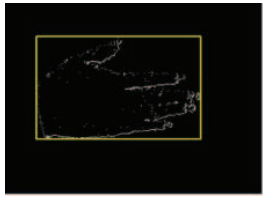
	(a) Hand Sign 1	(b) Hand Sign 2	(c) Hand Sign 3
Eigen Vectors			
Optimized OBB			
	(d) Hand Sign 4	(e) Hand Sign 5	(f) Hand Sign 6
Eigen Vectors			
Optimized OBB			
	(g) Hand Sign 7	(h) Hand Sign 8	(i) Hand Sign 9
Eigen Vectors			
Optimized OBB			

Table 4. Optimized OBB detection results on the hand signs (1-9)

Table 3. shows all the covariance matrix values as well as the eigen values and eigen vectors on the hand signs 1 to 9. Each hand sign has an eigen value proportioned to the input image object's ROI size. Figure 11 compares 90 samples for the hand signs representing from 1 to 9. When image interpretation is carried out using the correlation analysis on the major and minor eigen values, we can see that an initial classification is possible on the 1 to 9 hand signs. Table 4 shows the optimized OBB detection result for the object in the input image. The first row shows eigen vectors to be extracted from the result image of background subtraction and the second row represents the OBB optimized by transforming object coordinates using eigen vector matrix and by detecting object's outermost points.
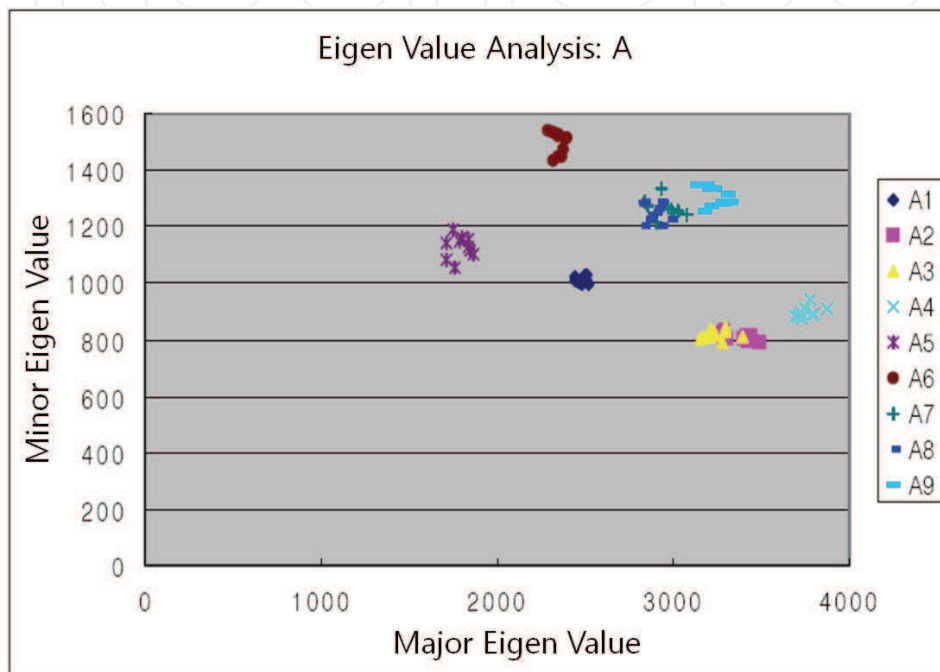

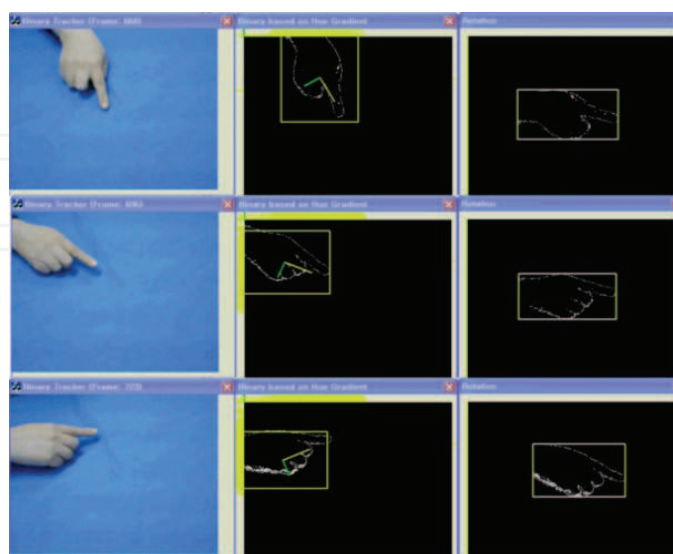Figure 11. Two-dimensional distribution on the eigen value for hand signs 1 to 9

Figure 12. Optimized OBB on hand sign 1 from different directions. (Left) Input image (Middle) ROI and eigen vectors prior to optimization (Right) Optimized OBB

Figure 12. shows the OBB that has been optimized on the same hand sign 1 expressed from different angles. As we can see, even with changes in the hand direction, the normalized OBB's direction is steadily maintained and enabling stable hand recognition independent of hand direction.

