GVIP 05 Conference, 19-21 December 2005, CICC, Cairo, Egypt

www.icgst.com

Real-Time Hand Tracking and Gesture Recognition System

Nguyen Dang Binh, Enokida Shuichi, Toshiaki Ejima
Intelligence Media Laboratory, Kyushu Institute of Technology
680-4, Kawazu, Iizuka, Fukuoka 820, JAPAN
[ndbinh, enokida, toshi]@mickey.ai.kyutech.ac.jp

Abstract

In this paper, we introduce a hand gesture recognition system to recognize real time gesture in unconstrained environments. The system consists of three modules: real time hand tracking, training gesture and gesture recognition using pseudo two dimension hidden Markov models (P2-DHMMs). We have used a Kalman filter and hand blobs analysis for hand tracking to obtain motion descriptors and hand region. It is fairy robust to background cluster and uses skin color for hand gesture tracking and recognition. Furthermore, there have been proposed to improve the overall performance of the approach: (1) Intelligent selection of training images and (2) Adaptive threshold gesture to remove non-gesture pattern that helps to qualify an input pattern as a gesture. A gesture recognition system which can reliably recognize single-hand gestures in real time on standard hardware is developed. In the experiments, we have tested our system to vocabulary of 36 gestures including the America sign language (ASL) letter spelling alphabet and digits, and results effectiveness of the approach.

Keywords: *Hand gesture recognition; Hand tracking; Kalman filter; Pseudo 2-D Hidden Markov models.*

1. Introduction

Gesture recognition is an area of active current research in computer vision. It brings visions of more accessible computer system. In this paper we focus on the problem of hand gesture recognition using a real time tracking method with Pseudo two-dimensional hidden Markov models (P2-DHMMs). We have considered single-handed gestures, which are sequences of distinct hand shapes and hand region. A Gesture is defined as a motion of the hand to communicate with a computer.

Many approaches to gesture recognition have been developed. Pavlovic [1] present an extensive review of existing technique for interpretation of hand gestures. A large variety of techniques have been used for modeling the hand. An approach based on the 2-D locations of fingertips and palms was used by Davis and Shah [2]. Bobick and Wilson [3] have developed dynamic gestures have been handled using framework. A state-based technique for recognition of gestures in which the define the feature as a sequence of states in a measurement or configuration space. Human-computer interaction using hand gestures has been studied by a number of researchers like Starner and Pentland [4], and Kjeldsen and Kender [5]. Use of inductive learning for hand gesture recognition has been explored in [6]. Yoon et al. [8] have proposed a recognition scheme using combined features of location, angle and velocity. Lockton et al. [9] propose a real time gesture recognition system, which can recognize 46 ASL, letter spelling alphabet and digits. The gestures that are recognized [9] are 'static gesture' of which the hand gestures do not move. Several system use Hidden Markov models for gesture recognition [7,17].

This research is focused on the application of the HMM method to hand gesture recognition. The basic idea lies in the real-time generation of gesture model for hand gesture recognition in the content analysis of video sequence from CCD camera. Since hand images are two-dimensional, it is natural to believe that the 2-DHMM, an extension to the standard HMM, will be helpful and offer a great potential for analyzing and recognizing gesture patterns. However a fully connected 2-DHMMs lead to an algorithm of exponential complexity (Levin and Pieraccini, 1992). To avoid the problem, the connectivity of the network has been reduced in several ways, two among which are Markov random field and its variants (Chellapa and Chatterjee, 1985) and pseudo 2-DHMMs (Agazzi and Kuo, 1993). The latter model, called P2-DHMMs, is a very simple and efficient 2-D model that retains all of the useful HMMs features. This paper focuses on the real-time construction of hand gesture P2-DHMMs. Our P2-DHMMs use observation vectors that are composed of two-dimensional Discrete Cosine Transform (2-D DCT) coefficients. In addition, our gesture recognition system uses both the temporal and characteristics of the gesture for recognition. Unlike most other schemes, our system is robust to background clutter, does not use special glove to be worn and yet runs in real time. Although use to our knowledge for recognition is

not new but this approach first time is introduced to the task of hand gesture recognition. Furthermore, the method to combine hand region and temporal characteristics in P2-DHMMs framework is new contribution of this work. Use of both hand regions, features of location, angle, and velocity and motion pattern are also novel feature in this work.

