Human Computer Interface for Gesture-Based Editing System

Ho-Sub Yoon[+*], Byung-Woo Min[+], Jung Soh[+], Young-lae Bae[+] and *Hyun Seung Yang
[+]Image Processing Div. / Computer Software Technology Lab. in ETRI
161, Kajung-Dong Yusung-Ku Taejon, Korea 305-350
*Department of Computer Science / Korea Advanced Institute of Science and Technology
373-1 Kusung-Dong Yusung-Ku Taejon, Korea 305-701
e-mail : yoonhs@etri.re.kr

Abstract

The use of hand gesture provides an attractive alternative to cumbersome interface devices for human-computer interactio n(HCI). Many methods for hand gesture recognition using visual analysis have been proposed such as syntactical analysis, neural network(NN), Hidden Markov Model(HMM) and so on. In our research, a HMM is proposed for alphabetical hand gesture recognition.

In the preprocessing stage, the proposed approach consists of three different procedures for hand localization, hand tracking and gesture spotting. The hand location procedure detects the candidated regions on the basis of skin-color and motion in an image.. The hand tracking algorithm finds the centroid of a moving hand region, connect those centroids, and thus, produces a trajectory. The spotting algorithm divides the trajectory into real and meaningless gestures.

In constructing a feature database, the proposed approach use the location, angle and velocity feature code, and employ a k-means algorithm for codebook of HMM. In our experiments, 2400 trained gestures and 2400 untrained gestures are used for training and testing, respectively. Those experimental results demonstrate that the proposed approach yields a higher and satisfying recognition rate with various gestures.

1.Introduction

Hand gesture recognition using visual devices has a number of potential application in HCI (human computer interaction), VR(virtual reality), machine control in the industry field, and so on[1,2]. Most conventional approaches to hand gesture recognition has employed external devices such as datagloves, maker and so on. But, for more natural interface, hand gesture must be recognized from visual images without any external devices.

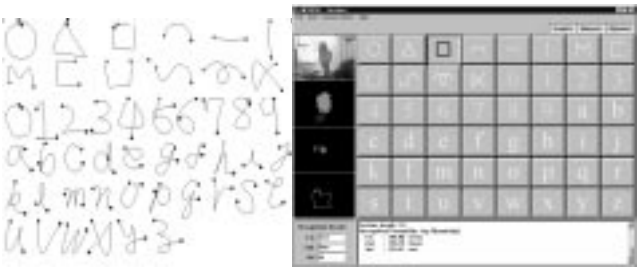
Many methods for hand gesture recognition using visual device have been proposed such as syntactical analysis, neural based approach, HMM (hidden markov model) based recognition[3,4]. As gesture is the continuous motion on the sequential time series, HMM must be a prominent recognition tool.

Several hand gesture recognition systems have been developed using various features computed from static images or image sequences[5]. Segan[6] used edge-based technique to extract image parameters from simple silhouettes and developed a system which can recognize 10 distinct pose in real-time. Hunter[7] used the Zernike moments as the image features and developed a system in which the sequence of hand gesture were recognized using HMM. Starner[3] used image geometry parameter as the image features and employed a HMM five states topology for the gesture classification.

In our research, we consider the planar hand gesture in front of camera and detect 16-dimension location codes as input vectors for the HMM network. We use a simple context modeling on “left-to-right” HMM model with 5 states. Our system is applied to 2 types gesture such as 12 graphic element and 10 Arabia digit element. This work explores the use of hand gesture in a realistic setting for the control the graphic editor. We don’t expect direct replacement of mouse, but some of part of the next generation user interface is hand gesture recognition with image sensor. In this paper, we are intended to develop a graphic editor system operated by hand gestures. The gestures are classified into two types. One is the graphic gestures which are six drawing elements such as circle, triangle, rectangle, arc, horizontal line, vertical line, and six edit commands such as move, copy, undo, swap, remove, close. The other gesture is the 10 Arabia digits from 0 to 9 and 26 alphabetical characters as shown in Figure 1 (a).

Our system's main screen is shown in Figure 1 (b). A camera placed with monitor and gives a sequence of image like those in Figure 2. The screen show that the rectangle's gesture trace is captured by system and recognition resulted by HMM is displayed in the low side.

In this paper, we mainly discuss the recognition of dynamic gestures by HMM. This paper is composed of 5 sections. In the 2nd section, we explain the preprocessing algorithm using image processing. Section 3 explains feature extraction method. In the 4th section, the recognition using the HMM is described. Finally, we have a conclusion in section 5.



