

# Hand Motion Analysis for Thai Alphabet Recognition using HMM

Kittasil Silanon and Nikom Suvonvorn

**Abstract**—In this paper, we propose a system for Thai alphabet recognition from hand movement trajectory, as a human-computer interaction method. Thai characters drawing by hand movement are analyzed and recognized by our system, which can apply for controlling any specific tasks. The method is based on hand motion analysis combining with Haar-like with a cascade of boost classifiers, as hand detection method. Hand is tracked with skin color using CamShift and Kalman filter. Trajectory features of hand are extracted and used for recognizing 12 Thai alphabets through the Hidden Markov Model.

**Index Terms**—Hand Movement Trajectory, Hand Gesture Recognition using HMM, Thai Alphabet Recognition.

## I. INTRODUCTION

In the field of computer vision, communication between human and computer becomes more important. Human-Computer Interaction (HCI) [1], which allows humans communicate with computer have been a popular research field for many years. Some active research in this field are human face recognition, eye gaze tracking, lip reading, hand gesture recognition and body pose tracking. In this paper, we emphasize on hand movement, which can freely moves and gesticulates more than other parts of body therefore hand movement gesture recognition can be applied in many applications such as sign language recognition, computer-controlled game, teleconference, and so on. Many researches of hand gesture recognition have been proposed, covering a wide variety of methods and approaches. For example, Wei Du et. al. [1] describes a system with one camera that can recognize four gestures and tracking hand by extracting the feature points on hand contour. Mahmoud et. al. [2] has developed a system that could recognize gesture for alphabets from hand motion using Hidden Markov Model (HMM). Juan et. al. [3] introduced a vision-based system that can interpret a user's hand gesture in real time to manipulate objects within a medical data visualization environment. Dinh et. al. [4] proposes a hand gesture classification system that able to efficiently recognize 24 basic signs of American Sign Language with Haar-like feature and AdaBoost learning algorithm.

In this paper, we proposed a system that can recognize the 12 important Thai alphabet letters selecting from the specific

group using hand movement trajectory features. The paper is organized in six sections by the following: proposed hand movement recognition system, hand detection and tracking, trajectory feature extraction, experimentation results, and conclusion respectively.

## II. PROPOSED SYSTEM

In this section, we introduce the system for hand gesture recognition based on hand motion by analyzing the image sequences, obtained from the webcam attached over the computer screen. There are three main parts of our system: hand detection and tracking, hand feature extraction and gesture recognition. The overall of system is illustrated in the figure 1. First part, a hand detector is implemented using a cascade of boost classifiers, which allows obtaining very robust object detector. However, hand pose can have many figures caused by translation and rotation in 3D, which needs to be found. The tracking method, CamShift and Kalman filter, is then applied in order to extract hand through image sequence. In the second part, the gesture trajectory features are extracted and enhanced by the Douglas–Peucker algorithm which gives us a smallest number of control points trajectory. The features are selected to give a discrete vector that use as input to discrete HMM for recognizing commands in the final part.

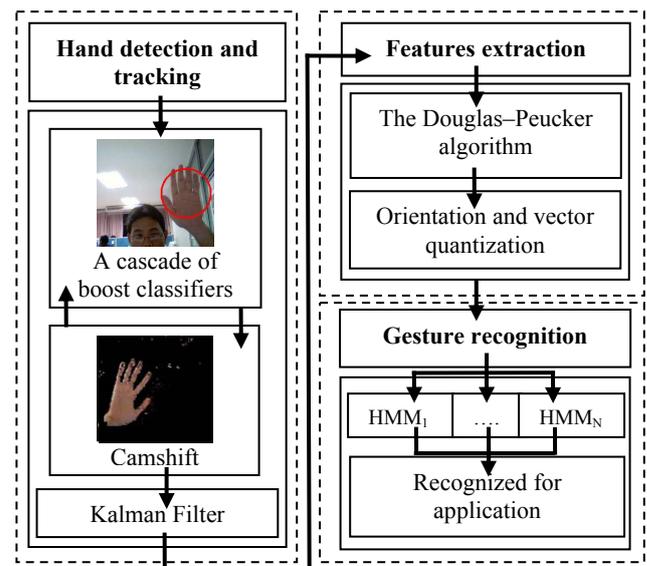


Fig.1. System overview: (left) Hand detection and tracking, (right) feature extraction and recognition process.

Figure 1 (left) shows the sequence of steps for hand detection and tracking; figure 1 (right) show the feature extraction and recognition process. These three parts will be

Manuscript received April 22, 2011; revised May 20, 2011.