The organization of the rest of the paper is as follows. Section 2 describes overview of the gesture recognition scheme. In Section 3, we discuss our tracking framework. We discuss hand gesture training and gesture recognition method in Section 4. The next section presents results of experiments and finally. The summarize the contribution of this work and identify areas for further work in the conclusion section.

2. Overview of The Gesture Recognition Scheme

In this paper we only consider single handed postures. A gesture is a specific combination of hand position, orientation, and flexion observation at some time instance. Our recognition engine identifies a gesture based upon the temporal sequence of hand regions in the image frame. The recognition process involves tracking of the gesturer's hand. The hand region being tracked is recognized by blob analysis and Kalman filter based approach. P2-DHMMs based approach uses shape and motion information for recognition of the gesture. The algorithm proposed in our recognition engine is show in figure 2:

1. Choose initial search window size and location.
2. While hand is in view,

(a) Track and extract the hand from an image sequence.

(b) Verify the extracted hand region.

3. Using P2-DHMMs recognize the gesture, which gives maximum probability.

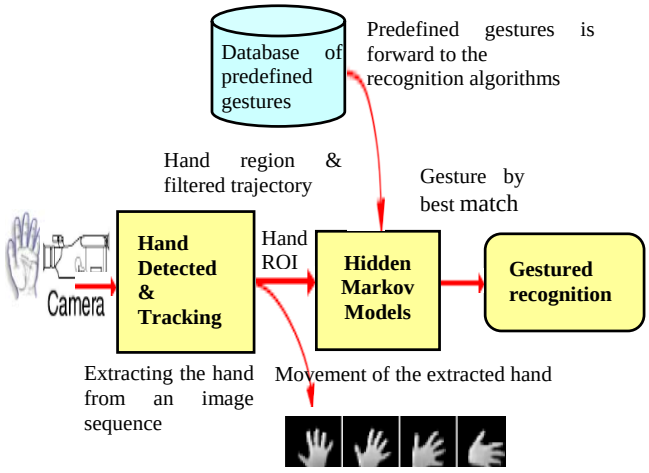


Figure 2: System Overview

3. Hand Tracking Framework

We develop a real time hand tracking method based on the Kaman filter and hand blobs analysis, which is robust and reliable on hand tracking in unconstrained environments and then the hand region extraction fast and accurately. We need to consider the trade-off between the computation complexity and robustness. The segmented image may or may not include the lower arm. To avoid this, the user is required to wear long-sleeves shirt. This is necessary to avoid using a wrist-cropping procedure, which is more inconvenient to users.

3.1 Hand detection in an image frame

We must first extract hand region in each input image frame in real time.

Extracting hand region. Extracting hand based on skin color tracking [20], we drew on ideas from robust statistics and probability distributions with Kalman filter in order to find a fast, simple algorithm for basic tracking. Our system identifies image regions corresponding to human skin by binarizing the input image with a proper threshold value. We then remove small regions from the binarized image by applying a morphological operator and select the regions to obtain an image as candidate of hand.

Finding a palm's center and hand roll calculation. In our method, the center of user's hand is given as first moments of the 2-D probability distribution during the course of hand tracking algorithm operation where (x, y) range over the search window, and I(x, y) is the pixel (probability) value at (x, y): The first moments are $M_{10} = \sum_x xI(x,y)$, $M_{01} = \sum_y yI(x,y)$.

Then the centroid is $x_c = M_{10}/M_{00}$, $y_c = M_{01}/M_{00}$.

Second moments are $M_{20} = \sum_x \sum_y x^2 I(x,y)$,

$$M_{02} = \sum_x \sum_y y^2 I(x,y).$$

then the hand orientation (major axis) is

$$\theta = \frac{\arctan\left(\dfrac{2\left(\dfrac{M_{11}}{M_{00}} - x_c y_c\right)}{\left(\dfrac{M_{20}}{M_{00}} - x_c^2\right) - \left(\dfrac{M_{02}}{M_{00}} - y_c^2\right)}\right)}{2}$$

The first two Eigenvalues (major length and width) of the probability distribution "blobs" may be calculated in closed form as follows [20]. Let $a = (M_{20}/M_{00}) - x_c^2$, $b = 2((M_{20}/M_{00}) - x_c y_c)$ and $c = (M_{02}/M_{00}) - y_c^2$ then length l and width w from distribution centroid are

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

When used in hand tracking, the above equations give us hand roll, length and width as marked in figure 3.