(a) Target gesture (b) System GUI

Figure 2. Human computer interaction by hand gestures

II. Preprocessing algorithm using Image Processing

2.1 Hand Location Algorithm Using Color Analysis

Hand detection is the first step in the visual image-based gesture recognition, and therefore directly influences the recognition rate. Our research exploits a priori knowledge such as skin-color, hand size, and positional information through the following steps.

Step 1) Color system Conversion from RGB to YIQ
Step 2) Estimate the similarity measures between model and input regions
Step 3) Threshold the similarity measures
Step 4) Noise removal and dilation
Step 5) Detection of hand candidate regions
Step 6) Selection of a hand region

Color is one of the most distinctive clues to find objects and is generally input as RGB information via a CCD or a video camera. The RGB color system is known to be sensitive to the conditions of lighting and takes much processing cost, since the RGB includes the mixed information of color and intensity. Therefore, the YIQ color system that is known to be less sensitive to lighting than the RGB is adopted. In the YIQ, Y means intensity, while I and Q represent color information. To reduce the effect of lighting and the processing cost, only I and Q values are used to build up a skin-color model and to extract hand areas from input images by calculating the similarity between a model and an input image. A skin-color model is constructed at the initial stage by an interactive method, and automatically updated using new skin-color information at the intermediate stages for extracting hand areas. It aims at maintaining a more robust one that could be adaptive to changing environmental conditions. Fig. 2 shows a skin color model constructed at the initial stage.

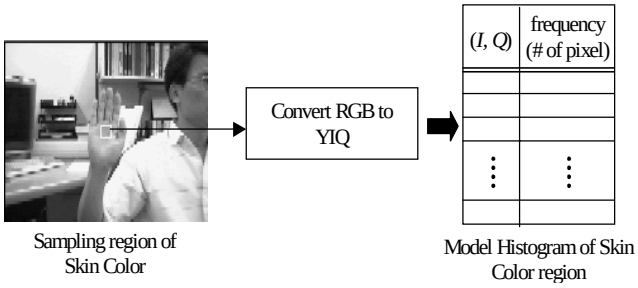


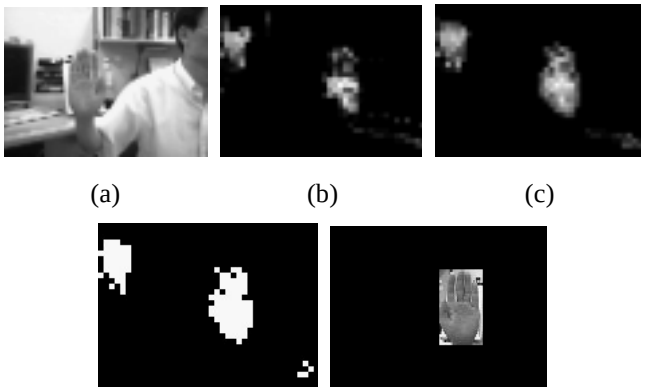
Figure 2. Skin color model histogram on I-Q space

In order to detect hand regions from video images, the skin-color similarity regions are firstly observed. Since image processing by a pixel unit takes much computing-cost, the input image is processed on the basis of a cell unit, 4 x 4 pixels. The similarity between a model histogram and a current cell is calculated using the histogram intersection method proposed by Swain and Ballard[8].

$$S_{M,I} = \frac{\sum_{i,q} \min(H^M(i,q), H^I(i,q))}{\sum_{i,q} H^M(i,q)}$$

H^M means the *i-q* histogram value of the model and H^I does that of an input cell. Using (1), the similarity cells are selected when they produce greater similarity values than a given threshold. Then, the selected cells are joined with neighboring similarity cells and become candidate regions. A candidate region might contain some false similarity regions in the background and also in general holes whose similarity is very low. This paper observes the regions usually composed of one or two cells and a few neighboring similarity regions, and then removes small and isolated regions not likely to be actual skin regions. After the above processing, the dilation process is applied by

increasing the similarity of cells surrounded by cells with relatively high similarity. As the next step, the remaining candidate regions are binarized using a threshold value. Since a fixed threshold value is not adaptive to the changes in lighting and background, this paper uses a variable threshold which is calculated using a method proposed by Otsu[9]. Using this method has the effect of adapting the threshold value to the overall similarity level of the skin-color regions. After the binarization, connected components of the binary image are located, and then each connected component becomes a hand candidate region. In Figure 3, (b), (c), (d), (e) are reversibly displayed on the horizontal axis in order to make the directions of moving hand and output image consistent.