Kittasil Silanon is with the Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, P.O. Box 2 Kohong, Hatyai, Songkla 90112 Thailand (e-mail: jalego3@hotmail.com).

Nikom Suvonvorn is with the Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, P.O. Box 2 Kohong, Hatyai, Songkla 90112 Thailand (e-mail: kom@coe.psu.ac.th).

detailed in the section 3 and 4 respectively. In section 5, the experimentation result is discussed for evaluating the performance of system.

### III. HAND DETECTION AND TRACKING

#### A. Hand Detection

A Cascade of Boost Classifier [5] [6] for object detection is originally developed by Viola and Jones. This method uses Haar-like features and a cascade of boosted tree classifier as a supervised statistical model of object recognition. Haar-like feature consists of two or three connected “black” and “white” rectangles. The feature value is defined by the difference between the sums of pixel values within the black and white rectangles. Figure 2 shows a basic set of Haar-like features.

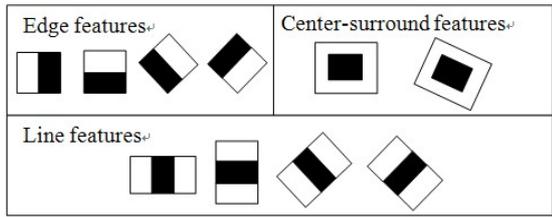


Fig.2. A set of Haar-like feature.

The AdaBoost algorithm is introduced to improve the classification performance that designed to select rectangle feature which best separates the positive and negative example. In the first iteration, the algorithm train a weak classifier  $h(x)$  using one Haar-like feature that achieves the best recognition performance for the training samples. The classifier consist of a feature  $f(x)$ , a threshold  $\theta_t$ , and parity  $p_t$ , indicating the direction of inequality sign.

$$h_t(x) = \begin{cases} 1 & \text{if } p_t f_t(x) < p_t \theta_t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In the second iteration, the training samples that were misclassified by the first weak classifier receive higher weights. The iteration goes on and the final result is a cascade of linear combinations of the selected weak classifiers  $H(x)$ , which achieves the required accuracy.

$$H(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (2)$$

In the detection process by using the trained cascade, the sub-windows must be test each stage of the cascade. A negative outcome at any point leads to the immediate rejection of the sub-window



Fig.3. Hand detection: (a) Input image (b) Detected hand.

#### B. Hand Tracking

CamShift [7] is a non-parametric technique using for color object tracking deriving from the Mean Shift algorithm. The main difference between CamShift and Mean Shift algorithm is that CamShift updates continuously its probability distributions; in generally the target object in image sequences changes significantly its shape size or color, while Mean Shift is based on static distributions. That why CamShift is suitable for tracking the rigid object. In the hand tracking, the process can be described as the following.

Step 1: the color probability distribution of detected hand image is determined from its histogram via hue component of HSV color space, related to skin color.

Step 2: this target distribution of detected hand is tracked on the searching window of next frame in image using mean shift algorithm. The mean shift vector, which is aimed for finding an optimized path that climbs the gradient of a probability distribution to the mode (peak) of nearest dominant peak, is necessary to be computed.

Step 3: the back-projection technique, which associates the pixel values in the image (tracking hand) with the value of the corresponding distribution, is applied.

Step 4: the center of mass of tracking hand is computed and defined as hand trajectory. This step will be detailed in the next section (trajectory parameters).

Step 5: on the next iterative, the current position of hand in image is used for defining the searching window on the next frame. The process is repeated at step 2 continuously.

Note that the step 1 will be re-executed systematically if the detected hand by Haar-like features with boost cascade of classifier found in the searching window. We found that Haar-like method provides very accurate results when hand is paralleled to the vertical axis, compared with CamShift, but missed mostly in other directions, so that CamShift is applied in order to solve the problem.



Fig.4. Hand tracking with Camshift.

#### C. Hand Movement Estimation

The Kalman filter [8] is a recursive linear filtering method. It addresses the general problem of trying to estimate the state of discrete time process that is described by the linear stochastic differential equation by the following.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (3)$$

with a measurement  $z \in \mathcal{R}^m$  that is

$$z_k = Hx_k + v_k \quad (4)$$

The random variables  $w_k$  and  $v_k$  represent Gaussian noise of the process and measurement respectively. The algorithm of Kalman filter estimates a process by using feedback control technique: estimating the process state by an appropriate model and doing feedback by noisy

measurements. As such, the equations of Kalman filter are formed into two groups: prediction and correction equations. In the post tracking of hand, the algorithm can be described as the following figure 5.