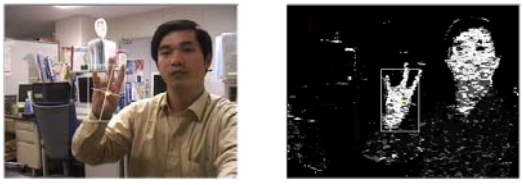


Figure 3: Orientation of the flesh probability distribution marked on the source video image from CCD camera.

3.2. Hand tracking and measuring trajectories

We use Kalman filter to predict hand location in one image frame based on its location detected in the previous frame. Kalman filter is used to track the hand region centric in order to accelerate hand segmentation and choose correct skin region when multiple image regions are skin color. Using a model of constant acceleration motion the filter provides and estimates the hand location, which guides the image search for the hand. The Kalman filter tracks the movement of the hand from frame to frame to provide an accurate starting point to search for a skin color region, which is the closest match to the estimate. Instead of segmentation the entire input image into multiple skin regions and then selecting the region. Filter estimated results are used as the starting point for the search for a skin color region in subsequent frame. The measurement vector y_t consists of the location of the center of the hand region. Therefore the filter estimate $H\hat{x}_{t-1}$ is the center of a distance-based search for a skin color pixel. First, we measure hand location and velocity in each image frame. Hence, we define the state vector as x_t :

$$x_t = (x(t), y(t), v_x(t), v_y(t))^T \quad (1)$$

Where x(t), y(t), v_x(t), v_y(t) shows the location of hand (x(t), y(t)) and the velocity of hand (v_x(t), v_y(t)) in the t^{th} image frame. We define the observation vector y_t to present the location of the center of the hand detected in the t^{th} frame. The state vector x_t and observation vector y_t are related as the following basic system equation:

$$x_t = \Phi x_{t-1} + G w_{t-1} \quad (2)$$

$$y_t = H x_t + v_t \quad (3)$$

Where Φ is the state transition matrix, G is the driving matrix, Φ is the observation matrix, w_t is system noise added to the velocity of the state vector x_t and v_t is the observation noise that is error between real and detected location. Here we assume approximately uniform straight motion for hand between two successive image frames because the frame interval ΔT is short. Then Φ, G, and H are given as follows:

$$\Phi = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) \qquad G = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (5)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (6)$$

The (x, y) coordinates of the state vector x_t coincide with those of the observation vector y_t defined with respect to the image coordinate system. Also, we assume that both the system noise w_t and the observation noise v_t are constant Gaussian noise with zero mean. Thus the covariance matrix for w_t and v_t become $\sigma_w^2 I_{2x2}$ and $\sigma_v^2 I_{2x2}$ respect, where I_{2x2} represent a 2x2 identity matrix. Finally, we formulate a Kalman filter as

$$K_t = \overline{P}_t H^T (H\overline{P}_t H^T + I_{2x2})^{-1} \quad (7)$$

$$\overline{x}_t = \Phi\{\overline{x}_{t-1} + K_{t-1}(y_{t-1} - H\overline{x}_{t-1})\} \quad (8)$$

$$\overline{P}_t = \Phi(\overline{P}_{t-1} - K_{t-1} H \overline{P}_{t-1})\Phi^T + \frac{\sigma_w^2}{\sigma_v^2} Q_{t-1} \quad (9)$$

where $\overline{x}_t$ equal $\overline{x}_{t|t-1}$, the estimated value of x_t from $y_0,...,y_{t-1}$, $\overline{P}_t$ equals $\overline{\Sigma}_{t|t-1} / \sigma_v^2$, $\overline{\Sigma}_{t|t-1}$ represents the covariance matrix of estimate error of $\overline{x}_{t|t-1}$, K_t is Kalman gain, and Q equals GG^T. Then the predicted location of the hand in the t+1th image frame is given as (x(t+1), y(t+1)) of $\overline{x}_{t+1}$. If we need a predicted location after more than one image frame, we can calculate the predicted location as follows:

$$\overline{x}_{t+m|t} = \Phi^m \{\overline{x}_t + K_t (y_t - H\overline{x}_{t-1})\} \quad (10)$$

$$\overline{P}_{t+m|t} = \Phi^m (\overline{P}_t - K_t H \overline{P}_t)\left(\Phi^T\right)^m + \frac{\sigma_w^2}{\sigma_v^2} \sum_{k=0}^{m-1} \Phi^k Q \left(\Phi^T\right)^k \quad (11)$$

where $\overline{x}_{t+m|t}$ is the estimated value of $\overline{x}_{t+m}$ from $y_0,...,y_t$, $\overline{P}_{t+m|t}$ equals $\overline{\Sigma}_{t+m|t} / \sigma_v^2$, $\overline{\Sigma}_{t+m|t}$ represents the covariance matrix of estimate error $\overline{x}_{t+m|t}$.

