Thai Alphabets Finger-Spelling Recognition System

Kittasil Silanon

A Thesis Submitted in Fulfillment of the Requirements for the Degree of

Doctor of Philosophy of Engineering in Computer Engineering

Prince of Songkla University

2016

Thesis Title Thai Alphabets Finger-Spelling Recognition System

Author Mr. Kittasil Silanon

Major Program Computer Engineering

Major Advisor :

……………………………..........................

(Asst. Prof. Dr. Nikom Suvonvorn)

Examining Committee :

………………………………………...Chairperson

(Assoc. Prof. Dr. Montri Karnjanadecha)

………………………...

(Asst. Prof. Dr. Nikom Suvonvorn)

…………………………………………...................

(Dr. Anant Choksuriwong)

…………………………………………...................

(Dr. Keerati Inthavisas)

The Graduate School, Prince of Sonkla University, has approved this thesis as fulfillment of the requirement for the Doctor of Philosophy Degree in Computer Engineering.

……………………………..….…..……………………

(Assoc. Prof. Dr. Teerapol Srichana)

Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate's own investigations. Due acknowledgement has been made of any assistance received.

……………………………………………Signature

(Asst. Prof. Dr. Nikom Suvonvorn)

Major Advisor

…………………………………………… Signature

(Mr.Kittasil Silanon)

Candidate

I hereby certify that this work has not been accepted in substance for any degree, and is not being currently submitted in candidature for any degree.

…………………………………….……Signature

(Mr.Kittasil Silanon)

Candidate

Thesis Title Thai Alphabets Finger-Spelling Recognition System

Author Mr.Kittasil Silanon

Major Program Computer Engineering

Academic Year 2015

ABSTRACT

This thesis proposes techniques for Thai alphabets finger-spelling recognition system based on hand posture estimation. The system is established into the following steps. Firstly, Initial hand segmentation: hand is detected to obtain region of interest and calculate initial hand feature such as finger tips, palm size, hand center and other necessary features. Secondly, Key hand posture selection: fingertip is tracked using concept of active contour which combine the energy of continuity, curvature, direction, depth and distance to track the fingertip in hand posture sequence. The difference of fingertip positions in consecutive frame is computed to remove unimportant frame and find stable key hand posture, the difference fall under a predefined threshold, during in sequence. Thirdly, hand posture estimation: representing hand posture, the four basic finger shapes have been defined (“Open”, “Bend”, “Point” and “Close”). Moreover, hand appearance which have finger relations (“Group”, “Separate” and “Cross”), hand movement and hand rotation will be used in this step as well. Finally, recognition model: this process will be divided into two steps. The first recognition step, recognize single key hand posture. The recognition of this step is to recognize the American finger-spelling from each key posture which is extracted in finger-spelling sequence. The hand features (finger shape and hand appearance) in each key hand posture will be encoded as a feature array chain code. The code will be compared with template feature array to recognize American finger-spelling. The second recognition step, due to Thai alphabets finger-spelling derives from combination of American finger-spelling hand posture. Therefore, this step is to recognize the combination of American finger-spelling hand posture to represent Thai alphabet finger-spelling using learning-based approach such as Hidden Markov Model (HMM). The experimentation result of Thai alphabets finger-spelling recognition is 68% in average. However, the recognition rate will depend on expert and data training that used to train the system.

ACKNOWLEDGMENTS

First of all, I would like to express my guidance sincere gratitude to my advisor, Asst. Prof. Dr. Nikom Suvonvorn, for his attention, insight, encourage, guided, and support during this research and for being available at any time to response my question, which has been valuable.

I would like to thank Assoc. Prof. Dr. Montri Karnjanadecha and Dr. Anant Choksuriwong for all assistance, helpful comments, and insightful suggestions.

I am grateful to all instructors and staffs for their kind help and support during my thesis Ph.D. program.

I especially acknowledge the Nation Research University Project of Thailand's Office of the Higher Education Commission for financial support.

Finally, my greatest thanks are to my beloved family and parents whose caring, understand, and possible attitude have encouraged me to go forward during difficult times. Whose never-ending love and support made the completion of this work completed and my dream of a graduate education come true.

Kittasil Silanon

CONTENTS

Page

CONTENTS

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF FIGURES

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER 1

INTRODUCTION

1.1 Motivation

The common method of communication for ordinary people is spoken language. However, there are some groups of people that cannot use spoken language to communicate. These are the hearing and speaking impaired persons who cannot hear or speak such as ordinary people. Therefore, they have used sign language. Sign language is a language which transmits visual sign patterns to express a speaker's meaning. A sign language system can be separated into 2 categories, as follows:

1). word-level vocabulary signs, those are the signing of hand shapes, direction and movement of the hands, arms or body, and facial expressions simultaneously to express meanings which are used in the majority of communication.

2). finger-spellings, which are using finger postures to alphabetically word in spoken language, for communicating names, places, technical words, or anything else which is not in sign language, and can be used as a source of new word.

From research, we found that hearing and speaking impaired persons who use sign language have problems with finger-spelling skill because they use word-level vocabulary signs for communicating each other. In contrast, finger-spelling has been used for 7% to 10% of communication with sign language in daily life. Most of them thought finger-spelling word recognition was the toughest part of learning sign language, such as “Finger-spelling is the first skill learned and the last skill mastered”. Thus, acquisition of finger-spelling skill typically lags far behind other sign language skills. From these reasons, most of the hearing and speaking impaired persons are inexperienced in finger-spelling skill, which is a cause of communication mistakes.

The problem of inexperienced in finger-spelling skill and learning system requires sign language expert for monitoring. This may be insufficient for classroom learners. Therefore, computer technology has been used to solve the problems. Many researchers developed a finger-spelling recognition system that can be applied to resources for self-study material. There are several methods which have been proposed for finger-spelling in each sign language such as

American (ASL) [18,24,26], British (BSL) [19], Australian (Auslan) [16], Chinese (CSL) [32,33], Japanese (JLS) [10,35], Thai (ThLS) [3,9,23,25,29,30,24] and others [2,6,17,20], which aim to improve a finger-spelling skill for hearing and speaking impairment persons that can improve skill, and decrease the error in finger-spelling communication.

1.2 Relate Works

Various methods have been proposed in finger-spelling recognition, but most of them have serious problems for making computer understand hand posture characteristic. Therefore, most work that relates to finger-spelling will use the hand posture estimation method [1, 7, 21, 28] as its main concept for development system. The taxonomy of hand posture estimation can be separated into the gloves-based and vision-based methods that are shown in Figure 1.1.

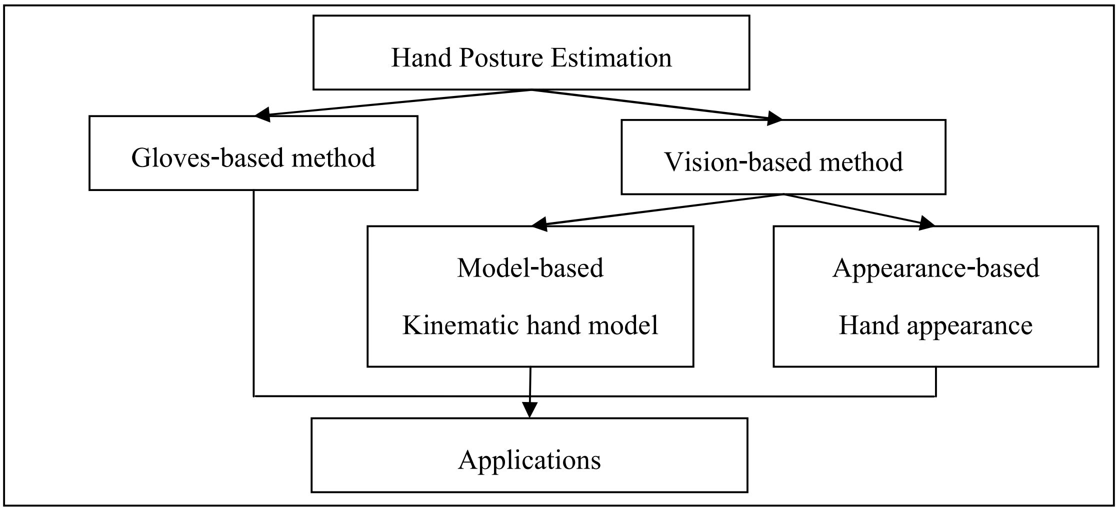


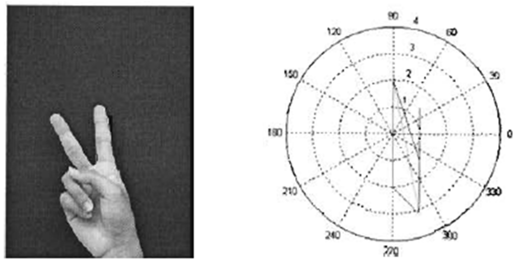
Figure 1.1 Hand posture estimation taxonomy.

The gloves-based methods use an electronic sensor device for hand and finger motion digitizing into multi-parametric data. All the gloves-based methods are designed to easily detect the hand shape, and finger motion, in real time. However, the mechanical gloves are expensive, troublesome to put on and take off and finger movement is limited. Therefore, the vision-based methods, consisting of model-based and appearance-based methods, are proposed to estimate hand posture, which provides a more suitable and natural model for human-computer interaction (HCI). The model-based method uses a kinematic hand model to estimate the articulated hand parameter (i.e. joint angle, hand position) leading to a full reconstruction of articulated hand motion with about 27 degrees of freedom (DOF). Generally, the approach is to try to estimate the kinematic hand articulated parameters. Therefore, the color marker, color glove, or depth information is used to extract fingertip and joint locations or some anchor points on the hand. The model-based method offers a rich description that potentially allows a realistic hand posture. However, the hand is an articulated deformable object involving degrees of freedom, which may cause higher computational complexity of the mapping between external position measurements and internal functional hand kinematic parameters to estimate hand posture. The appearance-based method uses computer vision techniques to extract 2-D image features such as point, edge, contour, or silhouette to model the visual appearance of the hand posture. The appearance-based method has the advantage that partial occlusions of an object can be handled easily, as well as considerable deformations or changes in viewpoint. However, the hand appearance feature may not provide enough information to estimate hand posture if there is some loss of feature information or noise in the images. In addition, the hand posture estimation can be applied in many complex and interesting applications such as virtual reality, robotics or games as well.

Several methods in finger-spelling recognition have been proposed. However, our work focuses only on Thai finger-spelling. For example, Supawadee [25] proposed a Thai Finger-Spelling Sign Language Recognition System (TFRS) which uses a data glove and motion tracker as shown in Figure 1.2. The data glove provides a signal of flexures and abductions of all fingers, and the motion tracker provides hand movement in Cartesian coordinates system including yaw, pitch and roll. The glove sensors can capture all the finger features more precisely. The key frames which represent hand postures of signing words are extracted and then recognized by the Elman Back Propagation Neural Network (ENN) algorithm.

Figure 1.2 The data glove and motion tracker sensor

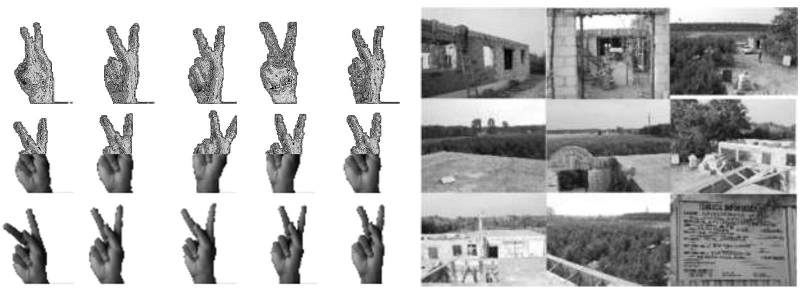
Wirote [29] presents recognition of image sign language for Thai consonants using artificial neural networks (ANNs). The 15 single-shot hand postures are shown in Figure 1.3.(a). Each image is preprocessed with resizing, noise reduction, edge detection and feature selection. The image feature is extracted from a polar orientation histogram as show in Figure 1.3.(b). The preprocessed images are trained and tested using ANNs by n-fold cross validation technique



(a) (b)

Figure 1.3 Thai alphabet hand posture: (a) original image (b) image feature extraction using polar orientation histogram

Jirapong [9] presents Thai alphabet finger-spelling which uses object detection method, Cascade of Boosted Classifiers, to train 31 learning models for each alphabet recognition. The system needs to collect positive sample sets, image hand posture for recognition, and negative sample sets, other images such as background or another object. Examples of positive and negative sets are shown in Figure 1.4.