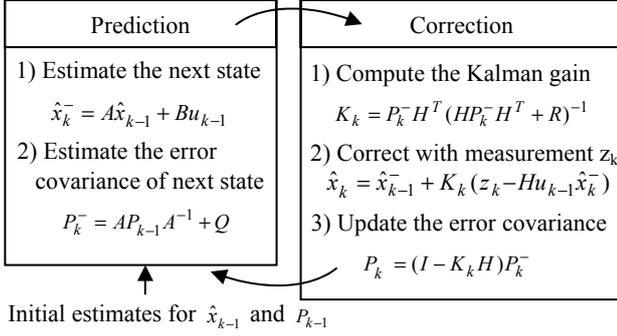


Fig.5.The operation of Kalman filter.

Experimentally, we found that hand detector by Haar-like features with boost cascade of classifier cannot provide total results and CamShift may wrongly track especially when there are other parts of body such as face or background, having color in skin color range, move close to tracking hand. Therefore, the Kalman filter is applied in order to predict hand position in the next frame based on previous frame, obtained by CamShift. To apply the Kalman algorithm, two principle equations needed to be declared: tracking process and measurement. The state of tracking process is measured from hand position and velocity in each image frame. So, we define the process of state  $x_k$  by the following:

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k,i} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k-1,i} + w_{k-1,i} \quad (5)$$

where  $x$ ,  $y$ ,  $v_x$ ,  $v_y$  are the position and the velocity of hand in the  $k^{\text{th}}$  image frame respectively. Here, we assume that the motion of hand between two successive frames can be uniformly approximated as straight line, the frame interval  $\Delta t$  is very short. For the measurement  $z_k$ , we define directly by the position value obtained from CamShift algorithm.

#### IV. FEATURE EXTRACTION

##### A. Trajectory Parameters

In order to form the trajectory of hand position from the step 4 of hand tracking process by CamShift algorithm and Kalman filter. We choose to use the center point of hand and the trajectory can be obtained by joining this point in every frame in the sequence, in each frame, we obtained the center point of hand's region that can be easily computed from the moments of pixels in hand's region, which is defined as :

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \quad (6)$$

In the above equations,  $I(x,y)$  is the pixel value at the position  $(x,y)$  of the image,  $x$  and  $y$  are range over the hand's region. The center point of hand  $(X_c, Y_c)$  is calculated as :

$$X_c = \frac{M_{10}}{M_{00}}, \quad Y_c = \frac{M_{01}}{M_{00}} \quad (7)$$

##### B. Hand Tracking

Since during gesturing the hand does not move very fast, the position of hand does not change much from one frame to the next one. Therefore, for trajectory formation it may not be necessary to consider all the frames in a gesture sequence. Accordingly, we propose to select key trajectory point to be stored in the new sequence. This reduces the memory requirement as well as speeds up the trajectory matching during recognition process. The algorithm that we use to extract key trajectory point is the Douglas–Peucker algorithm [9]. The algorithm recursively divides the line. Initially it is given all the points between the first and last point (figure 6 (a)). It marks the first and last point. It then finds the point that is furthest from the line segment with the first and last points as end points (this point is obviously furthest on the curve from the approximating line segment between the end points). If the point is closer than threshold to the line segment then any points not currently marked to keep can be discarded without the smoothed curve being worse than threshold. If the point furthest from the line segment is greater than threshold from the approximation then that point must be kept. The algorithm recursively calls itself with the first point and the selected point and then with the selected point and the last point (which includes marking the selected point being marked as kept). When the recursion is completed a new output curve can be generated.

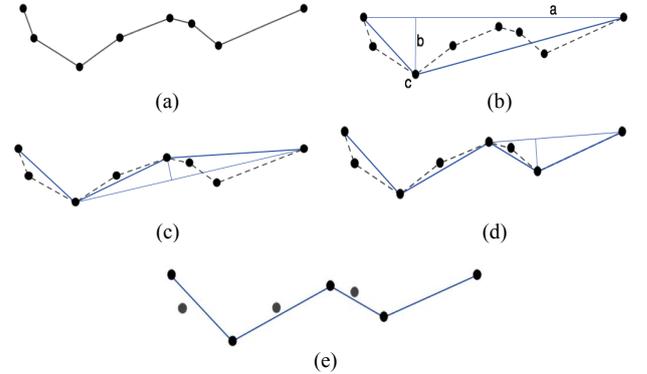


Fig.6.Smoothing a piecewise linear curve with the Douglas-Peucker algorithm.