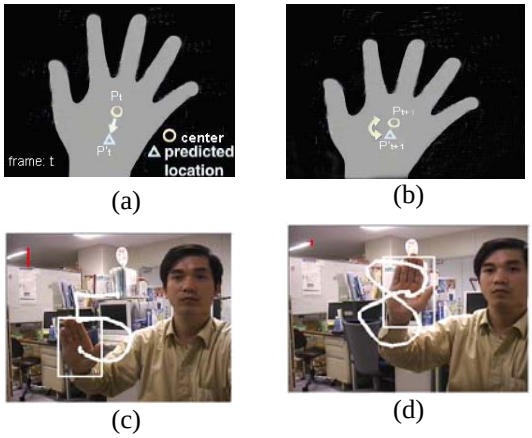


Figure 4: (a) Detecting hand's center; (b) Comparing detected and predicted hand location to determine trajectories; (c), (d) Measuring hand's center trajectories.

Determining trajectories. We obtain hand trajectory by taking correspondences of detected hand between successive image frames. Suppose that we detect hand centroid in the t^{th} image frame t. We refer to hand's location as P'_{t+1}. First, we predict the location P_{t+1} of hand in the next frame $(t+1)^{th}$ image frame t+1 with the predicted location P'_{t+1}. Finding the best combination.

4. Gesture Recognition

Hidden Markov Models are finite non-deterministic state machines, which have been successfully applied to numerous applications. They consist of a fixed number states with associated output probability density functions (pdfs) as well as transition probabilities a_{ij}. For a continuous HMM the pdf $b_j(\vec{o})$ of state S_j is usually given by a finite Gaussian mixture of the form:

$$b_j(\vec{o}) = \sum_{m=1}^{L} c_{jm} N(\vec{o}, \vec{\mu}_{jm}, \sum_{jm}) \qquad (12)$$

Where c_{jm} is the mixture coefficient for the m^{th} mixture and $N(\vec{o}, \vec{\mu}_{jm}, \sum_{jm})$ is multivariate Gaussian density with mean vector $\vec{\mu}_{jm}$ and covariance matrix $\sum_{jm}$. An HMM $\lambda(\vec{\pi}, \vec{a}, \vec{b})$ with M states is fully described by MxM-dimensional transition matrix $\vec{a}$, the M-dimensional output pdf vector $\vec{b}$ and the initial state distribution vector $\vec{\pi}$ which consists of the probabilities $\pi_j = P(q_{t=1} = s_j)$. After the model λ has been trained using the Baum-Welch algorithm feature sequences $\vec{O} = \vec{o}_1, ..., \vec{o}_T$ can be scored according to

$$\Pr(\vec{O}|\lambda) = \sum_{allQ} \Pr(O, Q|\lambda) = \sum_{q_1, q_2, .., q_T} \pi_{q_1} b_{q_1}(\vec{o}) \prod_{t=2}^{T} a_{q_{t-1} q_t} b_{q_t}(\vec{o}_t) \quad (13)$$

Here, q_t is one of the states from Q, the set of states, at time t.

Usually the likelihood $\Pr(\vec{O}|\lambda)$ is estimated by the Viterbi algorithm, which is an approximation based on the most likely state sequence (q_1,…,q_T). Although simple in form, the time requirement is exponential. Thanks to the use of the DP technique, this can be computed in linear time in T. However when it comes to 2-DHMMs formulation, even the DP technique alone is not enough. One research direction is the structural simplification of the model, and the pseudo 2-DHMMs is one solution.

4.1 Pseudo 2-DHMMs Construction

Description: Pseudo 2-DHMMs in this paper are realized as a vertical connection of horizontal HMMs (λ_k,). However it is not the only one. In order to implement a continuous forward search method and sequential composition of gesture models, the former type has been used in this research. There are three kinds of parameters in the P2DHMMs. However, since the hand image is two-dimensional, we further divided the Markov transition parameters into super-state transition and state transition probabilities; each is denoted as

$$\vec{a}_{kl} = P(r_{t+1} = l | r_t = k), \qquad 1 \le k, l \le N \quad \text{and}$$

$$a_{ij} = P(q_{t+1} = j | q_t = i), \qquad 1 \le i, j \le M$$

where r_t denotes a super-state which corresponds to a HMMs λ_k, and q_t denotes a state observing at time t. The mode has N super-states and the HMMs λ_k, is defined as standard HMM consisting of M states.