(a) (b) (c)

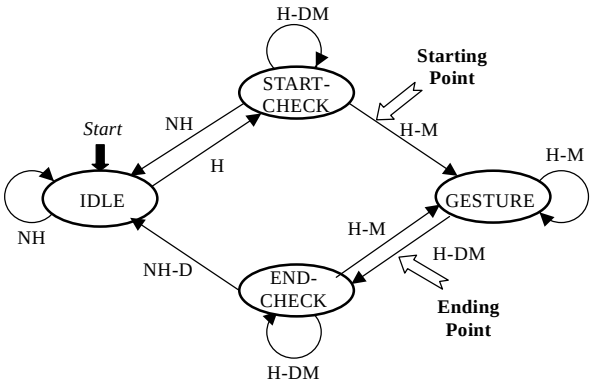
(d) (e)

Figure 3. Images in each processing step: (a) input image, (b) skin color similarity regions, (c) image after noise removal and dilation, (d) hand candidate regions, (e) detected hand region

Finally, it is required to detect the hand region from multiple candidate regions. The major difficulty in this step is that the face region can be usually one of the hand candidate regions, because it also has the almost same skin color. This paper uses *a priori* knowledge such as the hand location in a previous video image, the location where the face is usually positioned, and the size of a hand region. The following inference is done to determine a hand area from several candidates.

2 .2 Spotting Algorithm

The last step of preprocessing is pure gesture detection. The one gesture of our system is generated during the hand region is existing the screen. Therefor, the garbage movements which are the previous and post period of pure gesture are included the input gesture. To remove this garbage gesture, we use a spotting rule. The mean of spotting rules is that the user stops in a minute before meaningful gesture start and after meaningful gesture stop. Using this rule, the standstill area is responds to spotting area and the gesture between two spotting areas is pure gesture. In the gesture trace which consist of sequence of x, y position, the spotting area is detected by following Figure 4.



Symbol Definition

Symbol	Results of Hand Extraction	Results of Hand Tracking
H	Hand exists	Hand appeared and moved
NH	No hand exists	None
H-DM	Hand exists	Hand appeared but didn't move
H-M	Hand exists	Hand appeared and moved
NH-D	No hand exists	Hand disappeared

Figure 4. States for gesture spotting

III. Feature Extraction

The performance of a general recognition system firstly depends on how to get the efficient features representing the characteristics of patterns. There are several methods for representing features of a gesture trajectory such as using 2-dimentional edge, time edge, raw position, Cartesian velocity, polar velocity and angular velocity. There are also many other types of features such as chain code, mesh code, momentum, MRF and so on. But all of the features are based on only three basic informations from a gesture trajectory such as location from origin, orientation and velocity.

3.1 Location feature

The most basic feature from gesture trajectory is a location feature from origin. Each gesture's trajectory

points are composed of different x-y coordinate series at Cartesian coordinate system. The first step to use location feature is to detect points of MBR(minimum bounding rectangle) including one gesture for location normalization. The second step is to divide the MBR area into feature space. The k-means clustering algorithm is used for this purpose and described next section in detail. The proper number of feature space is decided on many experimental results.

3.2 Orientation feature

Chain code is a useful coding method for orientation feature. General chain code is based on grid and represented the 4-connectivity and 8-connectivity according to the connectivity. This method can be compactly enclosed by indicating the directions of transitions along their contours. A 8-connectivity code is assigned to each of the possible directions of moving along a contour, where 0 is assigned to movement to the right, and the codes increase by the unity going around counterclockwise to 7.

The basic problem in chain coding for contour tracing, starting point is not important at gesture trajectory coding. The gesture's start spotting point is set to starting point. Because the neighboring two point's distance in gesture trajectory is greater than 1, the limit of connectivity is set to free in our system. Following algorithm is chain code generation algorithm between two gesture points (X_t, Y_t) and (X_{t+1}, Y_{t+1}). NumOfPoints means x-y points including one gesture trajectory and NumOfChain means arbitrary number of chain code. The Optimal number of chain code is described in next section of experimental results.

```
for (i = 0; i < NumOfPoints; i++) {
    dx = X_i -X_{i+1};
    dy = Y_i -Y_{i+1};
    a = atan2(dy, dx);
    ChainCode[i] = (NumOfChain - (int)(a / (PI / 4) + 0.5 + (dy
    < 0) * NumOfChain)) % NumOfChain;
}
```