(a) (b)

Figure 1.4 Data sets: (a) positive samples (b) negative samples

Weerachai [30] introduced new Thai sign language recognition method that used a vision-based technique with a simple color hand glove as show in Figure 1.5. The features are represented by hand appearance based model features such as centriod, axis, edge, and height/width ratio. The hand pattern recognition is established by using a C5.0 algorithm which is adapted from decision tree theory. Some letters and numbers for Thai Finger-Spelling are recognized.

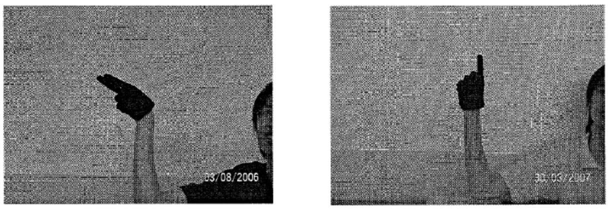


Figure 1.5 The simple color glove

Sirboonruang [34] proposed a method to combine Zernike moment and wavelet transformation for small changes in rotation and subtle differences of hand posture classification. The Zernike moment is used to capture global features and wavelet moment to differentiate between subtle variations. Fuzzy classification algorithm is used to classify hand posture for Thai finger-spelling recognition. Suwanee [23] developed a Thai finger-spelling sign language translation system, with 15 and 10 words. The scale invariant feature transform (SIFT) is applied to compute key point descriptors of hand posture, as in Figure 1.6. The recognition process compares the Euclidean distances between two nearest neighbor key point descriptors in the hand posture database to the current key point descriptor of hand posture input.

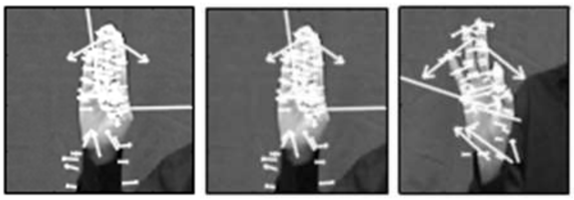


Figure 1.6 Key point descriptors

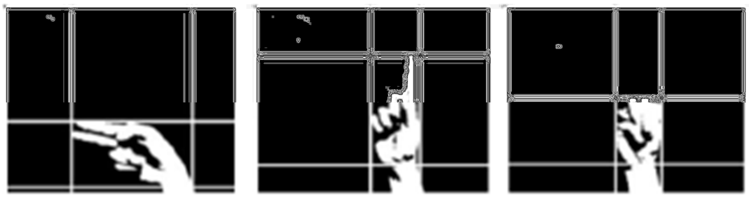
Budsara et al. [3] present an image processing method for Thai sign language identification. Optical flow is used for key frame selection. Feature extraction is achieved using finger pattern analysis. Finger pattern analysis starts from:

1). finding the hand boundary to specify the region of interest.

2). determining the hand orientation in 3 categories: horizontal, vertical and symmetrical, as in Figure 1.7.

3). determining the finger pattern to identify a special characteristic of each type of hand gesture such as the number of fingers in each direction, spreading fingers, and finger order.

The classification hand gesture is achieved simply by table lookup method in Table 1.1.



(a) (b) (c)

Figure 1.7 Hand orientation feature: (a) horizontal (b) vertical (c) symmetrical

Table 1.1 Example of Table lookup methods for classification hand gesture.

Alphabet	Orientation	Number of upper fingertips	Number of left fingertips	Number of right fingertips	Number of lower fingertips	Spread Finger	Finger Order	Note
B	Vertical	4	0	0	0	No	*	-

Furthermore, some world-level research has been proposed. Wuttichai [31] proposes Thai sign language recognition system using a Hidden Markov Model (HMM) for transcribing human sign language into text or speech. The system estimates hand posture using an electronic glove device called a CyberGlove. Sansanee [22] proposed Thai word sign language translation using SIFT with HMM. Thammanoon [27] proposed Thai Sign to Thai Machine (TSTM), an approach to machine translation which is used for translating from Thai sign language to Thai text. Phonkrit [14,15] built a dynamic Thai sign language translation system using upright speed-up robust feature and dynamic time warping, He also proposed Thai sign language translation system using upright speed-up robust feature and C-Mean clustering.

There are many researchers working on sign language. However, our work has focused only on finger-spelling. Therefore, a general condition comparison can be preformed between proposed methods, as show in Table 1.2. We have compared different conditions for handling several features. For instance:

1). the outfit feature considers whether the person signing wore long-sleeved clothing.

2). the device feature considers whether the signer uses spaciel device suchas electronic glove, color glove, marker, or depth camera.

3). the distance feature considers whether the signer requires fixing standing position, or the entire image contains only the hand posture.

4). the background feature considers whether the system uses uniform color background.

5). the process feature considers whether the system can immediately respond to signer.

Table 1.2 Work comparisons.

Work	language	method	Outfit	Device	Back ground	Distance	Real-time
[25]	ThSL	G-B*	No	Glove Sensor	No	No	Yes
[34]	ThSL	A-B*	No	No	Yes	Yes	No
[29]	ThSL	A-B*	No	No	Yes	Yes	No
[30]	ThSL	A-B*	No	Color glove	No	Yes	No
[23]	ThSL	A-B*	Yes	No	Yes	Yes	N/A*
[3]	ThSL	A-B*	No	No	Yes	Yes	No
[9]	ThSL	A-B*	No	No	Yes	Yes	Yes
[18]	ASL	A-B*	No	Depth camera	No	Yes	Yes
[24]	ASL	A-B*	No	No	Yes	Yes	Yes
[16]	Auslan	A-B*	Yes	No	Yes	Yes	Yes

Work	language	method	Outfit	Device	Back ground	Distance	Real-time
[10]	JSL	A-B*	No	Depth camera	No	Yes	Yes
[35]	JSL	G-B*	No	Glove sensor	No	No	Yes
[32]	CSL	A-B*	Yes	No	Yes	Yes	Yes
[2]	Korean	A-B*	Yes	No	Yes	Yes	Yes
[19]	BSL	A-B*	Yes	No	Yes	Yes	Yes
[6]	Turkish	A-B*	Yes	No	Yes	Yes	Yes
[17]	Malay	M-B*	No	Color glove	Yes	Yes	Yes
* G-B = Glove-Based, *M-B = Model-Based, *A-B = Appearance-Based							

In the Table 1.2, we have focused only on Thai finger-spelling. Many researchers have proposed work in this field. However, most of them have not yet achieved the critical criteria, such as, accuracy, flexibility, failure to cover all alphabets, or lack of real-time constraints. Other sign language finger-spelling recognition can overcome these problems. Therefore, for this reason, work in this field has become an interesting topic for getting robust and efficient systems for Thai finger-spelling sign language recognition.

1.3 Objectives

The major goal of this work was to propose a new technique for vision-based method hand posture estimation, which combines a model-based method and an appearance-based method together. Considered finger tracking to identify finger characteristic features and hand appearance features for increasing the accuracy of hand posture recognition. Furthermore, the algorithm will be adapted to a Thai alphabet finger-spelling sign language recognition system.

1.4 Scopes

The developed system covers only part of the Thai alphabet finger-spelling, which consists of spelling 42 letters and does not include vowels, intonation marks and the vocabulary recognition of the Thai alphabets finger-spelling combination. The system should not require much computation, and can be a run in real-time situation. Moreover, a depth sensor camera in millimeter units with a resolution of 320x240 pixels will be used to capture the hand posture image to reduce the complexity of hand texture information.

1.5 Organization of Work

The organization is as follows: This chapter introduces the motivation, objective and scope of the work. Chapter 2 gives the details of background knowledge used to implement our method. Chapter 3 explains Thai finger-spelling theory and proposes a method for hand posture estimation, which is adaptive for Thai alphabet finger-spelling recognition system. Chapter 4 describes the experimental results for our software system. Finally, Chapter 5 discusses the conclusions of this work and possible future work

CHAPTER 2

THEORY AND CONCEPT

2.1 Image Moment

In image processing and computer vision, an image moment is defined by a certain particular weight of the image's intensities. The image moments are useful to describe binary objects after segmentation. Simple properties of an image object can be specified via image moments including area, centroid, axis, and orientation, as shown in Figure 2.1. In general, image moment can be defined as:

$$M_{i,j} = \sum_x \sum_y x^i y^j I(x,y) \tag{2.1}$$

where (i,j) indicates power exponents that define the moment order, (x,y) is the pixel coordinates, and I(x,y) is the intensity at pixel (x,y). The centroid of the image can be calculated as follows:

Find the zeroth moment

$$M_{00} = \sum_x \sum_y I(x,y) \tag{2.2}$$

Find the first moment for x

$$M_{10} = \sum_x \sum_y xI(x,y) \tag{2.3}$$

Find the first moment for y

$$M_{01} = \sum_x \sum_y yI(x,y) \tag{2.4}$$

Then, the centroid is

$$X_c = \frac{M_{10}}{M_{00}}, \quad Y_c = \frac{M_{01}}{M_{00}} \tag{2.5}$$

The orientation of the image moment is obtained by using the second moment as in the following formula:

Second moment for x is

$$M_{20} = \sum_x \sum_y x^2 I(x, y) \tag{2.6}$$

Second moment for y is

$$M_{02} = \sum_x \sum_y y^2 I(x, y) \tag{2.7}$$

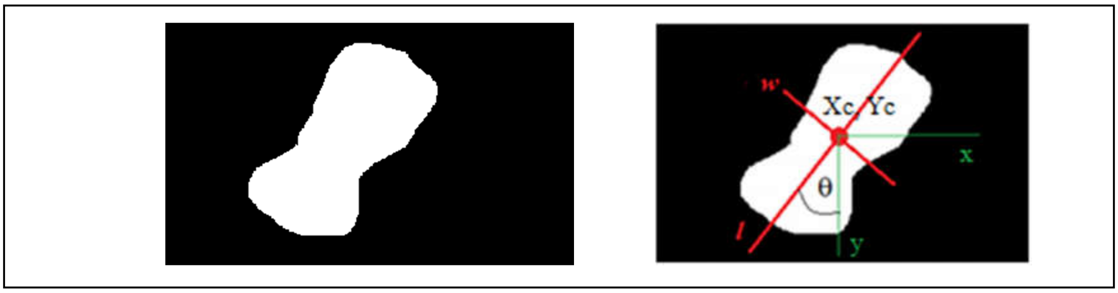
Then, the orientation can be performed as

$$\theta = \frac{1}{2}\arctan\left[\frac{2\left(\frac{M11}{M00} - XcYc\right)}{\left(\frac{M20}{M00} - Xc^2\right) - \left(\frac{M02}{M00} - Yc^2\right)}\right] \tag{2.8}$$

The length *l* and *w* axis for the centroid of the object can be calculated as follows:

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}, \quad w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \tag{2.9}$$

$$a = \frac{M20}{M00} - Xc^2, \quad b = 2\left(\frac{M11}{M00} - XcYc\right), c = \frac{M02}{M00} - Yc^2 \tag{2.10}$$



(a) (b)

Figure 2.1 Image moment: (a) original image (b) image properties.

2.2 **Active Contour Model**

The Active contour model is more widely known as the snake model, which are the energy-minimizing curves of the deformable splines influenced by constraints called image forces that pull its curves towards to the object contours. The simple snake model is defined by a set of N control points of spline, internal energy, and external energy. This model attempts to minimize an energy which is associated with the current contour as the sum of internal and external energies:

$$E_{snake} = \int_0^1 E_{snake}\big(V(S)\big)\ ds \tag{2.11}$$

$$= \int_0^1 \big(\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{image}\big)\ ds \tag{2.12}$$

Each energy term serves a different purpose:

- E_{cont} forces the contour to be continuous.
- E_{curv} forces the contour to be smooth.
- E_{image} attracts the contour toward the closest image edge.

E_{cont} and E_{curv} are internal energies and E_{image} is external energy. The parameters α, β, and γ control the relative influence of the corresponding energy terms, and can vary as continuous values. The snake model is a contour represented parametrically as V(s) = (x(s), y(s)) where x(s) and y(s) are the coordinates along the contour and s $\in$ [0,1] as shown in Figure 2.2.

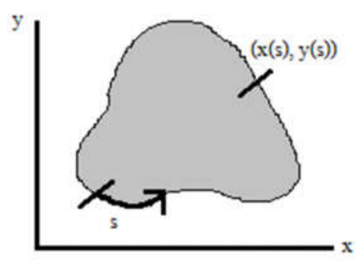


Figure 2.2 Snake contour.

The conventional continuity energy term is the E_{cont}. Its main role is to equalize spaces between control points, by minimizing the difference between the average distance and the distances among neighboring snake control points. In the discrete case, the contour is approximated by N control points $p_1, p_2, ..., p_N$ and the first derivative is approximated by a finite difference:

$$E_{cont} = \left\| p_i - p_{i-1} \right\|^2 \tag{2.13}$$
$$= (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \tag{2.14}$$

This formula tries to minimize the distance between control points. However, this formula may produce a non-uniform distribution of control points along the contour object, compacted in some regions and spread out in others. To keep a uniform average distance between the control points, the formula works much better is

$$E_{cont} = \left(\overline{d} - \left\| p_i - p_{i-1} \right\| \right)^2 \tag{2.15}$$

$\overline{d}$ is the average distance between any two consecutive points of the snake contour. The new energy E_{cont} of equation (2.15) attempts to keep equal distances between control points (i.e. spread them equally along the snake).

The curvature energy term is called E_{curv}, Its main role is to form a smooth contour between neighboring control points, and to avoid oscillation by penalizing high contour curvatures. In the discrete case, the curvature can be approximated by the following finite difference:

$$E_{curv} = \left\| p_{i-1} - 2p_i + p_{i+1} \right\|^2 \tag{2.16}$$
$$= (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \tag{2.17}$$

The edge energy term is called E_{image}, Its purpose is to attract the initial contour toward the target contour. This can be achieved by the following function:

$$E_{image} = -\left\| \nabla I(x, y) \right\| \tag{2.18}$$

The intensity I(x,y) is a gray-level image is a function of the continuous position variables (x,y), and ∇ is the gradient of the intensity computed at each snake point. However, E_{image} becomes very small when the snake points get close to an edge.

The energy function is computed for each p_i, i = 1... N, and each of its m x m neighbors its to find the location that minimizes the functional energy. The p_{i-1} point has already been moved to new position. Its location is used to compute the continuity energy term for p_i and its neighbor's points. The location of p_{i+1} has not moved yet. Its location is used to compute the curvature for p_i and each point in the neighborhood of p_i. The location having the smallest value is chosen as the new position of p_i. Figure 2.3 and Table 2.1 have shown how the method works and Figure 2.4 has shown sequence of snake execution.

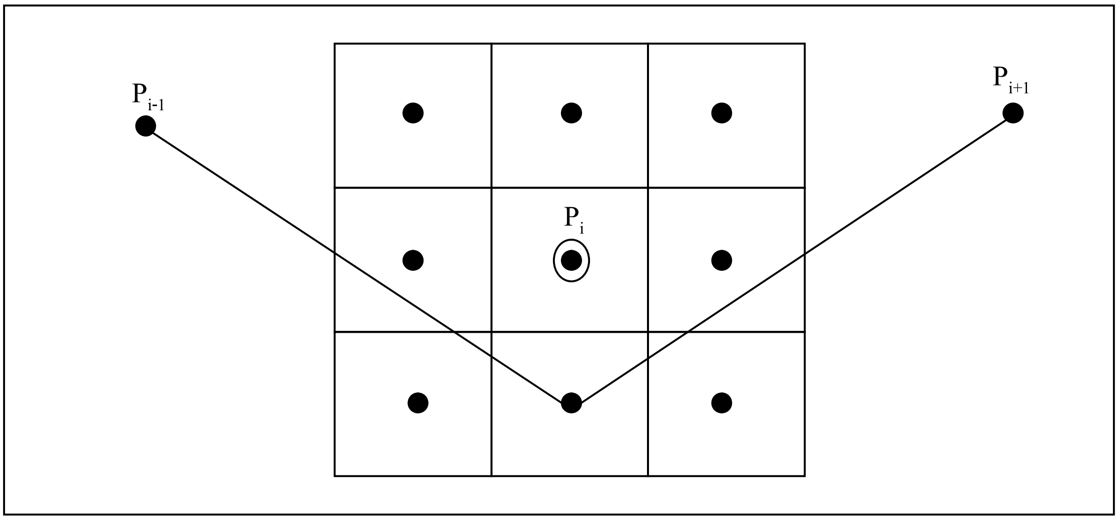


Figure 2.3 The energy function is computed at p_i and each of its eight neighbors.

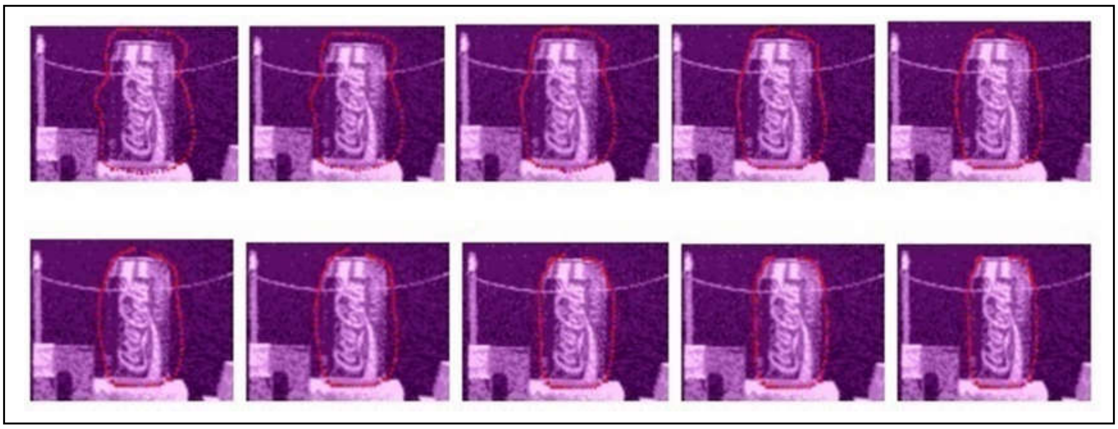


Figure 2.4 Sequence of snake execution.

Table 2.1 The active contour (snake) pseudo-code.

PSEUDO-CODE

Index arithmetic is modulo n.
Initialize α_i, β_i and γ_i to 1 for all i.
Do
    /* loop to move points to new location */
    For i = 0 **to** n /* point 0 is first and last one processed */
        E_{min} = **BIG**
        For j = 0 *to* m-1 /* m is size of neighborhood */
            $Ej = \alpha_i E_{cont,j} + \beta_i E_{cur,j} + \gamma_i E_{image,j}$
            If $E_j < E_{min}$ **then**
                $E_{min} = E_j$
                $Jmin = j$
        Move point $\mathbf{p}_i$ to location *jmin*
        If *jmin* **not** current location, *ptsmoved* += 1. /* count points moved */
    /* process determines where to allow corners in the next iteration */
    For i = 0 **to** $n - 1$
        $c_i = \left\| \vec{p}_i / \|\vec{p}_i\| - \vec{p}_{i+1} / \|\vec{p}_{i+1}\| \right\|^2$
    For i = 0 **to** $n - 1$
        If ($c_i > c_{i-1}$ **and** $c_i > c_{i+1}$ /* **if** curvature is larger than neighbors */
        and $c_i >$ *threshold1* /* **and** curvature is larger than threshold */
        and *mag*($\mathbf{p}_i$) > *threshold2* /* **and** edge strength is above threshold */
            Then $\beta_i = 0$ /* relax curvature at point */
Until *ptsmoved* < *threshold3*

2.3 Fuzzy Logic

A fuzzy logic system (FLS) can be defined as the nonlinear mapping of an input data set to a scalar output data which consists of three main parts: fuzzier, inference engine and defuzzier as shown in Figure 2.5. The process of fuzzy logic is explained as follows: firstly, a crisp set of input data is gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions. This step is known as fuzzification. Afterwards, an inference is made based on a set of rules. Lastly, the resulting fuzzy output is mapped to a crisp output using the membership functions, in the defuzzification step.

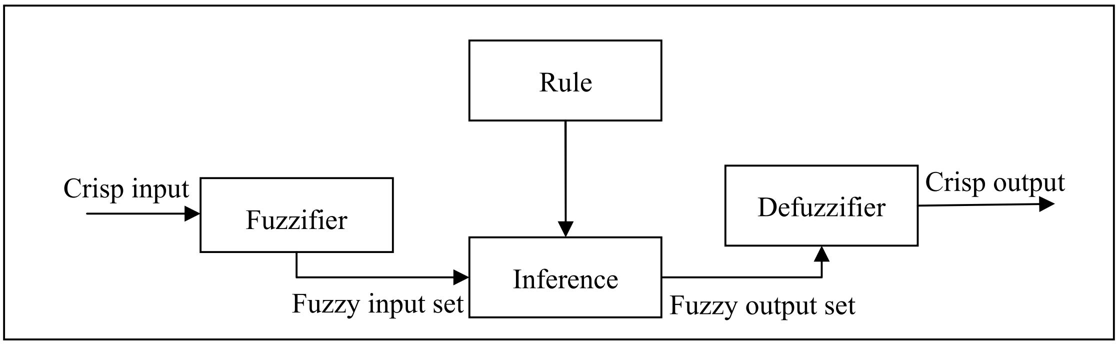


Figure 2.5 Fuzzy logic systems.

2.3.1 Fuzzifier

In the fuzzifier process, a crisp set of input data is gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions. The linguistic variables are the input or output variables of the system whose values are words or sentences from a natural language, instead of numerical values. A linguistic variable is generally decomposed into a set of linguistic terms. Assume that temperature (t) is the linguistic variable which represents the temperature of a room. To qualify the temperature, terms such as “hot” and “cold” are used in real life. These are the linguistic values of the temperature. Then,

$$T(t) = \{\text{too-cold, cold, warm, hot, too-hot}\}$$

T(t) can be the set of decompositions for the linguistic variable temperature. Each member of this decomposition is called a linguistic term, and can cover a portion of the overall values of the temperature. Membership functions are used in the fuzzifier and defuzzifier steps of a fuzzy logic system, to map the non-fuzzy input values to fuzzy linguistic terms and vice versa. On output, a membership function is used to quantify a linguistic term. For instance, in Figure 2.6

, membership functions for the linguistic terms of temperature variable are plotted. Note that an important characteristic of fuzzy logic is that a numerical value does not have to be fuzzified using only one membership function. In other words, a value can belong to multiple sets at the same time. For example, a temperature value can be considered as “cold” and “too-cold” at the same time, with different degree of membership than shown in Figure 2.6.