Evaluation algorithm: Let us consider a t^{th} horizontal frame, observation future vector $\vec{O}_t = \vec{o}_{1t}, ..., \vec{o}_{st}$, 1≤ t ≤T. This is a one-dimension feature sequences like that of $\vec{O}$ in Eq. (13). This is modeled by a HMM λ_k, with likelihood $P(\vec{O}_t | \lambda_k)$. Each HMM λ_k may be regarded as a super-state whose observation is a horizontal frame of states.

$$P_{r_t}(\vec{O}_t | \lambda_t) = \sum_{allQ} \Pr(\vec{O}_t, Q | \lambda_{r_t}) = \sum_{q_1, q_2, ... q_T} \pi_{q_1} b_{q_1}(\vec{o}_{1t}) \prod_{s=2}^{S} a_{q_{t-1} q_t} b_{q_t}(\vec{o}_{st}) \quad (14)$$

Now let us consider a hand region image, which we define as a sequence of such horizontal frames as $\vec{O} = \vec{O}_1, \vec{O}_2, .., \vec{O}_T$. Each frame will be modeled by a super-state or a HMM. Let Λ be a sequential concatenation of HMMs. Then the evaluation of Λ given feature sequence $\vec{O}$ of the sample image X is

$$P(\vec{O}|\Lambda) = \sum_{R} P_1(\vec{O}_1) \prod_{t=2}^{T} \vec{a}_{r_{t-1} r_t} P_{r_t}(\vec{O}_t) \qquad (15)$$

where it is assumed that super-state process starts only one from the first state. The P_{r_t} function is the super-state likelihood. Note that both of the Eqs. (14) and (15) can be effectively approximated by the Viterbi score. One immediate goal of the Viterbi search is the calculation of the matching likelihood score between $\vec{O}$ and HMM. The objective function for an HMM is defined by the maximum likelihood as

$$\Delta(\vec{O}_t, \lambda_k) = \max_{Q} \prod_{s=1}^{S} a_{q_{s-1} q_s} b_{q_s}(\vec{o}_{st}) \qquad (16)$$

where Q =q_1,q_2,…,q_s is a sequence of states of λ_k, and $a_{q_0 q_1} = \pi_{q_1}$. $\Delta(\vec{O}_t, \lambda_k)$ is the similarity score between two sequences of different length. The basic idea behind the efficiency of DP computation lies in formulating the expression into a recursive form

$$\delta_s^k(j) = \max_i \delta_{s-1}^k(i) a_{ij}^k b_j^k(\vec{o}_{st}), j = 1, ..., M_k, s = 1, ..., S, k = 1, ..., K$$

where $\delta_s^k(j)$ denotes the probability of observing the partial sequence $\vec{o}_{1t}, ..., \vec{o}_{st}$ in model k along the best state sequence reaching the state j at time/step s. Note that $\Delta(\vec{O}_t, \lambda_k) = \delta_S^k(N_k)$ where N_k is the final state of the state sequence. The above recursion constitutes the DP in the lower level structure of the P2DHMM. The remaining DP in the upper level of the network is similarly defined by

$$D(\vec{O}, \Lambda) = \max_k \prod_{t=1}^{T} \vec{a}_{r_{t-1} r_t} \Delta(\vec{O}_t, \lambda_{r_t}) \qquad (17)$$

that can similarly be reformulated into a recursive form. Here denotes the probability of transition from super-state r_1 to r_2. According to the formulation described

thus far, a P2DHMM add only one parameter set, i.e., the super-state transitions, to the conventional HMM parameter sets. Therefore it is simple extension to conventional HMM.

Design of hand gesture models. Although these paper [11,12] introduced the P2-DHMMs and show promising results, a formal definition of the model as well as details of the algorithms used have been omitted. But this approach is for the first time introduced to hand gesture recognition framework. For each gesture there is a P2-DHMMs.