5.2 Hand Recognition

The data used for this experiment consists of 1620 images – 180 images per hand sign (1 to 9). All 6 subjects individually changed hand poses on each of the hand signs resulting in 30 captured images per hand sign. 18 different hand features were learned through the SVM learning algorithm and hand recognition was performed. The result of the hand recognition is summarized in Table 5. This includes the result of failed recognition by one subject on 1,2, 4, and 7. The mean success rate of recognition on the 9 hand signs showed 92.6%.

For movies incoming at 30 frames per second, the proposed hand recognition method was able to extract and recognize hand poses at 20 frames per second. The hand recognition processing time on one frame showed 1.260 msec enabling real-time processing.

class	Number of Tests	Normal Recognition	Faulty Recognition	Recognition Rate
1	180	150	30	83.3
2	180	150	30	83.3
3	180	180	0	100
4	180	150	30	83.3
5	180	180	0	100
6	180	180	0	100
7	180	150	30	83.3
8	180	180	0	100
9	180	180	0	100
Sum	1620	1500	120	92.6

Table 5. Optimized OBB Detection Result on hand signs 1 to 9

6. Conclusions

This chapter introduced a hand pose recognition method to robustly extract objects under sudden illumination changes. The introduced method constructs the background model

based on the hue and the hue-gradient, and then robustly extracts the object contours from images obtained from a fixed camera by the background subtraction. It was shown that this method stably detects the object even under various lighting conditions set by the experiment.

This research carried out a comparison experiment on the different color models for image recognition systems. The HSI color model minimizes the illumination and shadow effects and by using the hue and hue-gradient values from this color model for the background subtraction method, it showed that the proposed method sharply decreases the noise and shadow effects caused by sudden illumination changes. The optimized OBB that was produced from the extracted region is then divided into 16 sub-regions.The number of edges of the hand in that sub-region and the normalized eigen values are used for defining the hand pose feature which are then trained by the SVM learning algorithm.

This method was designed for use in gaming environments where illumination conditions can change rapidly in the limited simple environment. When the proposed method was tested in a complex environment, it was shown that if the hand region has similar color to the background region, the hand's contour was lost. In the future, this method will be expanded to include stable hand pose recognition in complex background environments.

7. References

Choi, Y.;Lee, J.;Cho W. (2007) Robust Extraction of Moving Objects based on Hue and Hue Gradient, *LNCS*, Vol. 4555, pp.784~791

Crammer, K.; Singer, Y. (2001) On the Algorithmic Implementation of Multi-class SVMs, *JMLR*

Cristianini, N. (2000) An Introduction to Support Vector Machines, *Cambridge University Press*

Han, H. (2005) Introduction to Pattern Recognition, *Hanbit Media*, pp.274~282

Han, Y. (2003) Gesture Recognition System using Motion Information, *The KIPS Transactions: PartB* , Vol. 10, No. 4, pp.473-478

Haritaoglu, I.; Davis, L.; Harwood, D. (1998) W4(Who? When? Where? What?) a real time system for detecting and tracking people. *In FGR98*

Jang, H.; Bien, Z. (2006) A Study on Vision-based Robust Hand-Posture Recognition Using Reinforcement Learning, *Journal of IEEK* : CI. Vol. 43, No.3, pp.39-49

Jang, H.; Kim, D.; Kim, J.; Jung, J.; Park, K.; Z. Zenn Bien. (2004) Decomposition approach for hand-pose recognition, *International Journals of HWRS*, Vol. 5, No.1, pp.21-25

Jung, K. (2007) "Introduction to Ubiquitous Computing: Connection to new media", *jinhan M&B*, pp. 48~63

Licsar, A.; Sziranyi, T. (2005) User-adaptive hand gesture recognition system with interactive training, *Image and Vision Computing*, Vol.23, No.12 , pp.1102-1114

Kang, D.; Ha, J. (2003) Digital Image Processing using Visual C++, SciTech Media, pp.314~322

KAIST (2004) Hand signal recognition method by subgroup based classification, Korean Patent, 10-0457928

Kreyszig, E. (1999a) Advanced Engineering Mathematics, Eight Edition, *John Wiley & Sons*, Inc., pp. 371~375

Kreyszig, E. (1999b) Advanced Engineering Mathematics, Eight Edition, *John Wiley & Sons, Inc.*, pp. 1150~1151

Park, J.; Yi, J. (2002) Efficient Fingertip Tracking and Mouse Pointer Control for Implementation of a Human Mouse, *Journal of KISS* B Vol.29, No.11, pp.851-859

Park, S. (2004) A Hierarchical Graphical Model for Recognizing Human Actions and Interactions in Video*, Disseratation of Ph.D, The University of Texas at Austin*

Tanibata, N.; Shimada, N. (2002) Extraction of Hand Features for Recognition of Sign Language Words, *The 15th International Conference on Vision Interface*, pp.391-398

Weida, Z. (2002) Linear programming support vector machines, *Pattern Recognition*, Vol.35, No.12, pp.2927-2936.

Advances in Human Computer Interaction

Edited by Shane Pinder

In these 34 chapters, we survey the broad disciplines that loosely inhabit the study and practice of human-computer interaction. Our authors are passionate advocates of innovative applications, novel approaches, and modern advances in this exciting and developing field. It is our wish that the reader consider not only what our authors have written and the experimentation they have described, but also the examples they have set.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds