

Continuous Gesture Trajectory Recognition System Based on Computer Vision

Xu Wenkai¹, Eung-Joo Lee²^{1,2}Department of Information and Communications Engineering, Tongmyong University, Busan, KoreanCorresponding author: Email: Email Address: ejlee@tu.ac.kr

Received Jun. 23, 2011; Revised Aug. 4, 2011; Accepted Sep. 2, 2011

Published online: 1 January 2012

Abstract: In this paper, we propose an automatic system for recognizing continuous gestures in real-time, including Arabic numbers (0-9) and alphabets (A-Z). We present an improved method for hand area detection and segmentation based on YCbCr and HSI mixed skin color space, the improved CAMSHIFT algorithm used for hand tracking. Orientation dynamic features are obtained from spatiotemporal trajectories and then quantized to generate its code words. An improved HMM-FNN model is proposed for gesture recognition based on the code words, which combines ability of HMM model for temporal data modeling with that of fuzzy neural network for fuzzy rule modeling and fuzzy inference. The algorithm we presented has better performance and achieves average recognition rate 95.76% and 93.64% for Arabic Numbers and Alphabets, respectively.

Keywords: Gesture trajectory, gesture detecting, gesture tracking, trajectory recognition, HMM-FNN.

1 Introduction

Tremendous technology shift has played a dominant role in all disciplines of science and technology. Virtual reality technologies, which can give humans the sensation of being involved in computer world, have been a popular research field for many years. The use of hand gesture is an active area of research in the vision community, mainly for the purpose of sign language recognition and Human-Computer Interaction (HCI).

Gesture and posture recognition are application areas in HCI to communicate with computers. A gesture is spatiotemporal pattern which maybe static, dynamic or both. Static morphs of the hands are called postures and hand movements are called gestures. In gesture recognition, Yoon et al. [1] developed a hand gesture system in which combination of location, angle and velocity is used for the recognition. Liu et al. [2] developed a system to recognize 26 alphabets by using different HMM topologies. Hunter et al. [3] used HMM for recognition in their approach where Zernike moments are used as image features for sequence of

hand gestures.

In the last decade, several methods of potential applications in the advanced gesture interfaces for HCI have been suggested but these differ from one to another in their models. Some of these models are Neural Network [4], Hidden Markov Model (HMM) [5] and Fuzzy Systems [6]. Hidden Markov Models (HMM) is one of the most successful and widely used tools for modeling signals with spatiotemporal variability [7].

In this paper, we present a new method to get skin-color segmentation based on mixing nonlinear YCbCr elliptical cluster skin-color model and HSI skin-color segmentation model; we estimate the principal gesture plane using Least Squares Method and classifying gestures using HMM-FNN; the likelihood of each HMM to observation sequence is considered as membership value of FNN, and gesture is classified through fuzzy inference of FNN. By the propose algorithm we achieve better recognition result for continuous gestures trajectory.

2 Hand Gesture Segmentation Algorithm

To be color images, the information of skin-color is very important characteristics for human face. Research shows that: even though of different races, different ages and different gender, the difference in color chrominance is far less than the difference in the brightness. Skin distribution shows clustering distribution in the skin-color space without luminance influence.

Normally, in order to reduce the impact of brightness, we use nonlinear YCbCr elliptic cluster skin-color segmentation model that the illumination component is concentrated in a single component (Y) while the color is contained in the blue (Cb) and the red chrominance component (Cr). Cb and Cr are defined as the difference between the blue component and the difference between the red component and a reference value, respectively. We ignore Y channel in order to reduce the effect of brightness variation and use only the chrominance channels which fully represent the color information. The research of a large number of color pixels shows that skin-color cluster a very small range of CbCr color space. Normalized chrominance distribution maps, we can find that different skin-color have the same 2D Gaussian model. This method can be accurate to detect skin-color regions. The YCbCr color space decomposes the RGB color into luminance and chrominance information, the conversion formula as below:

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ Cb = 0.564(B - Y) \\ Cr = 0.713(R - Y) \end{cases} \quad (2.1)$$

Nevertheless, because of illumination and complex background are similar to skin-color effect, this method still may make skin-color region as non-skin color, and make non-skin color as skin-color[15].