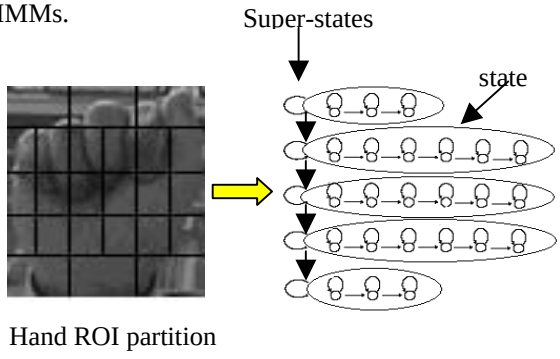


Figure 5: P2-DHMM

Fig. 5 shows a P2-DHMM consists of 5 super-states and their states in each super-state that model the sequence of rows in the image. The topology of the super-state model is a linear model, where only self transitions and transitions to the following super-states are possible. Inside the super-states there are linear one dimension hidden Markov model to model each row. The state sequence in the rows is independent of the state sequences of neighboring rows.

Improvement in our system. One major improvement in our system is the use of DCT coefficient as features instead of gray values of the pixels in the shift window where most of the image energy is found. Instead of use an overlap of 75% between adjacent sampling windows [13], we have to consider the neighboring sampling of a sampling window. Suppose we allow a deformation of up to ± d (d is a positive integer) pixels in either X or Y directions, we have to consider all the neighboring sampling within the distance d in order to detect a possible deformation. We use a shift window to improve the ability of the HMM to model the neighborhood relations between the sampling blocks.

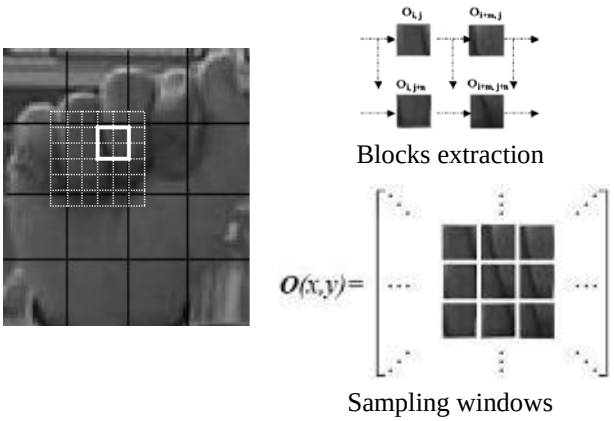


Figure 6: Sampling window

Training the hand model. Similar to neural network, P2HMMs can be trained by number of training samples. Each P2-DHMM is trained by hand gesture in the database obtained from the training set of each of the gesture using the Baum-Welch algorithm.

Intelligent selection of training images . Relatively low recognition rate of the classical P2-DHMMs for hand gesture recognition is, essentially, caused by the inappropriate selection of training images and therefore, lack of some important information. This information is required for an adequate training of model. In our proposed solution, the best images for model training are automatically selected among the available images of each subject. In the case of our hand gesture database, the problem consists of choosing 8 optimum training images out of 15 images available for each subject. Obviously, the best training set is one that contains images with different aspects of a subject hand gesture, taken in different conditions. In order to construct this training set, we used the DCT coefficients of the images, followed by the selection of images with the most distinguishable coefficients for P2-DHMMs model training. Thus, from the obtained data set, one can extract the most important information available in subject's images. The algorithm used for determining the best training images is as follows:

1. Select one training image randomly;
2. Compute the distance between the DCT vector of other images and the DCT vector of the selected image;
3. Select as the second training image, the image that has the biggest distance with the first training image;
4. For all remaining images, obtain the overall distance between each image and selected training images using equation (18);
5. Choose as the next training image, the image with the biggest distance;
6. If there is still training image, go to 4.
7. End.

By applying this algorithm, one can expect that the last image contain some information, which was not present in the previous hand images.. Equation used to calculate the distance between DCT vectors of the *i* and *j* images is as follows:

$$D_{i,j}^2 = \sum_{n=1}^{N} (d_i(n) - d_j(n))^2 \qquad (18)$$

Index of the next training image = $\text{argmax}_i(\min(D_{i,j}))$(19)

where 'N' is the length of image vector (No. of rows × No. of columns).

Gesture recognition. The Viterbi algorithm is used to determine the probability of each hand model. The image is recognized as the hand gesture, whose model has the highest production probability.