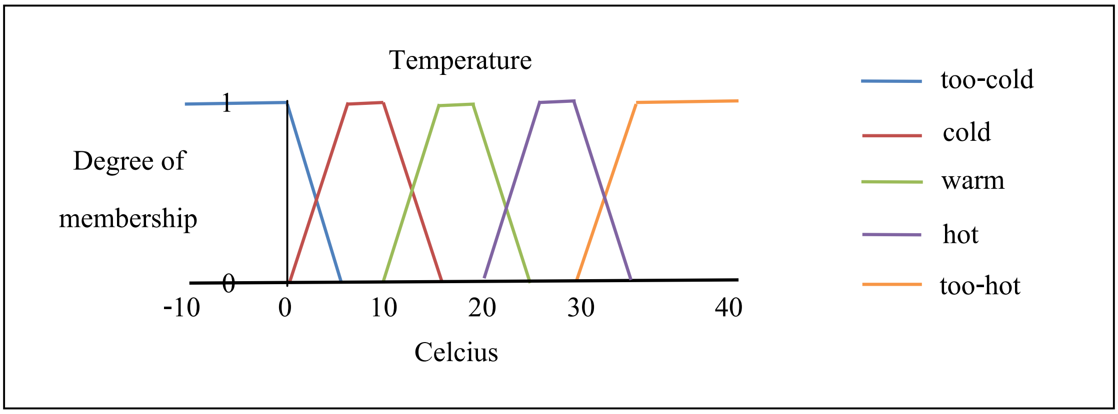


Figure 2.6 Membership function for T(temperature) = {too-cold, cold, warm, hot, too-hot}.

There are different forms of membership functions such as triangular, trapezoidal, Gaussian, or singleton as shown in Figure 2.7. The type of the membership function can be context dependent, and it is generally chosen arbitrarily according to the user experience.

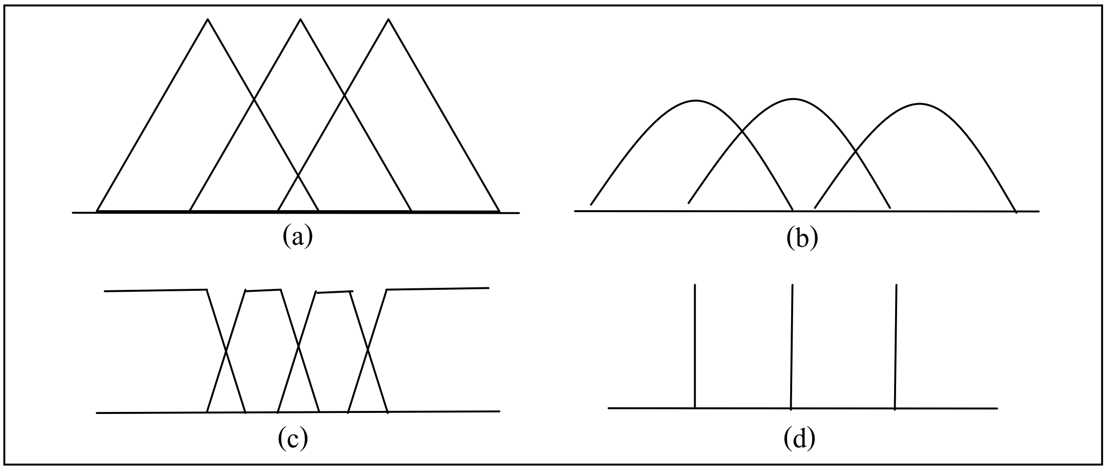


Figure 2.7 Membership functions: (a) triangular (b) Gaussian (c) trapezoidal (d) singleton.

2.3.2 Fuzzy Rules

In a fuzzy logic system, a rule base is defined to manage the fuzzy output variable. A fuzzy rule uses a simple IF-THEN rule with a condition and a conclusion. Table 2.2 shows the matrix representation of the fuzzy rules for the fuzzy logic system. Row captions in the matrix contain the values that current room temperature can take, column captions contain the values for target temperature, and each cell is the resulting command when the input variables take the values in that row and column. For instance, the cell (3, 4) in the matrix can be read as follows: If temperature is cold and target is warm then command is heat.

Table 2.2 Fuzzy matrix example of action.

		Target temperature				
Current temperature		**too-cold**	**cold**	**warm**	**hot**	**too-hot**
	too-cold	no-change	heat	heat	heat	heat
	cold	cool	no-change	***heat***	heat	heat
	warm	cool	cool	no-change	heat	heat
	hot	cool	cool	cool	no-change	heat
	too-hot	cool	cool	cool	cool	no-change

2.3.3 Fuzzy Set Operation

The evaluation of the fuzzy logic rules and the combination of the result of the individual rules are performed by using fuzzy set operations. The operations that is used most often in fuzzy logic system are OR (max), AND (min), and NOT (complement), respectively. After evaluating the result of each rule, these results should be combined to obtain a final result. This process is called inference. The maximum algorithm is generally used for accumulation.

2.3.4 Defuzzification

After the inference process, the overall result is a fuzzy value. This result should be defuzzified to obtain a final result crisp output. In Figure 2.8, the shaded areas all belong to the fuzzy result. The purpose is to obtain a crisp value, represented at a dot in the Figure 2.8, from this fuzzy result. There are many algorithms for defuzzification. However, the algorithm that is used most often is the center of gravity calculation (COG).

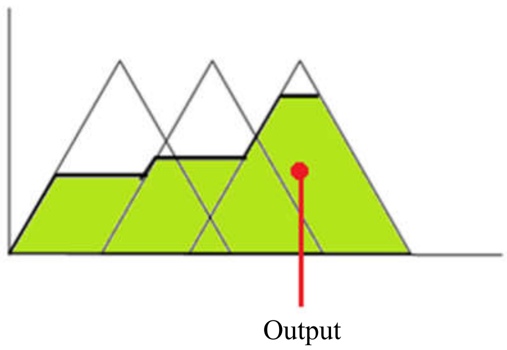


Figure 2.8 Center of gravity on defuzzification step.

2.4 The Ramer-Douglas-Peucker Algorithm

The Ramer–Douglas–Peucker algorithm (RDP) is an algorithm for reducing the number of points of a curve that is approximated by a set of points. The initial form of the algorithm was independently suggested in Urs Ramer, David Douglas and Thomas Peucker. The purpose of this algorithm is to find a similar curve with fewer points. The simplified curve consists of a subset of the points that defined the original curve. The algorithm can be calculated as follow:

1). In Figure 2.9 (a), the starting curve is an ordered set of points or lines and the distance dimension $\varepsilon > 0$.

2). Divide the line. Initially, it is given all the points between the first and last point. It automatically marks the first and last points to be kept as shown in Figure 2.9.

3). Then, find a point as the end point that is furthest from the line segment formed by the first and last points. If the distance between those points to the line segment is greater than ε, then the approximated point must be kept. The algorithm must be executed recursively for the new approximated line segments, as shown in Figure 2.9 (c). In Figure 2.9 (d) and 2.9 (e), The procedure is repeated until the error distance is less than ε. Finally, the new output curve is the defined by a new set of points obtained during each step, as shown in Figure 2.9 (f).

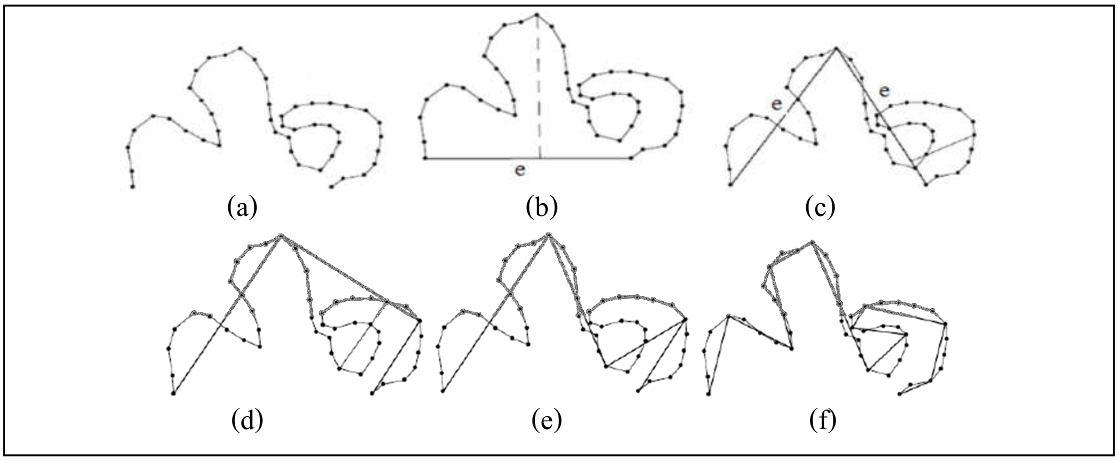


Figure 2.9 The RDP algorithm process.

2.5 Hidden Markov Model

The Hidden Markov Model (HMM) is a mathematical model of a stochastic process based on Markov theorem, where these processes generate a probability of random outcome sequence. The HMM is widely used in science, engineering, and many other areas (speech recognition, optical character recognition, machine translation, bioinformatics, computer vision, finance and economics, and in social science). The HMM consists of important element as follows:

- The set of states $S = \{s_1, s_2, ..., s_N\}$ where N is number of states.
- An initial probability for each state π_i , $i = 1, 2, ..., N$ such that $\pi_i = P(s_i)$ at the initial step.
- 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.
- A set of observation symbols $O = \{o_1, o_2, ..., o_M\}$, where M is the number of observation symbols.
- An N-by-M observation matrix, $B = \{bj(o_k)\}$ where $b_j(o_k)$, give the probability of emitting observation symbol o_k from state j.

The compact notation to indicate the complete parameter set of a model is shown in equation (2.19). The HMM can be divided into two parts: the training model and the evaluation model.

$$\lambda = (A, B, \pi) \tag{2.19}$$

2.5.1 Training model

For the training model, this step attempts to optimize the model parameters (A, B, π). The observation sequence used to adjust the model (λ) is called a training sequence since it is used to train the HMM. The forward-backward algorithm has been used to calculate the probability of each training sequence. Evaluation of new model parameters is calculated using the Baum-Welch method to create best model for real phenomena, as shown in Figure 2.10.

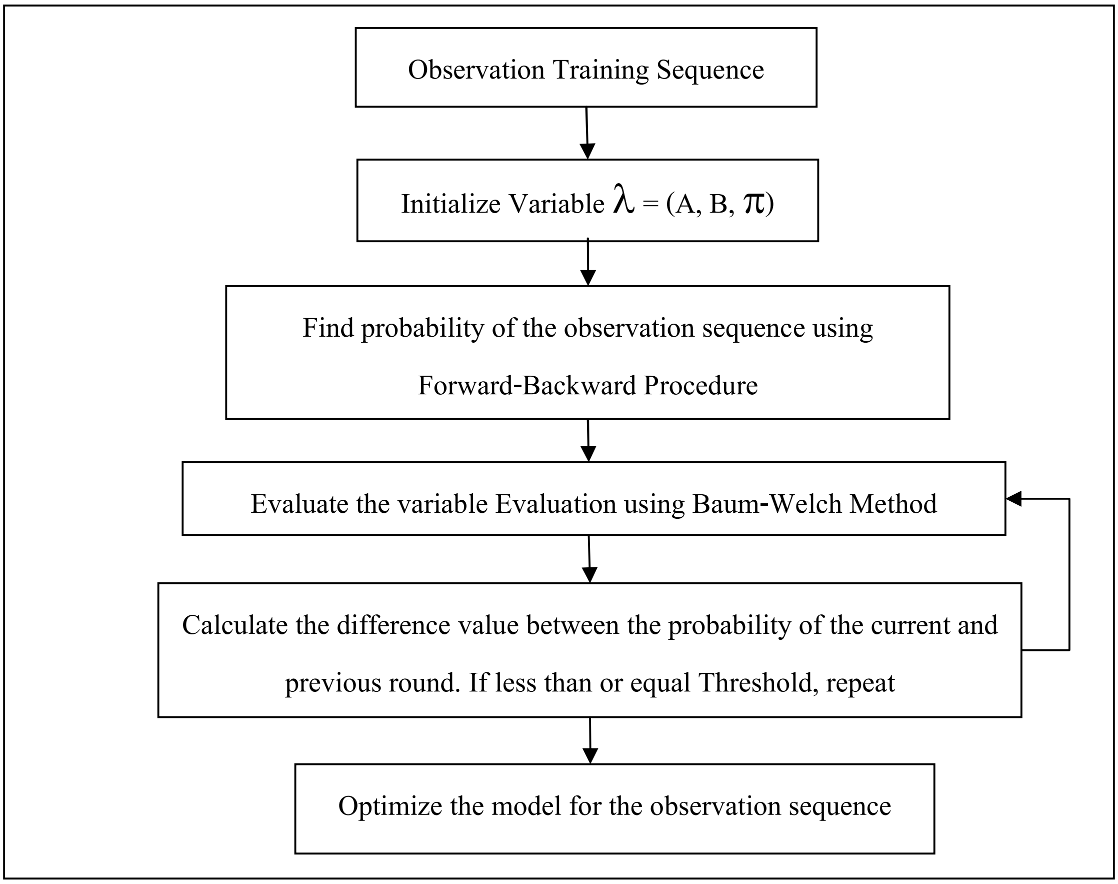


Figure 2.10 Training model process.