##### C. Features

After getting the key trajectory points, we calculated the orientation between consecutive points to obtain a sequence of angle.

$$\theta_t = \arctan\left(\frac{Y_{t+1} - Y_t}{X_{t+1} - X_t}\right); \quad t = 1, 2, \dots, T-1 \quad (8)$$

where  $T$  represents the length of gesture trajectory. Figure 7 shows the angle's domain is  $[0, 360]$  degrees. We divide this angle by  $20^\circ$  to quantize them to 18 directional codeword from 1 to 18. The codeword is used as input to HMM recognition model. Figure 8 show the result of gesture that was approximate by the Douglas–Peucker algorithm and define a gesture as a sequence of directional codewords which are  $O = \{13, 5, 10, 5, 3, 1, 16, 16, 14\}$ .

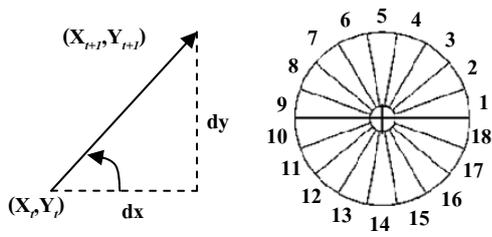


Fig.7.Orientation codewords.

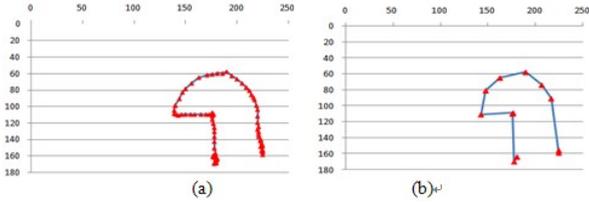


Fig.8.(a) Gesture curve (b) Gesture curve after approximate by the Douglas-Peucker algorithm

## V. RECOGNITION MODEL

### A. Hidden Markov Model

In the recognition process, the probabilistic approach, such as the Hidden Markov model [10], is applied for characterizing gestures representing the Thai alphabet letters. The HMM model can be defined by the following:

- 1). The set of states  $S = \{s_1, s_2, \dots, s_N\}$  where  $N$  is number of states.
- 2). An initial probability for each state  $\pi_i$ ,  $i = 1, 2, \dots, N$  such that  $\pi_i = P(s_i)$  at the initial step.
- 3). An  $N$ -by- $N$  transition probability matrix,  $A = \{a_{ij}\}$ , where  $a_{ij}$  is the transition probability of taking the transition from state  $i$  to state  $j$ .
- 4). The set of observation symbols  $O = \{o_1, o_2, \dots, o_M\}$  representing our 18 directional codewords,  $M$  is the number of observation symbols.
- 5). An  $N$ -by- $M$  observation matrix,  $B = \{b_j(o_k)\}$  where  $b_j(o_k)$  give the probability of emitting observation symbol  $o_k$  from state  $j$ .

An HMM requires specification of two model parameters ( $N$  and  $M$ ), specification of observation symbols, and the specification of the three probability measures:  $A$ ,  $B$ , and  $\pi$ . For convenience, we use the compact notation to indicate the complete parameter set of model.

$$\lambda = (A, B, \pi) \quad (9)$$

There are three basic problems for HMM. These problems are the following:

Evaluation problem: Given the observation sequence  $O = O_1O_2\dots O_T$ , and model  $\lambda = (A, B, \pi)$ , calculate the probability that model  $\lambda$  has generated sequence  $O$ .

Decoding problem: Given the observation sequence  $O = O_1O_2\dots O_T$ , and the model  $\lambda$ , calculate the most likely sequence of hidden states  $s_i$  that produced this observation sequence  $O$ .

Learning problem: How do we adjust the model parameters  $\lambda = (A, B, \pi)$  to maximize  $P(O|\lambda)$ .

The solutions to these three problems are

Forward-Backward algorithm, the Viterbi algorithm, and the Baum-Welch algorithm respectively.

## VI. EXPERIMENTATION RESULT

### A. Hand Detection

In our system, a command begins with open hand posture and stop using stopping hand movement. We have collected around 5,000 positive samples for each hand style from students in our university in various conditions such as: indoor with neon light and outdoor with natural light. Systematically, around 10,000 negative samples are selected from landscape, building, and human faces or body images. Figure 9 shows some of these samples.



Fig.9.Training samples: (a) Positive samples. (b) Negative samples.