Due to the structure of the P2-DHMMs, the most likely state sequence is calculated in two stages. The first stage is to calculate the probability that rows of the individual

images have been generated by one-dimensional HMMs, that are assigned to the super-stages of the P2-DHMMs. These probabilities are used as observation probabilities of the super-states of the P2-DHMMs. Finally, on the level second Viterbi algorithm is executed. The complete double embedded Viterbi algorithm is stated explicitly in [11]. Although the P2-DHMMs recognizer chooses a model with best likelihood, but we cannot guarantee that the pattern is really similar to the reference gesture unless the likelihood value is high enough. A sample threshold for the likelihood often does not work. Therefore, we are improvement model that yields the likelihood value to be used as a threshold. A gesture is recognized only if the likelihood of the best gesture model is higher than that of the threshold model.

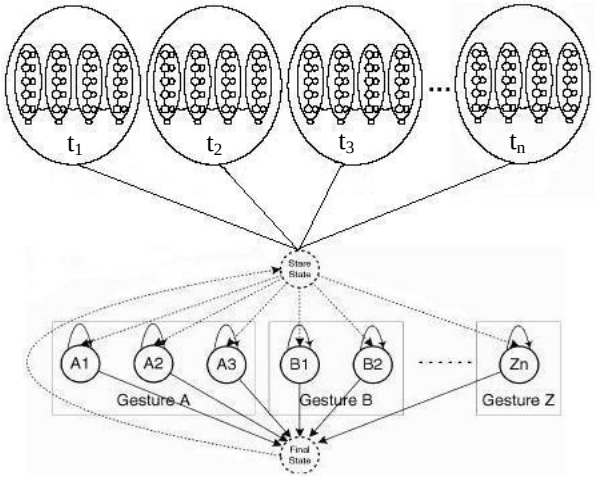


Figure 7: A simple structure of the P2-DHMMs - threshold model. The dotted arrows are null transitions.

The model transition probability into the threshold model p(TM) is to satisfy

$$P(X_G \mid \lambda_G)\, P(G) > P(X_G \mid \lambda_{TM}) P(TM) \quad (20)$$

$$P(X_{TM} \mid \lambda_G)\, P(G) < P(X_{TM} \mid \lambda_{TM}) P(TM) \quad (21)$$

where X_G, X_{TM}, λ_G, λ_{TM} denote a gesture pattern, a non-gesture pattern, the target gesture model and the threshold model, respectively.

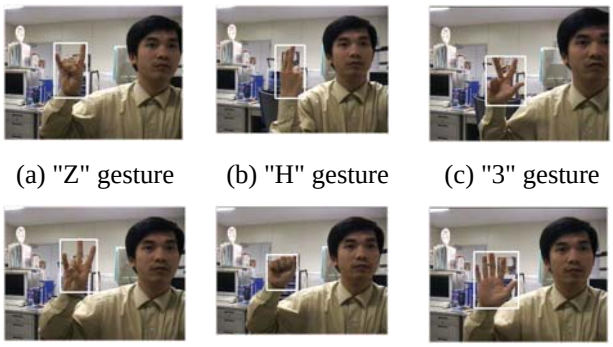
It is imply that a gesture should best match with the corresponding gesture model and that a non-gesture with the threshold model, respectively. Inequality (20) say that the likelihood of a gesture model should be greater than that of the threshold model.

5. Experimental results

In this section we present the recognition results of 36 gestures, which includes ASL letter spelling alphabet and digits (Fig. 9). We discuss results of each of the steps involved before presenting the overall results.

5.1 Results of tracking

The complete system works at about 25 frames/sec. If the hand moves too fast Kalman filter at this will not find hand pixels correctly, the tracker starts going out of track due to the translation and scale parameters get distorted values. Beside this if initialization is not good, the tracker can not archived the result of tracking and lost of track.



(a) "Z" gesture (b) "H" gesture (c) "3" gesture

(d) "Z" gesture (e)"A" gesture (f) "5" gesture

Figure 8: Some images sequence in tracking

5.2 Results of P2DHMMs based gesture recognition

We tested our recognition system using ASL show in Fig.9. As the training data for each gesture is used 30 different images of 36 gestures. The images of the same gesture were taken at different times. The best recognition rate is 98% overall result. Experiment shows that the system can work correctly with sufficient training data, to extend the training database is only a compromise to reduce the lack of training data.

Figure 9: 36 ASL used for hand testing

5.3 Comparison with other approaches

Not many vision-based approaches have been reported for real-time gesture recognition. It is P2-DHMMs in the sense that it is not a fully connected two dimensions network that would lead an algorithm running in exponential time. Compared to template based methods and 2-DHMMs, our proposed system offers a more flexible framework for gesture recognition, and can be used more efficiently in scale invariant systems. The P2-DHMMs approach is proposed and the comparison to 2-DHMMs is made. Both of them are robust enough in single gesture or size environment, but the P2-DHMMs performs much better and facilitates when recognition contain various gestures with variation of size. In other worlds, the P2-DHMMs offers promising potential to solve difficult hand gestures recognition.