Calculating the probability of the input observation sequence by using the Forward-Backward procedure can be described as follows:

In the forward algorithm, the forward variables are defined as following:

- $\alpha_t(i)$ Forward variable of state i at time t.

- π_i Probability of state i at time t.
- $b_j(O_t)$ Probability of observation sequence of state j at time t.
- a_{ij} Probability of taking the transition from state *i* to state *j*.
- Initialization

$$\alpha_1(i) = \pi_i b_i(O_1) \quad , \quad 1 \le i \le N \tag{2.20}$$

- Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] . b_j(O_{t+1}) \;,\; 1 \le t \le T-1 \;,\; 1 \le j \le N \tag{2.21}$$

- Termination

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.22}$$

In the backward algorithm, the backward variable defines as:

- $\beta_t(i)$ Backward variable of state i at time t.
- Initialization

$$\beta_{T(i)} = 1 \;,\; 1 \le i \le N \tag{2.23}$$

- Induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \;,\; t = T-1, T-2, \ldots, 1 \;,\; 1 \le i \le N \tag{2.24}$$

Adjust the model parameters (A, B, π) to maximize the probability of the observation sequence given the model. An iterative procedure such as the Baum-Welch method (or equivalently the expectation-modification) can be used for model parameter choosing.

- Define ξ_t(i,j) as the probability of being in state i at time t and state j at time t+1.

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{2.25}$$

- Define γ_t(i) as the probability of being in state i at time t, given model λ, and observation sequence O relative to all ξ_t(i,j)

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j) \tag{2.26}$$

- Evaluating new values of A, B and π

$$\pi = \gamma_1(i) \tag{2.27}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{2.28}$$

$$b_j(k) = \frac{\sum_{t=1 \atop s,t,O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{2.29}$$

2.5.2 Evaluation model

For the evaluation model, the probability of input observation sequence P(O|λ) for each HMM model is evaluated by the forward algorithm to compute the best path per HMM. The best path represents the most likely sequence of states in HMM. The input observation sequence is considered as recognized when the model with the maximum probability is found, as shown in Figure 2.11.

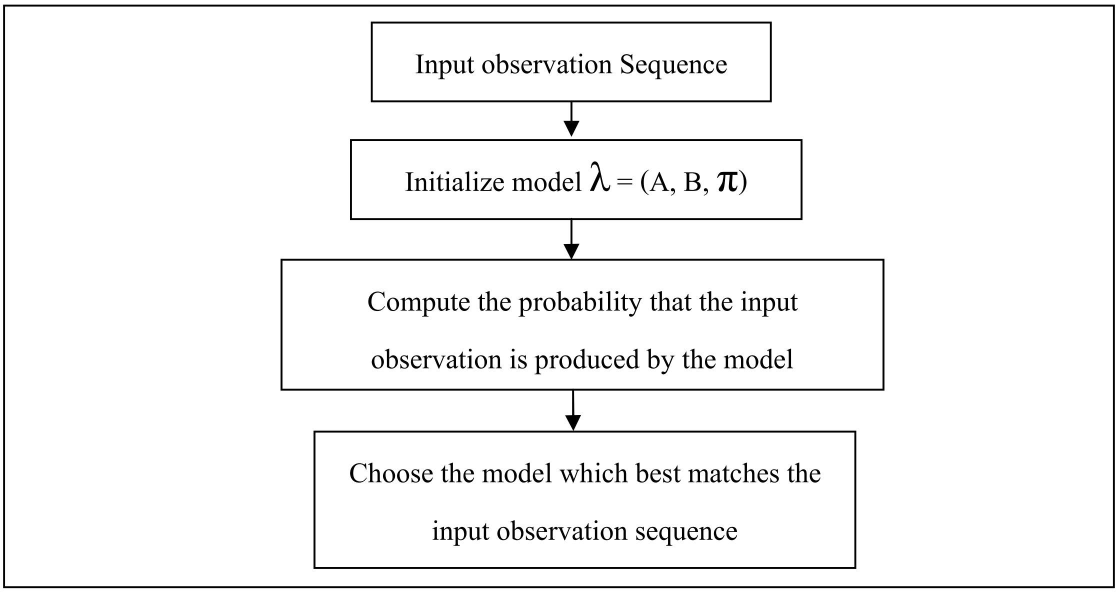


Figure 2.11 Evaluation model process.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Thai Finger-Spelling

Thai finger-spelling is the usage of hand posture for representing the alphabet, vowels, intonation marks, and numbers to spell the specific names, places, or technical words. Thai finger-spelling was developed in A.D. 1956 (Lady Karmonla Kririt). It is based on American finger-spelling, as shown in Figure 3.1.

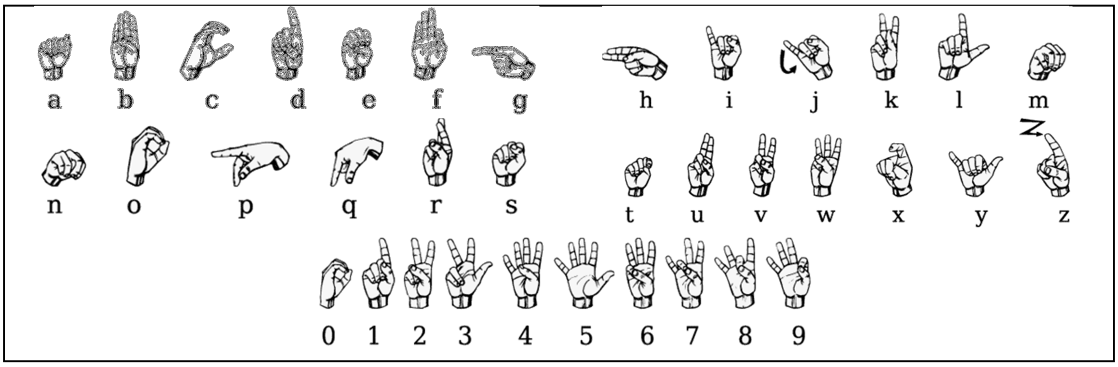
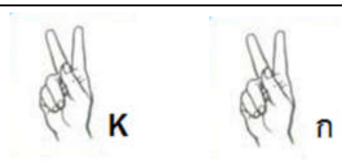


Figure 3.1 The American finger-spelling hand posture.

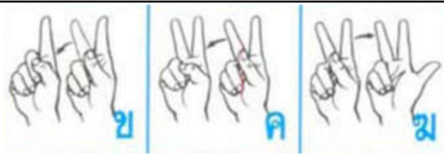
In Figure 3.5, the Thai finger-spelling is compared to the phonetics of American finger-spelling. Finger-spelling will be matched to American finger-spelling whose sound particular character is similar. For example, Thai alphabet “ก (Ko kai)” has a similar sound to the “K” letter in American finger-spelling. Therefore, the hand posture of “K” letter will be used to represent “ก (ko kai)” in Thai finger-spelling, as shown in Figure 3.2.



(a) (b)

Figure 3.2 Thai Finger-spelling: (a) hand posture of “K” (b) hand posture of “ก (Ko kai)”.

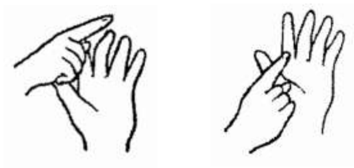
However, additional finger-spellings as the combination of hand postures of American finger-spelling, are extended in order to represent all 42 Thai letters. For example, “ข (Kho khai)” = k+1 as shown in Figure 3.3 (a). The usage frequencies of letters determine the order of finger-spelling for consecutive letters, such as “ข (Kho khai)” is spelled before “ค (Kho khwaiand)”, and “ฆ (Kho ra-khang)”, as shown in Figure 3.3 (b) and Figure 3.3 (c), respectively.



(a) (b) (c)

Figure 3.3 The order of finger-spelling: (a) “ข (Kho khai)” = k+1 (b) “ค (Kho khwai)” = k+2 (c) “ฆ (Kho ra-khang)” = k+3.

Comparing to the English language, in the Thai language there are additional vowels and intonation marks. The extended finger-spellings using a combination of two hands are defined in order to represent the meaning. For instance, some vowels such as “-ะ (Sara a)”, “-า (Sara ar)” or “เ- (Sara ae)” or intonation marks “ -้ (Mai tho)” are made this way. The two hands of Thai finger-spelling are formed by a dominant hand (either the left or the right hand) for the letter and by a subordinate hand, which selects the particular vowel by its position number, as shown in Figure 3.4.



(a) (b)

Figure 3.4 Two-hand sign: (a) spelling “ -้ (Mai tho)” (b) spelling “-า (Sara ar)”.

Figure 3.5 shows the complete set of Thai finger-spellings that was designed. It does not contain value for the old Thai consonants “ฅ” (Kho khon) and “ฃ” (kho khuat). The 42 of Thai letters and the 7 of the Thai vowels symbols have finger-spelling that require only one hand. The other symbols require two hands, and more than one such finger-spelling may be needed to represent the phoneme of that vowel.

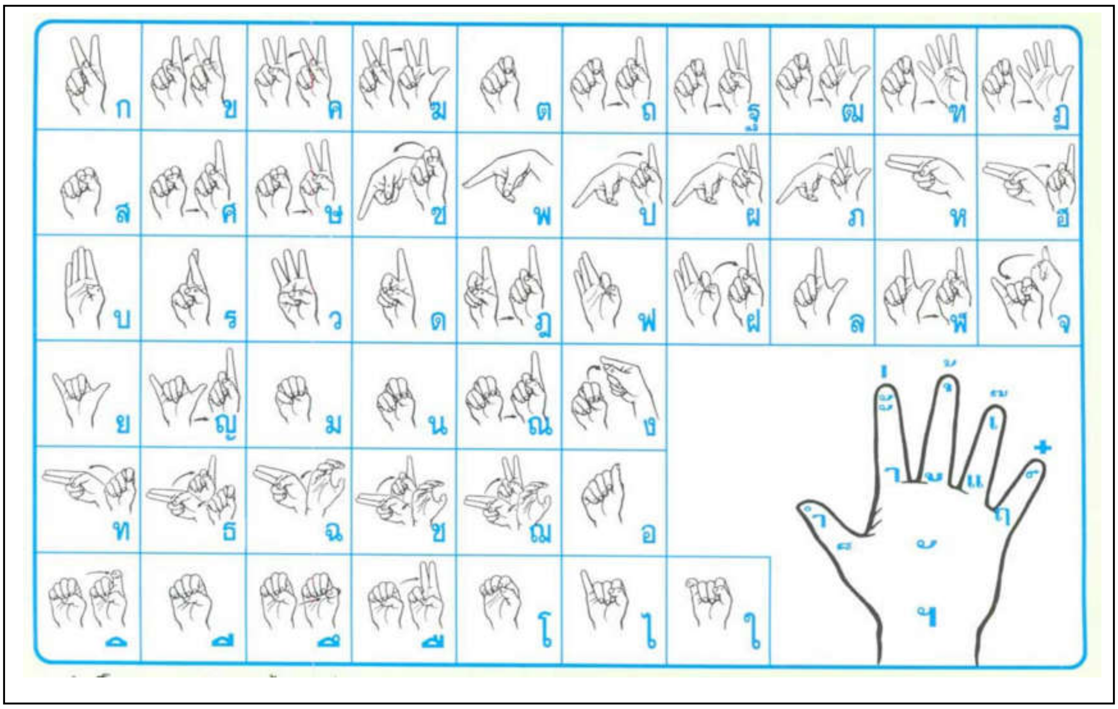


Figure 3.5 Thai finger-spelling.