We have trained 4 hand detectors with various states as show in table 1. By observing the true positive rate and false positive rate, the hand detector with more state reduces the false positive rate and also decreases the true positive rate. In order to evaluate the performance of these hand detectors, we applied receiver operating characteristic (ROC) curve to calculate the area under curve (AUC) of each hand detector, its value will always be between 0 and 1.0. If the AUC is close to 1, represents a very good performance. If the AUC is close to 0.5, represents a worse performance. We notice that the hand detector with 19 states (AUC = 0.94) is quite good, close to the value of 1.0 and much larger than other one. Therefore, we choose to use hand detector with 19 states for our system. A ROC curve showing the performance of our detector is depicted in figure 10. Hand detector result of the 17 and 19 states is displayed in the figure 11.

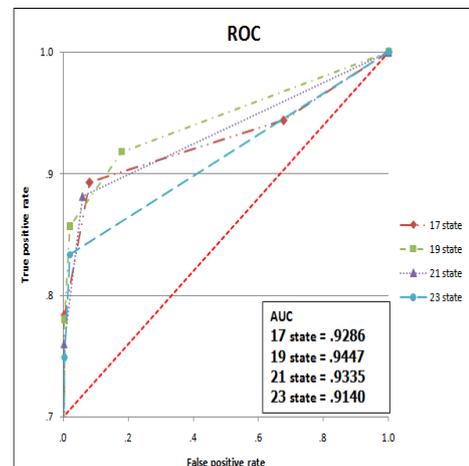


Fig.10.Hand detector ROC curve

TABLE1. HAND DETECTION RATE.

Rate	Hand detector (states)			
	17	19	21	23
True positive	94.3%	91.7%	88.1%	83.3%
False positive	67.7%	17.8%	5.8%	1.9%

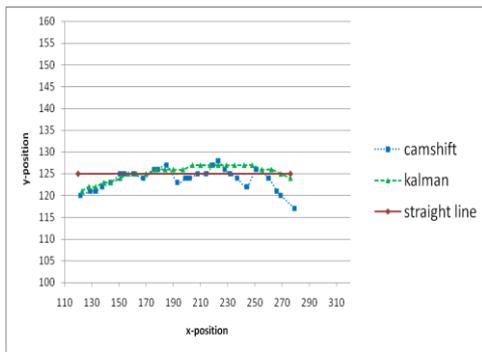


Fig.11. Hand detector result: (a) 17 states (b) 19 states

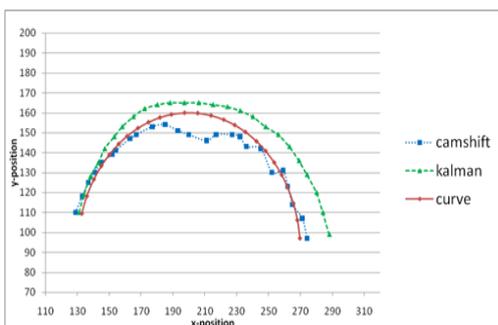
At this stage, the best performance of hand detection is at 91.7 % with 17.8% of false positives. The errors from this stage will be corrected with Kalman tracking method in the following section.

**B. Hand Movement Estimation**

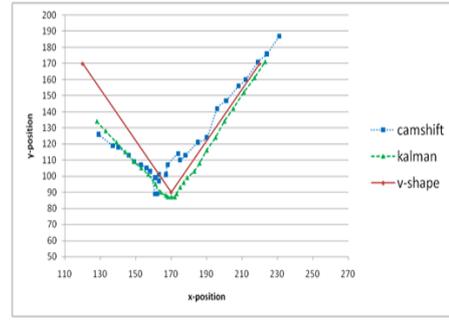
In this section, we evaluate the precision of trajectory obtained from Camshift and Kalman filters. During the correction phase of Kalman filter, the new position was estimated using detected position from Camshift and predicted position of prediction phase. The test is done by executing the hand movement following three types of ideal path: straight line, curve and v-shape respective, considered as basic components of the more complex movement. We compute the errors between ideal path and tracked position by using the sum of absolute difference. Figure 12 show graphs presenting the trajectories of hand motion obtained from camshift and kalman methods compared with the ideal path. We found that globally the kalman method gives the trajectories smoother than camshift, which is more appropriate for the considering application.



(a)



(b)



(c)

Fig.12. Hand motion estimation: (a) Line (b) Curve (c) V-shape

Table 2 shows the precision of trajectory using our hand estimation method. The test is executed ten times. We found that the error in average of curve (6.24 pixels) is bigger than v-shape (5.53 pixels) and line (1.61 pixels) respectively. This can be explained from the behavior of transition matrix that is estimated as linear motion. However, if we consider the distribution of positions, our method give quit good result; error is not more than two pixels.