3.3 Velocity feature

The last information from gesture trajectory is a velocity feature. This assumption is based on the factor that each gesture is made by different speed. Namely, the simple gesture such as circle gesture have unique speed and complex gesture such as q or w gesture have various speed during gesture trajectory generation. This velocity feature is extracted by difference of each connected two points such as some points (X_t, Y_t) and (X_{t+1}, Y_{t+1}). The following equation is described a generation method for velocity feature v_t.

$$V_t = \sqrt{(X_t - X_{t+1})^2 + (Y_t - Y_{t+1})^2}$$

$$V_{max} = \max\left(\sum_{t=1}^{n} V_t\right), \quad v_t = \frac{V_t}{V_{max}}$$

IV. Recognition using the HMM

4.1 Hidden Markov Model (HMM)

The HMM models are double stochastic processes as governed by an underlying Markov chain with a finite number of states, and a set of random functions each of which is associated with one state[12]. In the discrete time instants, the process is in one of the states, and generates an observation symbol according to the random function corresponding to the current state. The model is hidden in the sense that all that can be seen is a sequence of observations. Quantitatively, a HMM is described as following :

- set of observation string $O = \{O_1,...,O_T\}$, where $t = 1,...,T$
- set of N states $\{s_1,....,s_N\}$
- set of k discrete symbols from a fine alphabet $\{v_1,...,v_k\}$
- a state transition matrix $A = \{a_{ij}\}$, where a_{ij} is the transition probability from state s_i to s_j: $A = \{a_{ij}\} = Pr(s_j \text{ at } t+1 | s_i \text{ at } t)$, $1 \le i, j \le N$;
- an observation probability matrix $B = \{b_{jk}\}$, where b_{jk} is the probability of generating symbol v_k from state q_j
- the initial probability distribution for the states $\prod = \{\pi_j\}$, $j = 1,2,….,N$; $\pi_j = Pr(s_j \text{ at } t = 1)$

From the described definitions, the complete parameter set of the HMM can be expressed compactly as $\lambda = (A, B, \pi)$. There are in general three basic problems that must be solved for the real application of HMM: classification (evaluation), decoding, and training. The solutions to those problems are in general processed as the Forward-Backward algorithm, the Viterbi algorithm, and the Baum-Welch algorithm.

The generalized topology of HMM is a fully connected structure known as an ergodic model, where every state of the model could be reached from every other state of the model. In this model, the state index transits only from left to right as time increases and this property is well applied to the dynamic gesture recognition. While, it is clear that the left-right model is just same to the ergodic model rearranged using the transition conditions. The global structure of HMM is constructed by parallel connection of each HMM (λ_1, λ_2 , …λ_M) and thus adding of new HMM or deleting of existing HMM can be easily done. Where λ means constructed HMM model of each gestures and M means number of gestures for recognition.

4.2 Observation sequence generation using Vector Quantization

The extracted features are quantized to obtain discrete symbols for applying to the HMM. From our trajectory of gesture in location, we make the discrete symbols using k-means vector quantization algorithm[9]. The k-means algorithm is adopted to classify the gesture-tokens into L clusters on the feature space. This algorithm is based on the minimum distance between the center points of each cluster and feature points.

The codebook consists of the symbol number and the centroid coordinate of each cluster. In the experiments, the symbol codes of observations can be determined using the distance between an observation and the centroid of each cluster.

4.3 Training and Experiment Results

An experimental system using HMM for alphabetical hand gesture recognition system was implemented on a personal computer with an image capture board(Matrox Meteor). Also, an input image was captures by a CCD camera with the resolution 120x160. The computing power is 5 ~ 7 frames per second. Since our computer can process over 5 frames per second, it is possible to use the proposed system for real-time interaction. One gesture captured by our system has 20 ~ 30 traced points. Therefore, the total processing time including recognition takes 4 ~ 6 seconds for one gesture.

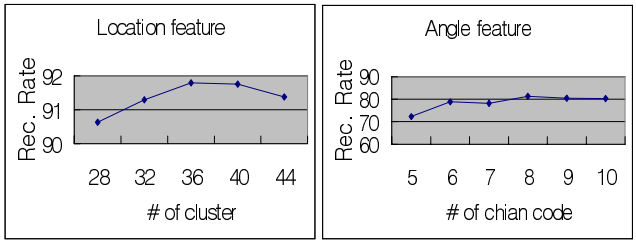
The recognition software is implemented in Visual C++ 5.0 on Windows 95. The proposed algorithm in this paper was applied to the database containing 4800 alphabetical gestures of 20 persons. Each person was asked to draw his/her alphabetical gestures 5 times of each 48-input character. There were two databases such as training and testing data constructed for approving this proposed algorithm. To divide database, we scan the 4800 alphabetical gestures using interlaced scan method, and then even gestures saved training data set and odd gestures saved testing data set. Our gesture database is available by mailing request with the yoonhs@etri.re.kr. This database, however, contains the sequences of x-y coordinates representing unspotted gestures.