3.2 Proposed Method the

In this research, the work concentrated on Thai alphabet finger-spelling for sign language. A visual prototype for the 42 Thai letters finger-spelling recognition system was proposed, which considered only a single instance of right-hand finger-spelling in front of a single video camera. The system consisted of four main parts:

1) Hand segmentation: segment the region of interest of the hand from an image sequence, and calculate initial features.

2) Key hand posture selection: determine the key frame representing the hand posture of finger-spelling from the image sequence.

3) Hand feature definition: define the finger shapes and hand appearance features, and represent that a chain code sequence.

4). Recognition model: recognize a single Thai letter finger-spelling from hand features using the scored voting method and the Hidden Markov Model learning based (HMM). The system overview is shown in Figure 3.6.

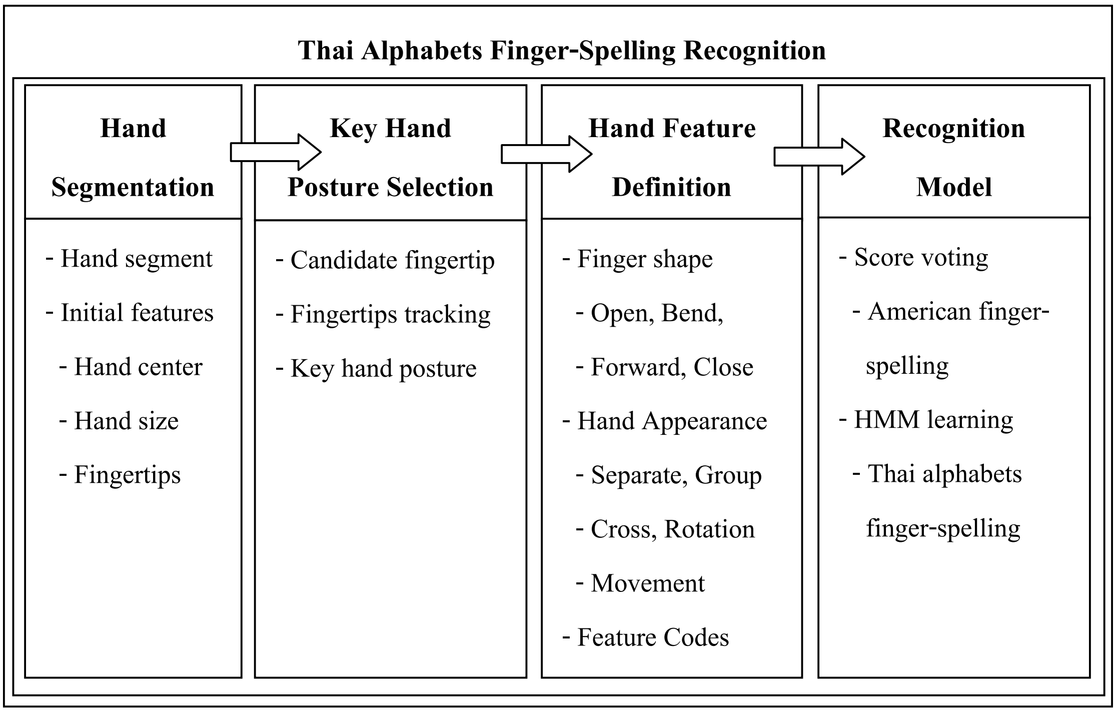


Figure 3.6 The system overview

3.2.1 Hand Segmentation

In this step, the method focused on the region of interest, which is segmented from the depth image, and extract necessary hand features to initialize the system.

A. Hand region Segmentation

We assume in Figure 3.7 (a) that user's right is the closest object to the camera. Using a depth image solved the problem of segmenting the hand from the complex background. Depth information is a 3D position vector (x,y,z) obtained from the depth camera. The the x and y values are distance information corresponding to the rows and columns of the array as in an ordinary image. The z value is the corresponding depth reading that is stored in the pixels, for detecting how far away from camera that the object is.

(a) (b) (c)

Figure 3.7 Depth information: (a) complex background (b) depth image (c) hand's region.

The depth image in Figure 3.7 (b) provides a strong clue for separating between the hand and the complex background (the closer object is the darker intensity). Thus, the system takes advantage of depth information to extract only the hand's region of the image as shown in Figure 3.7 (c). After extracting hand's region, the initial hand's features will be estimated, such as hand center, fingertips position, palm size. All initial hand features are used as reference values to compare changing hand gestures in the finger-spelling recognition system.

B. Hand Center Position

We obtained the center point of the hand's region, which can be computed easily from the moments of pixels in the hand's region, defined as:

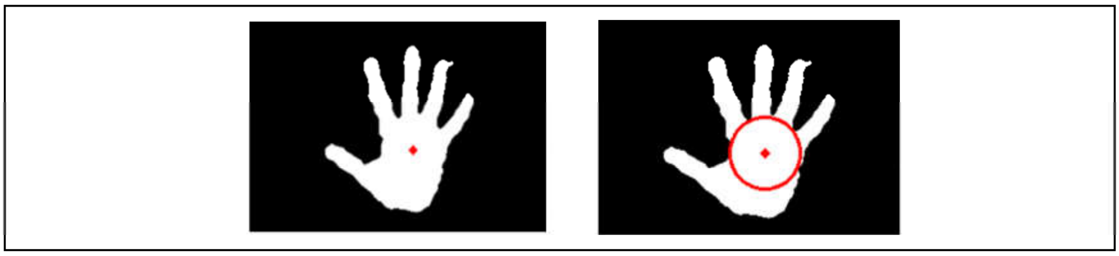
$$M_{ij} = \sum_{x}\sum_{y} x^i y^j I(x,y) \tag{3.1}$$

In the above equations, $I(x,y)$ is the pixel value at the position (x,y) of the image x and y range over the hand's region. The hand center point, $C_0(X_c, Y_c)$, is calculated as show in Figure 3.8 (a).

$$Xc = \frac{M10}{M00}, \quad Yc = \frac{M01}{M00} \tag{3.2}$$

The palm size (R) is defined as the closest Euclidean distance (d) between the center point of hand (C_0) and the pixel (P) on hand boundary (B), as shown Figure 3.8 (b).

$$R = \underset{P \in B}{argmin} \{d(C_0, P)\} \tag{3.3}$$

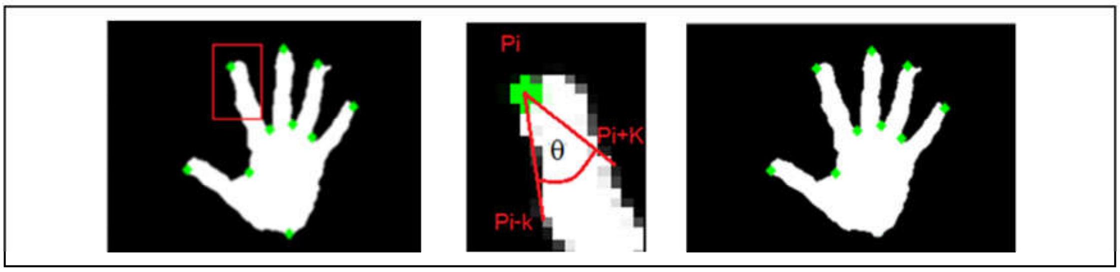


(a) (b)

Figure 3.8 Hand region: (a) hand center (b) palm size.

C. Fingertip Position

The user is required to initialize the system by producing the pose of an "open" right hand facing the camera as shown in Figure 3.9 (a). Therefore, it is simple to locate the fingertips from the curvature of the boundary point of the hand's region. However, it may not be necessary to consider all the boundary points of the hand's region. Thus we used the polygon approximation method to extract key point [34] and stored in a new series of key point $P_1,\ldots,P_n$. Each key point P_i has two parameters, the angle (θ) and slope direction (D_i). In Figure 3.9 (b), the angle can be estimated by using k-curvature [13], which calculates the angle of a key point by two vectors [$P(i\text{-}k)P(i)$] and [$P(i)P(i+k)$] with the same range (k). The key point, if its curvature value is in the appropriate threshold and slope direction is positive ($D_i = 1$), is a candidate fingertip. From this method, five fingertip points and four valley points between fingertips (D_i=-1) are detected as shown in Figure 3.9 (c).



(a) (b) (c)

Figure 3.9 Fingertips position: (a) key point (b) curvature calculation (c) fingertip points and valley points between fingertips.

After the initial positions of the five fingertips are detected, each of them will be given a label that corresponds to the thumb, index, middle, ring and pinky fingers. Since we constrain initial open hand posture to frontal view, we can simply label fingers based on sorting the five points by clockwise arrangement around palm center. Nevertheless, it can only detect fingertips in an open hand. Thus, a tracking method is used for fingertips tracking, giving the changed position at all times in finger-spelling process.

3.2.2 Key Hand Posture Selection

This section, details how to select the key frames of the hand posture that represent the finger-spelling letters in the image sequences. Note that not every frame in the image sequences is necessary to the recognition system.

A. Locating Fingers

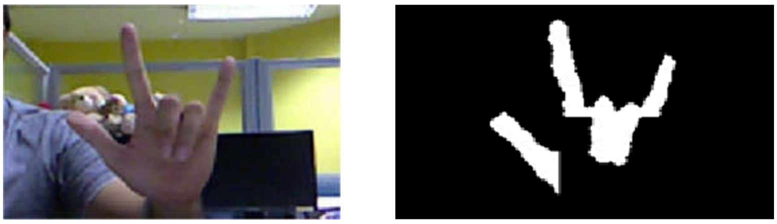
Most movements in finger-spelling are finger movements, as in Figure 3.10 (a). Therefore, we have defined two conditions to segment finger locations from depth image in Figure 3.10 (b). For a stretching finger, using distance condition (*d*), it is apparent that:

$$d\left(p, C_0\right) > k.d_\tau \tag{3.4}$$

where p is the pixel on hand region, C_0 is the hand center, k is a scaling factor, and d_τ is a pre-defined distance threshold. For a bending finger, using depth condition (*z*), it is apparent that follow:

$$\left|Z(p) - Z(C_0)\right| < k.Z_\tau \tag{3.5}$$

where z(*p*) is the depth value of a pixel in the hand region, z(C_0) is the depth value of the hand center, k is a scaling factor, and z_τ is a pre-define depth threshold.