TABLE2. HAND TRACKING ERRORS.

No.	Testing errors (pixels)		
	Line	Curve	V-shape
1	1.54	3.79	8.05
2	1.70	6.61	3.31
3	1.39	6.87	4.22
4	1.59	6.68	9.13
5	1.97	7.55	4.98
6	0.97	8.93	3.51
7	1.40	8.09	4.31
8	2.20	4.23	4.44
9	1.76	6.05	7.05
10	1.84	6.97	6.38
$\mu$	1.61	6.27	5.53
$\sigma$	0.29	1.74	1.89

We found that the final trajectories obtained from this stage can firstly eliminate the false positives produced during the detection process. Secondly, it filled up some loss position or false negatives by reliable values from our prediction and estimation procedure .

**C. Thai Alphabet Recognition**

For the performance evaluation of our recognition process, we select the 12 Thai alphabets from the 12 groups of 44 Thai alphabets [11] by considering from its similarities, as shown in table 3. We notice that each selected Thai alphabet can be decomposed into the three basic components: line, curve, and v-shape.

TABLE 3. THE 12 GROUPS OF THAI ALPHABET.

Group 1	ก ญ ฎ ฏ ถ ภ	Group 7	ผ ผฝ พ ฟฝ
Group 2	ช ฌ ฎ ฐ	Group 8	ฮ ฮี
Group 3	ค ฅ ฆ ง	Group 9	ฐ ฑ ฒ
Group 4	ฌ ฎ	Group 10	ล ลี
Group 5	ท ฑ ฒ	Group 11	ณ ฎ ฏ
Group 6	บ ฅ ษ	Group 12	ณ ฒ ฎ ฏ ฐ ฑ

We execute 150 sequences of hand motion for each

alphabet, 1800 in totals. The hand motion trajectory obtained from the 12 Thai alphabets using the method presenting last section is approximated by the Douglas-Peucker algorithm. The appropriate trajectories for recognition process of Thai alphabets are shown in the figure 13.

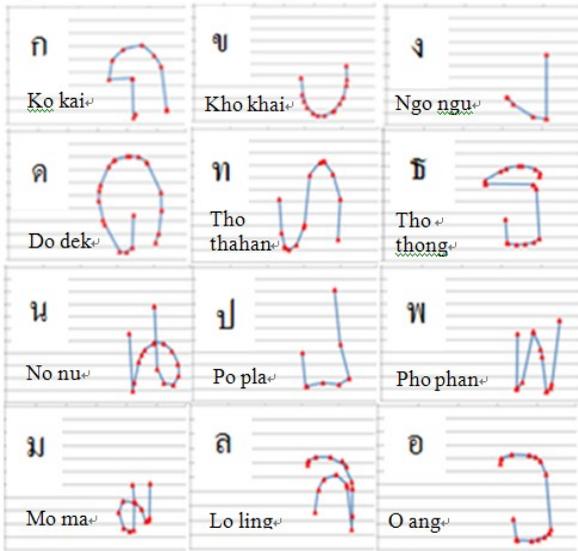


Fig.13.Hand trajectory of the 12 Thai alphabet.

In our recognition system, the fully connected HMM topology is applied. The number of state that we take in our consideration is depending on the possible number of segment parts of trajectory. The principle directions of segment part are consists of 8 directions (up, down, left, right, up-right, up-left, down-right and down-left). Intuitively, we use 8 hidden states with 2 auxiliary states as initialization and finalization.

However, it is better that this number need to be tested experimentally. Therefore, we evaluate the number of HMM state with 10 and its variations: 5 and 15 states. The table 4 shows the result from a selective recognition process. We can notice that the HMM with 10 states improve significantly the hit rate comparing to HMM with 5 state (9%). Although the HMM with 15 states give in global better result than HMM with 10 states but the improvement of correction rate is very insignificant (1%). In conclusion, the HMM with 10 states is good enough for our Thai alphabet recognition system.

TABLE4. RECOGNITION RATE FOR THE 12 THAI ALPHABETS.