6. Conclusions

We have developed a gesture recognition system that is shown to be robust for ASL gestures. The system is fully automatic and it works in real-time. It is fairly robust to background cluster. The advantage of the system lies in the ease of its use. The users do not need to wear a glove, neither is there need for a uniform background. Experiments on a single hand database have been carried out and recognition accuracy of up to 98% has been achieved. We plan to extend our system into 3D tracking. Currently, our tracking method is limited to 2D. We will therefore investigate a practice 3D hand tracking technique using multiple cameras. Focus will also be given to further improve our system and the use of a lager hand database to test system and recognition.

References

[1] V.I. Pavlovic, R. Sharma, T.S. Huang. Visual interpretation of hand gestures for human-computer interaction, A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7): 677-695, 1997.

[2] J.Davis, M.Shah. Recognizing hand gestures. In Proceedings of European Conference on Computer Vision, ECCV: 331-340, 1994.

[3] D.J.Turman, D. Zeltzer. Survey of glove-based input. IEEE Computer Graphics and Application 14:30-39, 1994.

[4] Starner, T. and Pentland. Real-Time American Sign Language Recognition from Video Using Hidden Markov Models, TR-375, MIT Media Lab, 1995.

[5] R.Kjeldsen, J.Kender. Visual hand gesture recognition for window system control, in IWAFGR: 184-188, 1995.

[6] M.Zhao, F.K.H. Quek, Xindong Wu. Recursive induction learning in hand gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11): 1174-1185, 1998.

[7] Hyeon-Kyu Lee, Jin H. Kim. An HMM- based threshold model approach for gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. 201(10): 961-973, 1999.

[8] Ho-Sub Yoon, Jung Soh, Younglae J. Bae, Hyun Seung Yang. Hand gesture recognition using combined features of location, angle, velocity, Pattern Recognition 34 : 1491-1501, 2001.

[9] R. Lockton, A. W. Fitzgibbon. Real-time gesture recognition using deterministic boosting, Preceedings of British Machine Vision Conference, 2002.

[10] K. Oka, Y. Sato and H.Koike. Real-Time Fingertip Tracking and Gesture Recognition. IEEE Computer Graphics and Applications: 64-71, 2002.

[11] O.E. Agazzi and S.S.Kuo. Pseudo two-dimensional hidden markov model for document recognition. AT&T Technical Journal, 72(5): 60-72, Oct, 1993.

[12] F. Samaria. Face recognition using hidden markov models. Ph.D thesis, Engineering Department, Cambridge University, 1994.

[13] S.Eickeler, S.Muler, G. Rogoll. High quality Face Recognition in JPEG Compressed Images. In Proc IEEE Interna. Conference on Image Processing, Kobe, 1999.

[14] Chan Wa Ng, S. Rangganath. Real-time gesture recogntition system and application. Image and Vision computing (20): 993-1007, 2002.

[15] J.Triesch, C.Malsburg, Classification of hand postures against complex backgrounds using elastic graph matching. Image and Vision Computing 20, pp. 937-943, 2002

[16] Y. Wu, T. S. Huang. View-independent Recognition of Hand Postures in Proc. IEEE Conf. on CVPR'2000, Vol II: 88-94, 2000.

[17] A.Ramamoorthy, N.Vaswani, S. Chaudhury, S. Banerjee. Recognition of Dynamic hand gestures, Pattern Recognition 36: 2069-2081, 2003.

[18]. Quek, F. Toward a Vision-based Hand Gesture Interface, Proc. of VRST : 17-31, 1994.

[19]. Freeman, W.T., Weissman, C.D. Television control by hand gestures, Proc. of 1st IWAFGR . 179-183, 1995.

[20] G.R. Bradski, S. Clara. Computer Vison Face Tracking For Use in a Perceptual User Interface. Intel Technology Journal Q2'98, 1998.

[21] E. Levin, R. Pieraccini. Dynamic planer warping for optical character recogntion. Proc. ICASSP 3, San Fransisco, CA, 149-152, 1992.

[22] R. Chellapa, S. Chatterjee. Classification of textures using Gaussian Markov random fields. IEE Trans. ASSP 33 (4), 959-963, 1985.