The other commonly used method of getting the skin area is based on HSI color space. HSI color space contains hue (H), saturation(S) and luminance (I). The HSI color space is very important and attractive color model for image processing applications because it represents colors similarly how the human eye senses colors. Thus the analysis of skin-color can be by hue and saturation space as to reduce the impact of

luminance. This color space can detect skin-color well, so it is used in many skin-color detecting researches. The transition from RGB color space to HSI color space is displayed as the formula as below:

$$H = \cos^{-1} \left\{ \frac{1/2[(R-B) + (R-G)]}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right\} \quad (2.2)$$

Nevertheless, under the influence of the environment, this method still may make non-skin color as skin-color. Based on the skin color segmentation results in YCbCr Color Space and HSI Color Space, we analysis the advantages of their own and find out the shortages to think out a better method to get the more satisfactory results. By many times experiments we find that: By fusing the results receive from two methods, in other words, we perform every pixel with "OR" operation on two binary images which get from two color space.

Compared with the two methods which process the image only in YCbCr or HSI Color Space, we can get the better segmentation result [8], the experimental results are shown as below (Figure 1):

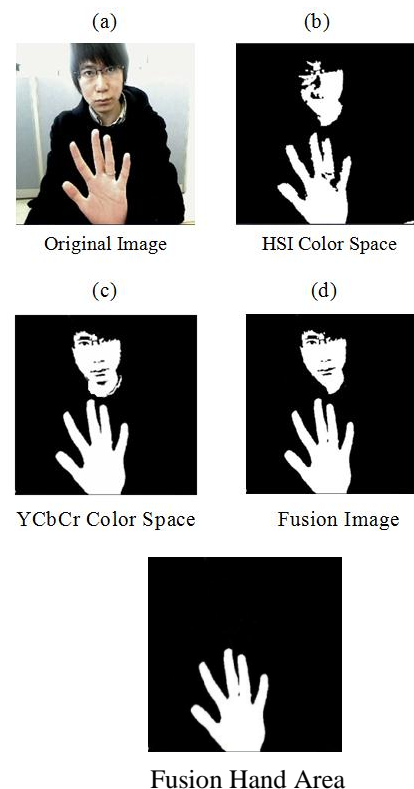


Figure 1: The Hand Detection Procedure

As we just care about the information of hand area, the head area is the interference region, the hand area

should be segmented. Normally, the head is assumed to exhibit no motion or slight motion only. Therefore, this region can be removed by a difference operation where the current image frame is subtracted by the previous image frame. The difference operation can be performed faster on binary images as compared to the color image. The binary image of the current frame is subtracted from the previous binary image to remove edge noise, like face region and other stationary objects that close to the skin color. Thus, the background and the most part of face are removed as it is shown in Figure 1 (e).

3 Hand Tracking based on improved CAMSHIT Algorithm

Computer vision hand tracking is an active and developing field, yet the hand trackers that have been developed are not sufficient for our needs. We want a tracker that will track a given hand in the presence of noise. And it must run fast and efficiently so that objects may be tracked in real time (24 frames per second) while consuming as few system resources as possible for example that running on inexpensive consumer cameras [9].

Compared with the other similar algorithms, the method in this paper improved the accuracy without adding computational complexity and also suitable for all the registration parameters. We plan to select a method named mean shift algorithm, which is a simple iterative procedure that climbs the gradient of a probability distribution to find the nearest dominant mode. The mean shift algorithm operates on probability distributions. To track hand in video frame sequences, the image data has to be represented as a probability distribution. Distributions derived from video image sequences change over time, so the mean shift algorithm has to be modified to adapt dynamically to the probability distribution it is tracking. The new algorithm that meets all these requirements is called CAMSHIFT.

3.1 CAMSHIFT Algorithm

CAMSHIFT algorithm is a dynamic change in the distribution of the density function of the gradient estimate of non-parametric methods [10]. The course of algorithm is as follows:

1. Select a search window W size of s in skin

color probability distribution.

2. Calculate the zeroth moment.

$$Z_{00} = \sum_x \sum_y I(x, y) \quad (3.1)$$

Calculate the first moment of x and y

$$Z_{10} = \sum_x \sum_y xI(x, y) \quad (3.2)$$

$$Z_{01} = \sum_x \sum_y yI(x, y) \quad (3.3)$$

Where $I(x, y)$ is the pixel (probability) value at position (x, y) in the image, and x and y range over the search window.