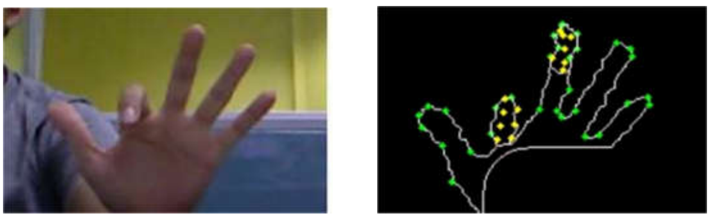


(a) (b)

Figure 3.10 Locating Fingers: (a) origin hand posture (b) extraction of finger location using pre-defined conditions.

B. Candidate Fingertips

We define searching an area to locate the candidate fingertip positions (Cf_i). We assume that fingertip positions should be points on the hand contour. However, the hand contour points do not change much from the current point to the next one. Therefore, it may not be necessary to consider all points on the hand contour. Accordingly, the polygon approximation algorithm [36] is used to extract key points. Figure 3.11 (b) shows candidate fingertip positions on the contour for a stretching finger. In addition, candidate fingertip positions can be found by assuming that they are closest to the camera in each finger region. We use the depth value to find the point which has minimum depth to be candidate fingertips. Figure 3.11 (b) shows these points inside the contour.



(a) (b)

Figure 3.11. Candidate fingertips: (a) orgin hand posture (b) candidate fingertip using polygon approximation and minimum depth value.

C. Fingertip Tracking

Fingertip positions changed over time, thus a tracking process is used to track fingertip positions which occur in a finger movement sequence. The correspondence of the fingertip position between successive frames is built by the concept of an active contour [4,12]. The possible candidate fingertips (Cf_i) of each fingertip will be assigned energy values and then the maximum energy point is chosen to be the fingertip in the next frame. It is simpler to go directly to a discrete formulation of the energy function which can be written as:

$$E(Cf_i) = E_{internal}(Cf_i) + E_{external}(Cf_i) \tag{3.6}$$

The energy for each candidate fingertip can be decomposed into two basic energy terms. $E_{internal}$ represents the internal energy of the candidate due to bending or stretching of the finger, and $E_{external}$ is the external constraint introduced by user. The internal energy of candidate fingertips is defined as:

$$E_{internal} = E_{continuity} + E_{curvature} + E_{depth} + E_{direction} \tag{3.7}$$

The $E_{continuity}$ is the energy of continuity. It forces the candidate fingertip points to be continuous, because the fingertip (f_j) should not change much from the current point to the next one. Therefore, this term tries to find the minimize distance between the average distance ($\overline{d}$) and candidate fingertip point. The $E_{continuity}$ attempts to keep the points at appropriate distance. The form of $E_{continuity}$ is:

$$E_{continuity} = \overline{d} - \left\| f_j - Cf_i \right\|^2 \tag{3.8}$$

The purpose of $E_{curvature}$ this term is to find the smoothness of the candidate fingertips by considering contour curvature. In the discrete case, $E_{curvature}$ is the angle between the two vectors **A** = ($x_i - x_{i-k}$, $y_i - y_{i-k}$) and **B** = ($x_{i+k} - x_i$, $y_{i+k} - y_i$), where k is constraint. The curvature threshold will be defined during the initial process. If the candidate fingertips have a curvature value that fall under a predefined threshold, these points will be kept to be possible candidate fingertips. The formula for $E_{curvature}$ is:

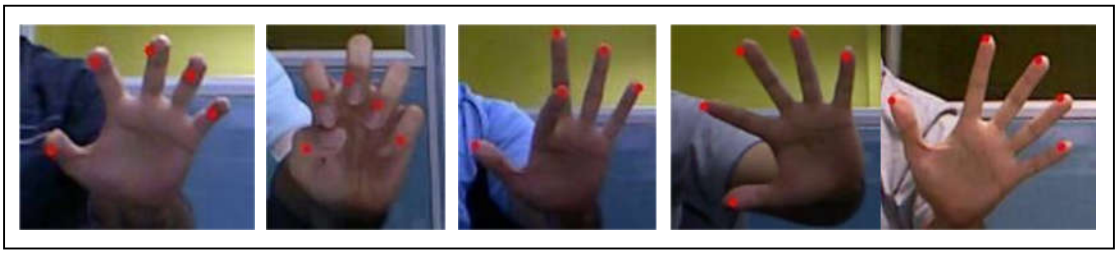
$$E_{curvature} = \cos^{-1} \frac{\mathbf{A}.\mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|} \tag{3.9}$$

Because we have used the property of depth, which E_{depth} has been established. The E_{depth} is distance between the candidate fingertips and the camera, which represent 16-bit depth data units in millimeters. As we have assumed, the fingertips should be found at the closest point to the camera when a finger is bending or moving into the palm. Therefore, E_{depth} becomes a maximum value when the candidate fingertips get close to the camera. The closest point will be given more priority than the rest of the points in order of depth.

A fingertip has moved from its location in frame i to a new location in frame i+1. Additionally, our method has used the direction to estimate the new location of the fingertips. Suppose that $\mathbf{f}_j$ is the current vector of the fingertip, and $\boldsymbol{cf}_j$ is the vector of candidate fingertips. Thus, the candidate fingertips should have the same direction as the movement of fingertips. The direction will be considered from the positive and negative signs of a vector in x and y axis. The $E_{direction}$ occurs from vector from $\mathbf{f}_j$ to $\mathbf{c}_{fi}$ that can be defined as:

$$E_{direction} = \overrightarrow{\mathbf{f}_j \mathbf{c}f_i} \tag{3.10}$$

As we mentioned previously, the $E_{external}$ is the external constraint. Thus, in our system, the distances from all fingertips to considering candidate fingertip in image are used to describe the $E_{external}$. For instance, if we are considering the movement of the index fingertip, the distances from other fingertips to the candidate fingertip point are equivalent to the external energy, which will be used to estimate suitability for choosing the candidate fingertip to be the index fingertips. The distance from index fingertip to candidate fingertip should be shorter than to other fingertips. On the other hand, if the distances from other fingertips are shorter than the index fingertip distance, the candidate fingertip should be assigned to another fingertip. The energy function represents the importance of candidate fingertips relative to each fingertip. The candidate fingertip with maximum energy is selected to be the new location of the fingertip in the next frame. In order to reduce the tracking error due to losing depth value, a simple low-pass filter is applied for the smoothness trajectory of fingertip tracking. The average point between the selected point and the current fingertip point will be estimated. The pseudo-code for fingertip tracking the four basic finger motions (bending, closing, crossing, rotation) will be show in Table 3.1 and Figure 3.12.



(a) (b) (c) (d)

Figure 3.12 Finger tracking: (a) bending (b) closing (c) crossing (d) 45^0 hand rotation.

D. Key Hand Posture

The most stable frame image will be selected as the key hand posture in the finger-spelling flow. This can eliminate unimportant hand posture during transition which does not carry obvious meaning. The frame differencing method is used to detect the temporal difference between successive frame images, and to determine the frame image with minimum difference as the key hand posture in the video sequence. The form for frame differencing is:

$$D_k(x,y) = \sum_{i=0}^{n} \left| P_{k-1}(x,y) - P_k(x,y) \right|_i \leq \tau_D \tag{3.11}$$

The frame differencing can be computed by differencing in terms of the fingertip's position, where D represents the frame difference, and P represents the fingertip's position. The frame differencing is calculated to find a stable hand posture where the difference falls under a predefined threshold. This is called key hand posture. In Figure 3.13, frames that are in the red square will be selected for key hand postures because differencing of fingertip's positions are quite stable, and they fall under a predefined threshold.

Figure 3.13 Key hand posture selections.

Table.3.1. Fingertip tracking algorithm

Pseudo-Code
Input: The initial fingertips, $\{f_j\}$
Output: The updated position of the fingertips
Do /******** *loop to move point to new position******** /*
For $j = 0$ **To** n /* *n is number of fingertips* */
$E_{max} = 0$ /* *initial energy value* */
$I = 0$ /* *initial selected candidate fingertip (x,y)* */
For $i = 0$ **To** m /* *m is number of candidate fingertips* */
$E_i = E_{internal} + E_{external}$ /* *sum of energy function*/
If $E_i > E_{max}$ **Then**
$E_{max} = E_i$ /* *assign energy value* */
$I = i$ /* *assign selected candidate fingertip (x,y)* */
End
$f_j = avg(f_j, I)$ /* *update fingertips position* */
End
End
Return $\{f_j\}$ /*return the adjusted f_j fingertips positions*/

3.2.3 Hand Feature Definition

The definition of hand features is detailed in this section. Two types of characteristics are used in order to obtain the invariant and robust features: finger shape features and hand appearance features. All features will be encoded as chain code feature sequences to represent hand posture.

A. Finger Shape Features

Finger shapes will be used to discriminate the difference between a pair of elements in the set of hand postures. For reduced computational complexity, there are four basic fingers shapes (“Open”, “Bend”, “Point” and “Close”) as shown in Figure 3.14.

(a) (b) (c) (d)

Figure 3.14 Basic finger shapers: (a) open (b) bend (c) point (d) close.

In this step, fuzzy logic [5,8] is used to classify finger shapes. The process of fuzzy logic is explained in Figure 3.15. Firstly, a crisp set of input data is gathered and converted to fuzzy sets using linguistic variables, fuzzy linguistic terms and membership functions (μ). This step is called fuzzification. Subsequently, an inference is made to get an evaluation result, based on a set of rules. Finally, the resulting fuzzy output is mapped to a crisp output using membership functions, in the defuzzification step.

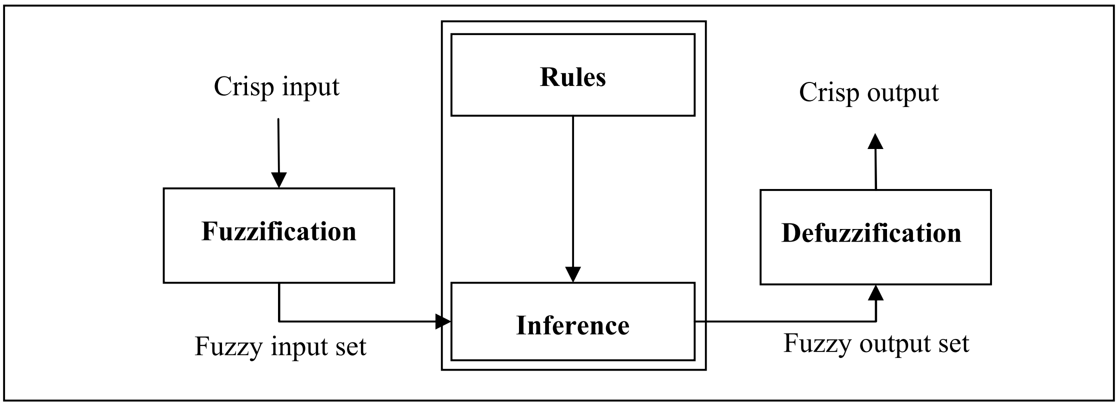


Figure 3.15 The fuzzy logic system.