Gesture model	Correct recognition		
	5 state	10 state	15 state
ก	96	82	86
ข	99	90	87
ง	92	95	86
ด	95	98	96
ท	24	65	68
ธ	21	69	59
น	96	93	97
ป	77	75	75
ฟ	58	70	72
ม	68	69	95
ล	76	95	98
อ	91	97	91
$\mu$	74.41	83.16	84.16
$\sigma$	27.40	12.81	12.79

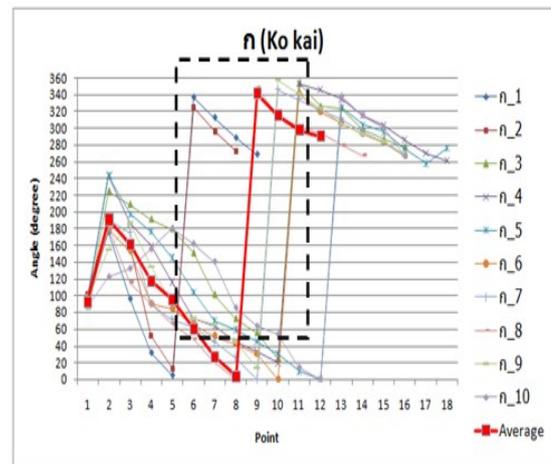
By using the HMM with 10 state, we study more deeply on

the 12 Thai alphabet recognition. We use 50 observation sequences for training and 100 observation sequences for testing. The table 4 shows the recognition rate produced by HMM with 10 states.

TABLE5.HMM WITH 10 STATE RECOGNITION RATES.

Gesture	HMM with 10 state	
	Correctness	Incorrectness
ก	82	ด = 16, น = 1, ม = 1
ข	90	ท = 9, ธ = 1
ง	95	ด = 2, ธ = 3
ด	98	ก = 1, น = 1
ท	65	ก = 3, ข = 6, ด = 6, น = 4, พ = 8, ม = 2, ล = 3, ธ = 3
ธ	69	ข = 1, ป = 2, ธ = 28
น	93	ด = 1, ท = 3, พ = 1, ธ = 2
ป	75	ข = 9, ธ = 1, ธ = 15
ฟ	70	ข = 11, ท = 14, น = 1, ป = 3, ล = 1
ม	69	ก = 12, ข = 2, ท = 11, น = 5, ธ = 1
ล	95	น = 2, ม = 1, ธ = 2
อ	97	ธ = 2, ป = 1

We found that our recognition system can give globally a good recognition rate at 84% in average. Some alphabets, such as ด (Do dek), ล (Lo ling) and อ (O ang), hit a very high rate more than 95%. Only some alphabets like ท (Tho thahan) have quit low rate. Technically, HMM try to choose the gesture model which has the best match to the input sequence. But it may still have errors in matching, such as ก (Ko kai) gesture is recognized as ด (Do dek), น (No nu) or ม (Mo ma) gesture, as shown in the last column of table 5. We realize that hand gestures considering only angles of segment part may leads to a very confusing decision situation, as shown in figure 14. The changing of principle angle in trajectories will dominate the small angle variations, specifying the different alphabets, during the recognition which produce the major recognition error. In figure 14, we noticed that the angles, draw in rectangle, of the 4 different hand gestures are similar. Intuitively, the probability of testing sequence (in the rectangle) may match to the other gesture model more than its own.



(a)

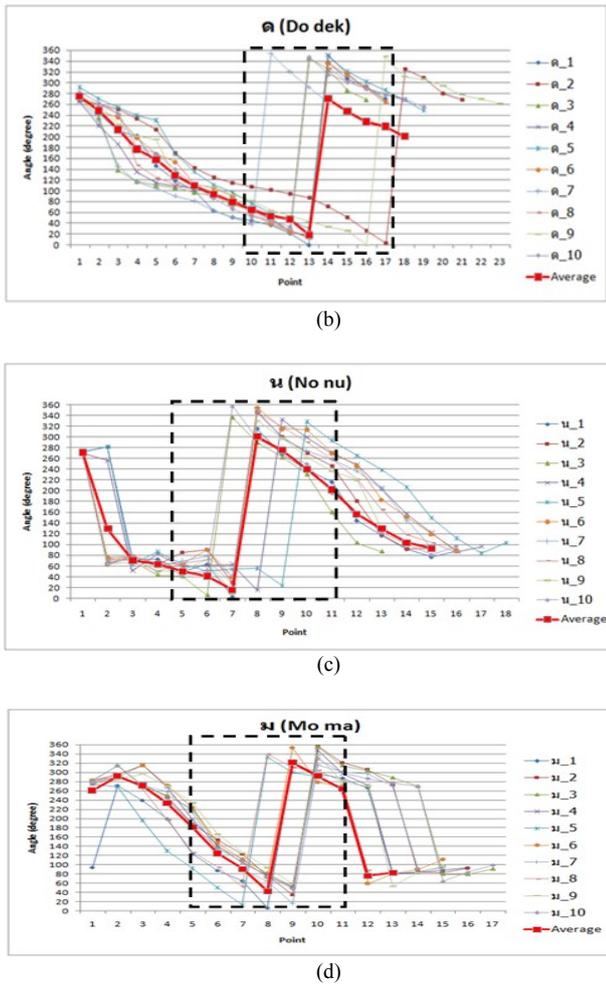


Fig.14. Angle sequence: (a) น (Ko kai) (b) น (Do dek) (c) ๓ (No nu) (d) ๓ (Mo ma)