3. Calculate the mean search window location (the centroid) is:

$$x_c = Z_{10} / Z_{00}, \quad y_c = Z_{01} / Z_{00} \quad (3.4)$$

4. Set the search window size equal to a function of the zeroth moment found in step 2.

5. Repeat steps 2, 3 and 4 until convergence (mean location moves less than a preset threshold).

The initial moment reflects the area of object in the image, and the chart of skin color probability distribution is discrete gray scale image which has the max value is 255. So the relation between the size of search window s and Z_{00} is as follows:

$$s = 2\sqrt{Z_{00} / 256} \quad (3.5)$$

Consider the symmetry: s gets results close to the singular.

By calculating the second-order moment can be obtained the long axis, short axis and the direction angle of object. Second-order moments are as follows:

$$Z_{20} = \sum_x \sum_y x^2 I(x, y) \quad (3.6)$$

$$Z_{02} = \sum_x \sum_y y^2 I(x, y) \quad (3.7)$$

$$Z_{11} = \sum_x \sum_y xy I(x, y) \quad (3.8)$$

The direction angle of object is

$$\theta = \frac{1}{2} \arctan \left[\frac{2(Z_{11} / Z_{00} - x_c y_c)}{(Z_{20} / Z_{00} - x_c^2) - (Z_{02} / Z_{00} - y_c^2)} \right] \quad (3.9)$$

Suppose

$$a = Z_{20} / Z_{00} - x_c^2 \quad (3.10)$$

$$b = Z_{11} / Z_{00} - x_c y_c \quad (3.11)$$

$$c = Z_{02} / Z_{00} - y_c^2 \quad (3.12)$$

The long axis l and short axis w is

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (3.13)$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (3.14)$$

When CAMSHIFT algorithm track a specific color object, the images do not have to calculate each frame all the pixels of the color probability distribution, just calculate pixel color probability distribution in the area that larger than the current search window. This can save a lot of computing [9].

3.2 The Improved CAMSHIFT Algorithm

By the formula-based region of interest are square, but hand is closer to rectangular, when the hand rotation or hand towards the camera angle change, the aspect ration of regular of hand changes.

Assuming that there is no rotating hand, the region of concerned with a width of b , height of h , the search window size is s ,

$$b * h = s^2 \quad (3.15)$$

As the long axis l and short axis w , it has

$$b/h = w/l \quad (3.16)$$

So,

$$b = \sqrt{w/l} * s, \quad h = \sqrt{l/w} * s \quad (3.17)$$

When the target spins, we can get the rotation θ from the formula. Taking into account the rotation of a rectangular rigid body, the region of interest to determine the width and height:

$$b_n = (b \cos \theta + h \sin \theta) \quad (3.18)$$

$$h_n = (b \sin \theta + h \cos \theta) \quad (3.19)$$

As the hand are flexible, and not a true rectangular. An amendment to the formula can be obtained following:

$$b_n = (b \cos \theta + 0.5h \sin \theta) \quad (3.20)$$

$$h_n = (0.5b \sin \theta + h \cos \theta) \quad (3.21)$$

Experiments show that when $\theta < 10$, the calculation of b and h as a region of interest can be satisfied with the results; when $\theta > 10$, the calculation of b_n and h_n as a region of interesting can get better results (Figure 2).



Figure 2: Result Images of Hand Tracking Using the Improved CAMSHIFT Algorithm

4 Dynamic Feature Extraction

4.1 Estimating Principal Gesture Plane

For a nonlinear gesture, we have found that the trajectory of it is almost in a plane, which we call principal gesture plane [11]. To find the principal gesture plane is the most important step. The gesture coordinate system is established by the gesture trajectory and can be represented by principal gesture plane and its normal vector.

From different viewpoints we get different 3D trajectories while the projections of the trajectories on the principal gesture planes are almost the same. So we use Least Squares Method to estimate principal gesture plane. The principal plane is described in the form of $ax + by + cz + d = 0$. The trajectory in 3D space is described as the follow system of homogeneous equations :

$$\begin{cases} ax_1 + by_1 + cz_1 + d = 0 \\ ax_2 + by_2 + cz_2 + d = 0 \\ \dots \\ ax_n + by_n + cz_n + d = 0 \end{cases} \quad (4.1)$$

We could describe it in the way as:

$$Pv = 0 \quad (4.2)$$

Here, a gesture is defined as a series of points $G(p_1, p_2, p_3, \dots, p_n)$ in Cartesian space, where $p_n = (x_n, y_n, z_n)$, and $v = (a, b, c, d)^T$.