From the proposed system, the sequence of x-y coordinate is converted into location, chain, and velocity feature code, and this observation sequence for HMM is quantified by k-means clustering algorithm. The higher recognition rates using HMM are 97.6% and 91.8% for training data and testing data (see Table 1). The Table 1 shows the results using 2400 training data and 2400 testing data. As expected, training data yields a higher recognition rate than testing data.

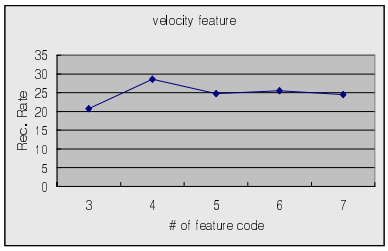
Table 1. Total test results

feature space	# of feature code	Recognition Results(%)		
		Training Data	Testing Data	Overall
location	36	97.58	91.79	94.69
chain	8	93.67	81.29	87.48
velocity	4	37.21	28.53	32.87

The next Fig. 5 (a), Fig. 5 (b) and Fig. 5 (c) show the optimal number of feature code according to the change of number of feature code. From our experimental results, we can get the important information that the location and angle feature are more descriptive features than velocity feature. The some reason of low recognition rate in velocity feature may be inconsistent processing speed at preprocessing step and multi-processing OS, Windows 95.



(a) location feature (b) angle feature



(c) velocity feature

Figure 14. the optimal number of feature code according to the change of number of feature code.

V. Conclusion

Recently research in hand gesture recognition aims at applying to sign language recognition, control of household electronic appliances, human-computer interaction, VR, and so on. In this paper, we are intended to develop a graphic editor system operated by hand gestures. We don't expect direct replacement of mice, but some of part of the next generation user interface is hand gesture recognition with image sensor.

For the hand gesture recognition, we have 4 main steps. The first step is the hand location algorithm that detects skin-colored regions in an image by using a color histogram matching technique. And second step is the spotting algorithm, which detects a spotting area using domain knowledge. Thirdly, the feature detection algorithm is occurred in location, angle and velocity features that consist of MBR including meaningful one gesture. The final step is recognition algorithm using HMM.

HMM is a very effective tool for hand gesture recognition as a time series domain. As the batch test results using file information, our method has good results over 92% success in any case. Also, the on line test results that use the direct camera input and real time HMM recognition has a positive result 87% success.

Feature research can focus on two areas. The first one is the enhanced hand location algorithm using color analysis because color information is very sensitive under image capturing environments. The second one is new feature detection for HMM because we doesn't fully reflect the characteristics of the dynamic gesture such as the optical flow information, the hand posture, momentum, curvature, and so on.

References

[1] W. T. Freeman and C. D. Weissman, "Television control by hand gesture," *Proceedings of Workshop on Automatic Face- and Gesture-Recognition*, pp. 179-183, June 1995.

[2] J. S. Kim, et al., "Real-Time Hand Gesture Recognition for Avatar motion control, " Proceedings of HCI'97, pp. 96-101, February 1997.

[3] T. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Model," *Proceedings of Workshop on Automatic Face- and Gesture-Recognition*, pp. 189-194, June 1995.

[4] J. Yang, et al., "Human Action Learning via Hidden Markov Model, " IEEE Trans. On Systems, Man, and Cybernetics, Vol. 27, No. 1, pp. 34-44, January 1997.

[5] T. S. Huang,and A. Pentland, "Hand gesture modeling, analysis, and synthesis", *Proceedings of Workshop on Automatic Face- and Gesture-Recognition*, pp.73-79, Zurich, Switzerland, June 1995.

[6] Jakub Segan, "Controlling computer with gloveless gesture", *in Virtual Reality System'93*, pp.2-6, 1993

[7] E. Hunter, et al. "Posture Estimation in Reduced-Model Gesture Input Systems", *Proceedings of The First Automatic Gesture and Face Recognition*, pp. 290-295, June 1995.

[8] M. J. Swan and D. H. Ballard, "Color indexing," International Journal of Computer Vision, pp.11-32, 1991.

[9] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.

[10] L. R. Rabiner, "A tutorial on hidden Markov models and selected application in speech recognition," *Proceedings. IEEE 77*, pp. 267-293. 1989.