Experimentally, we found that our proposed system is quite powerful, which the recognition processing can be done globally in very high rate at real time constraints. The demonstration of our system in real-time is shown in the video at <http://www.youtube.com/watch?v=jEZi02EuweY>.

## VII. CONCLUSIONS

We have introduced the recognition system for the Thai alphabets using hand motion trajectory with Hidden Markov Model. The reliable features are extracted using mixture of appropriate methods. The HMM topology with well parameterized parameters were applied in our system. The result shows that the system can recognize in average about 84 % of recognition rate in real time.

## REFERENCES

- [1] Wei Du and Hua Li , “Vision based gesture recognition system with single camera”, in 5th *International Conference on Signal Processing*, pp.1351-1357 vol.2, 2000
- [2] Elmezain M., Al-Hamadi, “Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models,” *Signal Processing and Information Technology, 2007 IEEE International Symposium*, pp.1192-1197, 15-18 Dec. 2007
- [3] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler, “A Real-Time Hand Gesture System Based on Evolutionary Search,” Vol. 22, no. 3, Oct. 2006, Dearborn, Mich, *Society of Manufacturing Engineers*.
- [4] Qing Chen, Georganas, N.D., Petriu, E.M., “Real-time Vision-based Hand Gesture Recognition Using Haar-like Features,” *Instrumentation and Measurement Technology Conference Proceedings, 2007. IEEE*, pp.1-6, 1-3 May 2007.
- [5] Viola, P. and Jones M., “Rapid object detection using a boosted cascade of simple features,” *Proceedings of the IEEE Computer Vision and Pattern Recognition*, vol.1, pp. I-511- I-518 vol.1, 2001
- [6] Lienhart R. and Maydt J., “An extended set of Haar-like features for rapid object detection,” *International Conference on Image Processing*, vol.1, pp. I-900- I-903 vol.1, 2002.
- [7] G. R. Bradski, “Computer vision face tracking for use in a perceptual user interface,” *Intel Technology Journal*, Q2 1998.
- [8] Greg Welch and Gary Bishop, “An Introduction to the Kalman Filter,” Annual Conference on Computer Graphics & Interactive Techniques. ACM Press, Addison-Wesley, Los Angeles, CA, USA (August 12–17), SIGGRAPH 2001 course pack edition.
- [9] Ramer-Douglas-Peucker algorithm, july 2010, Available: [http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker\\_algorithm](http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm)
- [10] L.R. Rabiner, “A tutorial on hidden markov models and selected application in speech recognition,” *Proc. IEEE* 77, 1989, pp 267-293
- [11] Thai script, October 2010, Available: [http://en.wikipedia.org/wiki/Thai\\_script](http://en.wikipedia.org/wiki/Thai_script)



**Kittasil Silanon** was born in Narathiwat, Thailand, on April 14, 1986. He received the B.Eng (Computer Engineering) in 2008, from Prince of Songkla University (PSU), Hatyai, Songkla, Thailand.

He is currently pursuing M.Eng (Computer Engineering) at PSU in computer vision. His research interests are in the areas of image processing and computer vision.



**Nikom Suvonvorn** was born in Trang, Thailand, on November 26, 1976. In 2006, He received a PhD in computer science from l’Université de Paris Sud (XI), Orsay, France. In 2003, He obtained a DEA (Diplôme d’Etudes Approfondies), on Electronic System and Information Processing (SETI) from l’Institute d’Electronique Fondamentale (IEF) at the same university. In that year, He also got another master’s degree on computer engineering from Ecole Supérieure de Mécanique et d’Electricité (ESME)-Sudria engineering school, Paris.

He is currently a lecturer and research scientist at Department of Computer Engineering (CoE), Faculty of Engineering (ENG), Prince of Songkla University (PSU), Hatyai, THAILAND. His research corresponds to computer vision, image processing, and its related applications. The actual research is emphasized on the OpenVSS project, the next generation multimedia technologies applied for the Surveillance & Smart Environment System.