We can get suitable v by minimizing

$$E = |Pv|^2 = v^T P^T P v \quad (4.3)$$

We choose singular value decomposition to compute the eigenvectors and eigenvalues [12]. Vector $e_1 = (a_1, b_1, c_1, d_1)$ is the solution minimizing and the equation of principal gesture is $ax_1 + by_1 + cz_1 + d = 0$. Vector $l = (a_1, b_1, c_1)$ is the normal vector of principal gesture plane.

4.2 Trajectory Projection

The next step after getting the normal vector of the principal gesture plane, we should project the gesture trajectory onto the plane. As frontal view gestures are chosen as training sets, we rotate side principal gesture plane parallel to the frontal ones firstly. We can get the projection result through calculate and the inference reached as follow :

$$\begin{aligned} p_i' &= (x_i', y_i') = p_i \times R \times P_r \\ &= (x_i, y_i, z_i) \times R \times P_r \end{aligned} \quad (4.4)$$

Here, R is the rotation matrix between side view gesture coordinate system and frontal view gesture coordinate system. We calculate R from R^{-1} ,

$$R^{-1} = \begin{pmatrix} a_1/\sqrt{a_1^2+c_1^2} & 0 & c_1/\sqrt{a_1^2+c_1^2} \\ 0 & 1 & 0 \\ c_1/\sqrt{a_1^2+c_1^2} & 0 & a_1/\sqrt{a_1^2+c_1^2} \end{pmatrix} \quad (4.5)$$

And P_r is the projection matrix,

$$P_r = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad (4.6)$$

So we can use the projection result p_i' to simplify the gesture coordinate in 3D space, and then the following recognition part is based on 2D trajectory $G'(p_1', p_2', p_3', \dots, p_n')$.

4.3 Feature Extraction

There are three basic features: location, orientation and velocity. The previous research [13, 14] showed that the orientation feature is the best in term of accuracy results. Therefore, we regard the orientation feature as the main feature during our research process.

The $p_i'(x_i', y_i')$ position data of points of the hand trajectory are converted into direction codes representing direction vectors (Figure 3). The moment distance d_i and changing angle θ_i are calculated for each position by the following equations:

$$d_i = \sqrt{(x_i' - x_{i-1}')^2 + (y_i' - y_{i-1}')^2} \quad (4.7)$$

$$\theta_i = \tan^{-1} \left(\frac{y_i' - y_{i-1}'}{x_i' - x_{i-1}'} \right) \quad (4.8)$$

The angles are each converted into one of the eight direction codes that are shown in Figure 4. The angle ranges of the direction codes have different widths. The feature of 2D motion trajectory of hand gesture is represented by a series of discrete movement direction value. For the 2D motion plane, we divide the direction into eight discrete values as shown (Figure 4). Therefore, the trajectory of dynamic gesture can be described by the sequence of discrete direction value $T: t_1, t_2, t_3, \dots, t_i (1 \leq t_i \leq 8)$

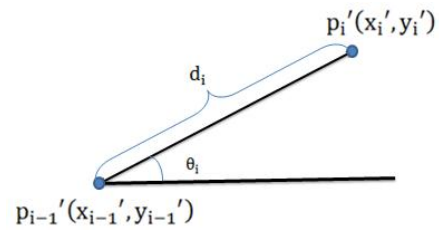


Figure 3: Direction vector

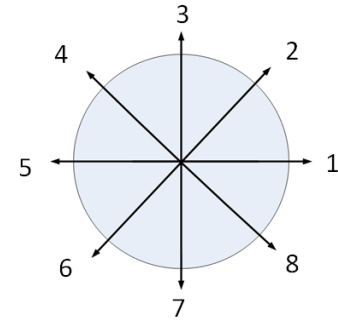


Figure 4: Eight direction codes

Thereby, the discrete vector is determined and then is used as input to recognition system.

5 Dynamic Gesture Recognition Based On HMM-FNN

Although gesture can be categorized in several different ways [15], when considered from the view of motion feature, gesture usually could be categorized into static and dynamic gesture. Static gesture, also called posture, is defined as a static movement, it express meaning just by means of hand shape or finger configuration, and it can be regarded as a special case of dynamic gesture. Dynamic gesture is defined as a dynamic movement, it involves a fixed posture and change in the position or orientation of the hand, such

as making a pinching posture and changing the hand's position.

It's well known that HMM model has strong ability for temporal data modeling, so we apply left-right banded HMM to model gesture trajectory. Fuzzy Neural Network has strong ability for fuzzy rule modeling and fuzzy inference due to its integration of fuzzy set theory and Neural Network together. Since traditional FNN cannot model temporal data and conventional HMM do not own ability for fuzzy inference [16], we integrate the two models together to represent complex gesture trajectory and perform inference by the integrated HMM-FNN model, for the recognition of dynamic gesture.

HMM-FNN model includes five layers. Its first layer, second layer and HMM layer constitute the fuzzy preprocessing part, third layer and fourth layer constitute fuzzy inference part, fifth layer is the defuzzification part of HMM-FNN and produce distinct output. The following will introduce these five layers in detail.

The first layer is the input layer of the model and it has three neurons, which correspond to the two movement components of dynamic gesture: Q_T and Q_S , respectively. Therein, Q_T express the changing sequence of the 2D gesture trajectory and Q_S express the changing sequence of the palm area.

The second layer and HMM layer compose fuzzification layer. Each HMM model is related to a neuron in second layer, which represent a fuzzy class to which the input observation possibly belongs. The likelihood of input observation sequence Q to each HMM model $p(Q/\lambda)$ is considered as membership value of the corresponding fuzzy class variable. At the same time, the neurons in second layer constitute the antecedent part (conditional part) of fuzzy rule. The number of neurons of this layer is $m_1 + m_2$, where m_1, m_2 are the class numbers of 2D trajectory and movement in the Z-axis direction respectively.

The third layer is the layer of fuzzy inference, and each neuron represents a fuzzy rule. The connecting weights between neurons in second and third layer imply the contribution degree of the antecedent part for

this rule. The output of neuron in third layer is calculated as shown:

$$O^{(3)} = b = \sum_{i=0}^m \omega_i I_i^{(3)} = \sum_{i=0}^m \omega_i p\left(\frac{Q}{\lambda}\right) \quad (5.1)$$

Where $\sum_{i=0}^m \omega_i = 1$.

The fourth layer is normalization layer, the neuron number of which is equal to that of third layer. In order to speed up convergence of the network during training, the output of third layer is normalized to assure the sum of them is equal to 1. Output of its neuron is shown as

$$O_i^{(4)} = I_i^{(4)} / \sum_{i=0}^m I_i^{(4)} \quad (5.2)$$

The fifth layer is the defuzzification layer, the output of which is shown as

$$O^{(5)} = \sum_{j=1}^N \omega_j O_j^4, \quad \sum_{j=1}^N \omega_j = 1 \quad (5.3)$$

Where ω_j implies the importance of each rule for the final classification output, N is the total number of fuzzy rules.

Suppose that complex gesture trajectory has already been decomposed into two independent parts during hand tracking. The feature sequences are considered as input of HMM-FNN model, and calculate the likelihood of HMM model according to forward probability method. The isolated and continuous gestures paths are recognized by its discrete vector and HMM Forward algorithm corresponding to maximal gesture models over the Viterbi best path. Moreover, BW algorithm is used to do a full training for the initialized HMM parameters to construct gestures database.

We choose left-right banded model [18] as the type of HMM model due to its straightforward structure. Corresponding to the features' type, the type of HMM models for posture changing and movement in Z-axis direction are one-dimensional continuous HMM models, while that of 2D trajectory is a one-dimensional discrete one. As for continuous HMM model, we employ Gaussian Mixture Model (GMM) as the emission probability of observation, which has the likelihood as described in Eq.5.4.

$$p\left(\frac{O}{\lambda}\right) = \sum_{i=1}^M \omega_i g_i(x) \quad (5.4)$$

Where ω_i is the weight of i^{th} Gaussian component, $g_i(x)$ is one-dimensional Gaussian density function $g_i(x) \sim N(\mu, \sigma^2)$

Suppose that complex gesture trajectory has already been decomposed into three independent parts during hand tracking. The three feature sequences are considered as input of HMM-FNN model, and calculate the likelihood of HMM model according to forward probability method. The final output of HMM-FNN model indicates the class type to which the input gesture belongs, such as the output of trajectory A is between the range $(\alpha, \beta]$ and trajectory B is between $(\beta, \gamma]$ and so on. The continuous gestures paths are recognized by its discrete vector and HMM Forward algorithm corresponding to maximal gesture models over the Viterbi best path. Moreover, BW algorithm is used to do a full training for the initialized HMM parameters to construct gestures database.

6 Experimental Results

In our experiments, we evaluate the performance of proposed model by comparison with conventional HMM model, through 357 times tests, we get the results as shown in Table1. The experimental procedure and results are shown as Figure 5. And through our experiment, we find that several characters are similar to each other in the feature space, for example, ‘2’and ‘Z’. These similar characters we call them confusion characters (Table2).

Orientation dynamic features are obtained from spatiotemporal trajectories and then quantized to generate its code words. The algorithm we present have the best performance and achieves average rate recognition 95.76% and 93.64% for Arabic Numbers and Alphabets, respectively (as show in Table 1).

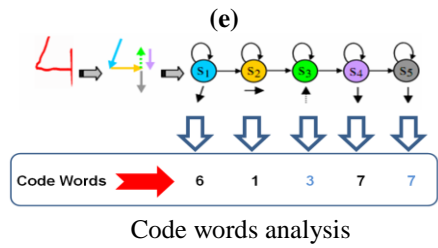
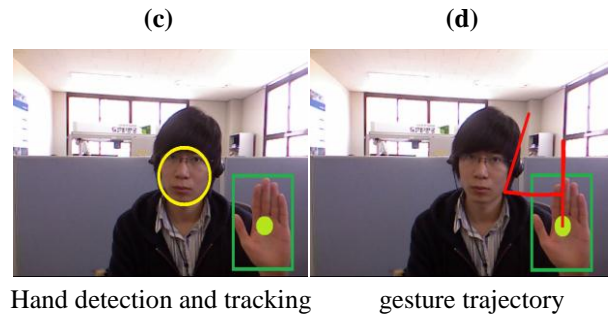
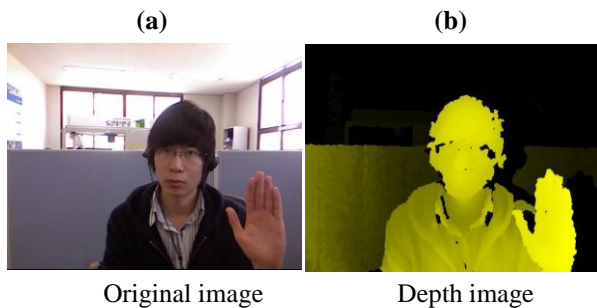


Figure 5: The experimental procedure and results

Table 1: Comparison of recognition rate between proposed method and conventional HMM model

Type	Average Recognition Rates	
	HMM	HMM-FNN
Arabic Numbers(0-9)	93.72%	95.76%
Alphabets (A-Z)	89.8%	93.64%

Table 2: Confusion Character

Type	Confusion Character				
Character	0	1	2	C	D
Similar	O	I	Z	G	P

7 Conclusion

In this paper, we propose an automatic system for recognizing continuous gestures in real-time, including Arabic numbers (0-9) and alphabets (A-Z). An improved method based on YCbCr and HSI mixed skin color space used for hand area detection and segmentation, the improved CAMSHIFT algorithm used for hand tracking. Orientation dynamic features are obtained from spatiotemporal trajectories and then

quantized to generate its code words. An improved HMM-FNN model is proposed for gesture recognition based on the code words, which combines ability of HMM model for temporal data modeling with that of fuzzy neural network for fuzzy rule modeling and fuzzy inference. In this proposed system, the algorithm presents satisfactory performance and achieves average rate recognition 95.76% and 93.64% for Arabic Numbers (0-9) and Alphabets (A-Z), respectively.

Acknowledgements: This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the IT/SW NHN Program supervised by the NIPA (National IT Industry Promotion Agency) (“NIPA-2011-(C1820-1102-0010)).

References

- [1] H.S. Yoon, J. Soh, Y. J. Bae and H.S. Yang, "Hand gesture recognition using combined features of location, angle and velocity," *Pattern Recognition* (2001), vol. 34, 1491-1501.
- [2] N. Liu, B. Lovel, and P. Kootsookos, "Evaluation of hmm training algorithms for letter hand gesture recognition," in *Proc. IEEE Int'l Symp. on Signal Processing and Information Technology* (2003), 648- 651.
- [3] E. Hunter, J. Schlenzig, and R. Jain, "Posture estimation in reduced-model gesture input systems," in *Proceedings of International Workshop on Automatic Face-and Gesture-Recognition*,(1995), 290-295.
- [4] X. Deyou. A Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG. In *ICPR 06 Conference*, (2006), 519-522.
- [5] M. Elmezain, A. Al-Hamadi, and B. Michaelis. Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences. *W S C G Journal* (2008), Vol.16 (1), 65-72.
- [6] E. Holden, R. Owens, and G. Roy. Hand Movement Classification Using Adaptive Fuzzy Expert System. *Expert Systems Journal*(1996), Vol. 9(4), 465-480.
- [7] L.R.Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proc. IEEE*, vol.77,no.2, (1989), 257-286.
- [8] Wenkai Xu and Eung-Joo Lee, "Improved Hand Detection and Gesture Recognition Algorithm," *Korea Multimedia Society conference* (2011), 99.
- [9] Bai Yu and Eung-Joo Lee, "The hand mouse: Hand detection and hand tracking," *International Conference on Multimedia, Information Technology and its Application 2009(MITA 2009)*, 244-245.
- [10] Xia Liu, Hongxia Chu and Pingjun Li, "Research of the Improved Camshift Tracking Algorithm," *IEEE Conference on ICMA 2007*, 968~972.
- [11] Ruifeng Yuan, et al. "View Invariant Hand Gesture Recognition using 3D Trajectory," *IEEE Conference* (2010), 6315 – 6320.
- [12] Forsyth D A, Ponce J. "Computer Vision: A Modern Approach," Prentice hall, 2002.
- [13] M. Elmezain, A. Al-Hamadi, and B. Michaelis. "Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences," *W S C G Journal*, Vol. 16(1), (2008), 65-72.
- [14] L. Nianjun, C. L. Brian, J. K. Peter, and A. D. Richard. "Model Structure Selection & Training Algorithms for a HMM Gesture Recognition System," In *International IWFHR* (2004), 100-106.
- [15] Valdimir I. Pavlovic, R. Sharma, T. S. Huang. "Visual interpretation of hand gesture for Human-Computer Interaction: a review," *IEEE PAMI*, vol.19 (7), (1997), 677-695.
- [16] Xiyang Wang, Guozhong Dai, "A Novel Method to Recognize Complex Dynamic Gesture by Combining HMM and FNN Models," *Computational Intelligence in Image and Signal Processing, CIISP 2007. IEEE Symposium*,(2007), 13-18.
- [17] Shah, C.A., Y.W. Sheng and L.C. Smith, Automated Image Registration Based on Pseudo invariant Metrics of Dynamic Land-Surface Features. *IEEE Transactions on Geoscience and Remote Sensint*. Vol.46, No.11, (2008), 3908-3916.
- [18] Lawrence R. Rabiner. "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of IEEE*, 77(2), (1989), 257-286.



recognition.

Wenkai Xu received his B. S. at Dalian Polytechnic University in China (2006-2010). Currently, he is studying in Department of Information and Communications Engineering Tongmyong University, Korea for master degree. His main research areas are image processing, computer vision, biometrics and hand



Eung-Joo Lee received his B. S. , M. S. and Ph. D. in Electronic Engineering from Kyungpook National University, Korea, in 1990, 1992, and Aug. 1996, respectively. Since 1997 he has been with the Department of Information & Communications Engineering, Tongmyong University, Korea, where he is currently a professor. From 2000 to July 2002, he was a president of DigitalNetBank Inc.. From 2005 to July 2006, he was a visiting professor in the Department of Computer and Information Engineering, Dalian Polytechnic University, China. His main research interests includes biometrics, image processing, and computer vision.