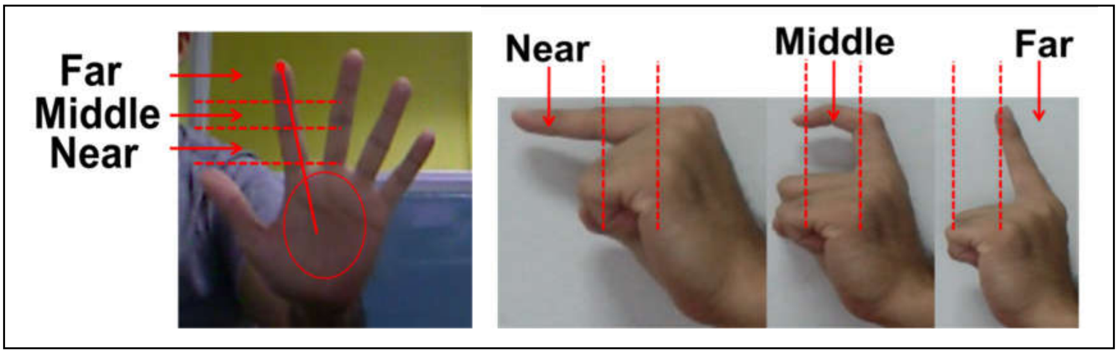
In the system, the linguistic variables are defined as depth(D), distance(Dist) and shape(S). The depth (depth value at the fingertip position, *D(t)*) and distance (distance from the fingertip position to the hand center point, *Dist(t)*) are input variables, but shape is an output variable. In linguistic terms, the depth and distance variables are decomposed into a set of "Near", "Middle" and "Far" terms:

$$D(t) = \{Near, Middle, Far\}\ ,\ Dist(t) = \{Near, Middle, Far\}.$$

The range of each term is shown in Figure 3.16. For the shape variable (*S(t)*), the linguistic terms are open, bend and point:

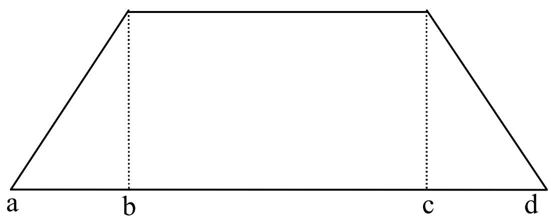
$$S(t) = \{Open, Bend, Point\}$$

We do not include "close" in the set of shape lingusitic terms because the "close" shape will be classified immediately when the fingertip position is in the palm region. In Figure 3.17, we use a trapezodial curve as the type of the membership function of each linguistic term. The fuzzification will be done to convert all crisp input to fuzzy set input as shown in Figure 3.18.



(a) (b)

Figure 3.16 Linguistic terms: (a) distance (b) depth.



$$\left\{\begin{array}{ll} 0 & ,x \le a \\ \dfrac{x-a}{b-a} & ,a \le x \le b \\ 1 & ,b \le x \le c \\ \dfrac{d-x}{d-c} & ,c \le x \le d \\ 0 & ,d \le x \end{array}\right\}, x \in R$$

Figure 3.17 Trapezoidal curve membership functions.

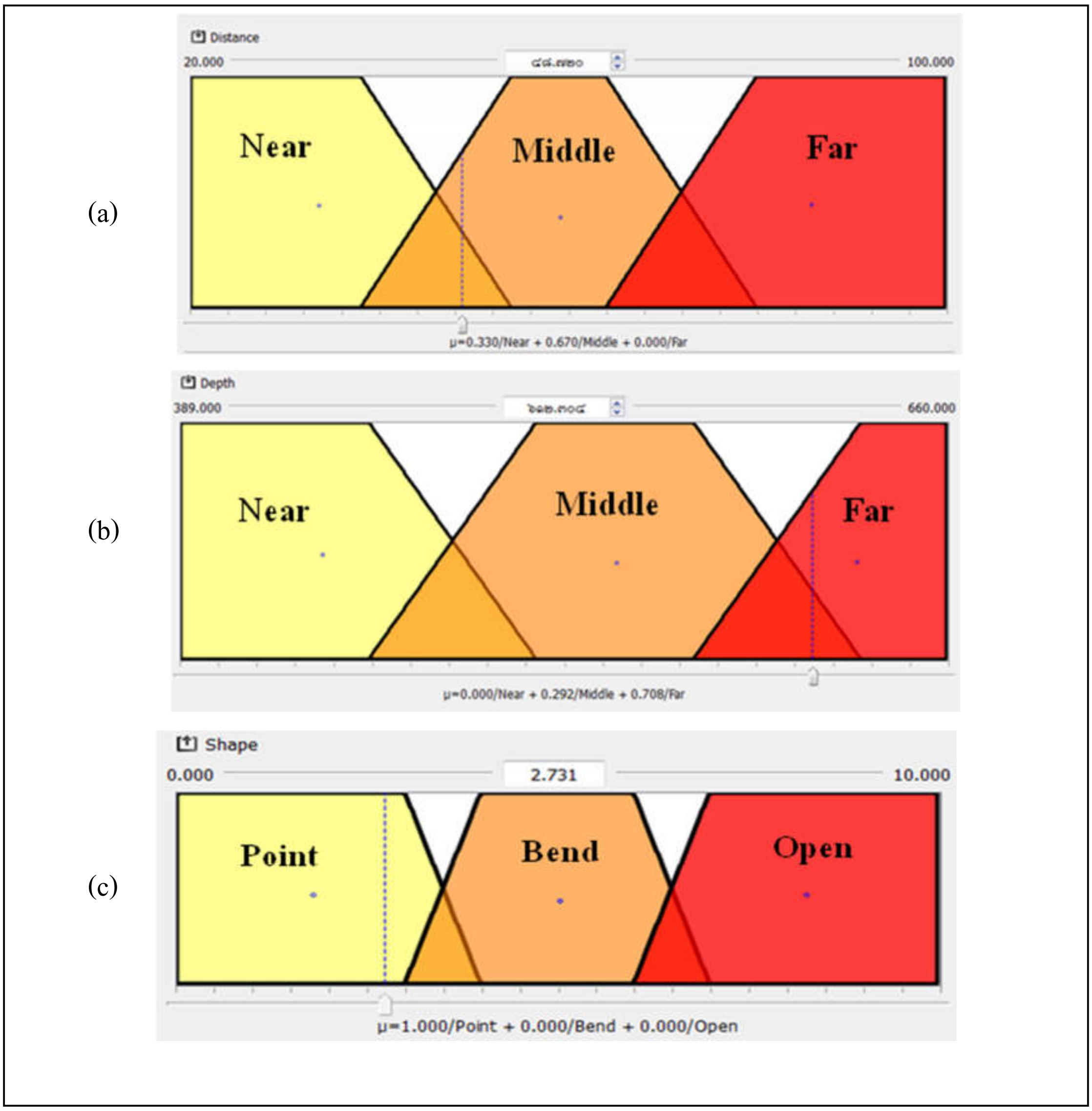


Figure 3.18 Fuzzy sets: (a) distance (b) depth (c) shape.

The rule base is defined to control the output variable. The fuzzy rule is a simple IF-THEN rule with a condition and conclusion. Table.3.2 shows the matrix representation of the fuzzy rules. Row captions contain the value that depth can take, column captions contain the value for distance, and each cell is the resulting shape when the input variables take the values in that row and column. For example, the cell (1,1) in matrix can be read as follows: IF distance is "Near" AND depth is "Near" THEN shape is "Point". Table.3.3 shows fuzzy rule for the finger shape classification.

Table 3.2 Fuzzy matrix.

	Depth			
Distance		**Near**	**Middle**	**Far**
	Near	***Point***	Point	Point
	Middle	Point	Bend	Bend
	Far	Open	Open	Open

Table 3.3 Fuzzy rules for finger classification system.

Fuzzy Rules
1. IF Distance is Near AND Depth is Near THEN Shape is Point
2. IF Distance is Near AND Depth is Middle THEN Shape is Point
3. IF Distance is Near AND Depth is Far THEN Shape is Point
4. IF Distance is Middle AND Depth is Near THEN Shape is Point
5. IF Distance is Middle AND Depth is Middle THEN Shape is Bend
6. IF Distance is Middle AND Depth is Far THEN Shape is Bend
7. IF Distance is Far AND Depth is Near THEN Shape is Open
8. IF Distance is Far AND Depth is Middle THEN Shape is Open
9. IF Distance is Far AND Depth is Far THEN Shape is Open

The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations, this process is called inference. The operation that is used often for fuzzy logic are OR*(max)*, AND*(min),* and NOT*(complement)*, respectively. After evaluating the result of each rule and combing them to obtain a final result, Defuzzification is performed according to the membership functions of the fuzzy output variable. For instance, we have the result in Figure 3.19 at the end of inference.

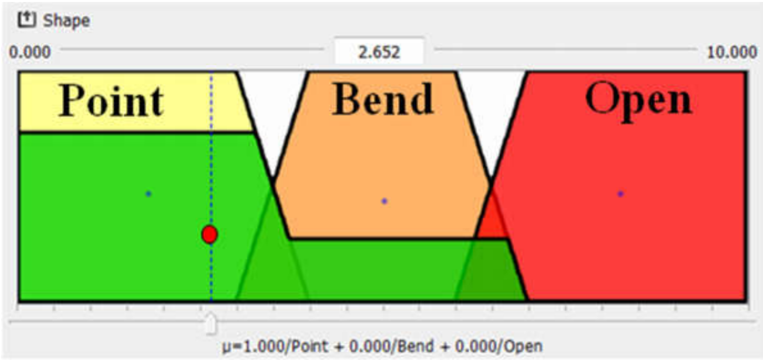


Figure 3.19 Fuzzy set result.

In Figure 3.19, the shaded areas all belong to the fuzzy result. This result should be defuzzified to obtain a final crisp output. There are different algorithms for defuzzification. We have used center of gravity (COG) to map fuzzy output to be a crisp output, represented with a dot in the Figure 3.19, from the fuzzy result. The COG algorithm is written as:

$$COG = \frac{\sum_{i=1}^{N} x\mu(x)}{\sum_{i=1}^{N} \mu(x)} \tag{3.12}$$

The example of finger shape feature classification which uses fuzzy logic for “Open”, “Bend” and “Point” shapes is shown in Figure 3.20.

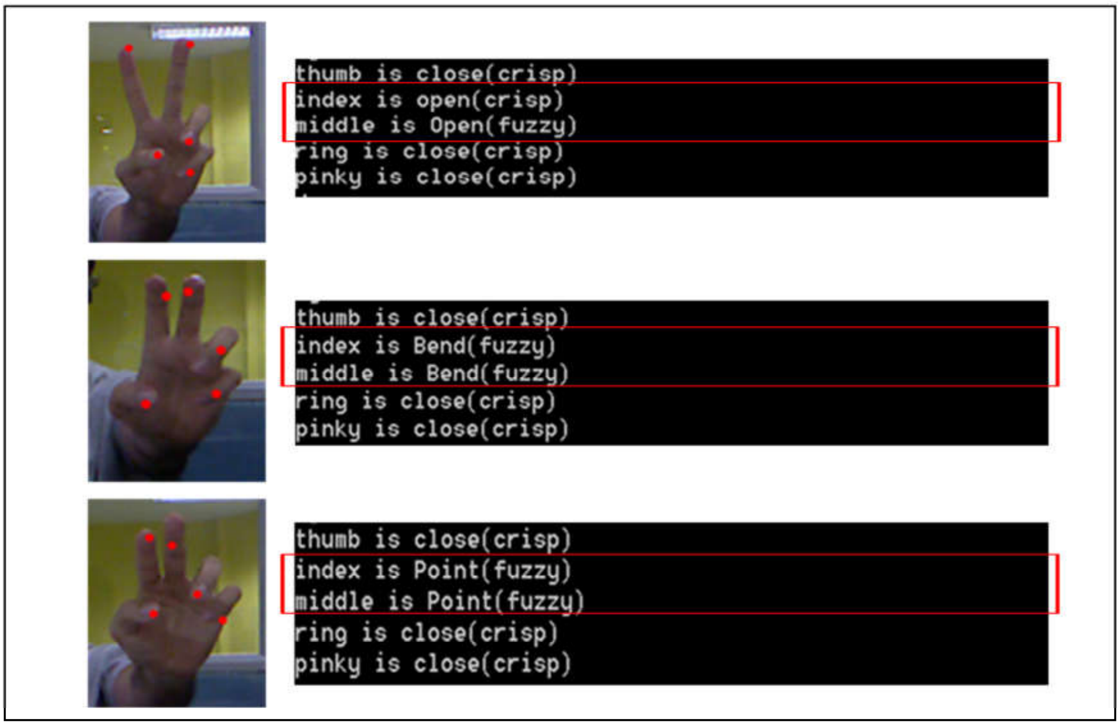
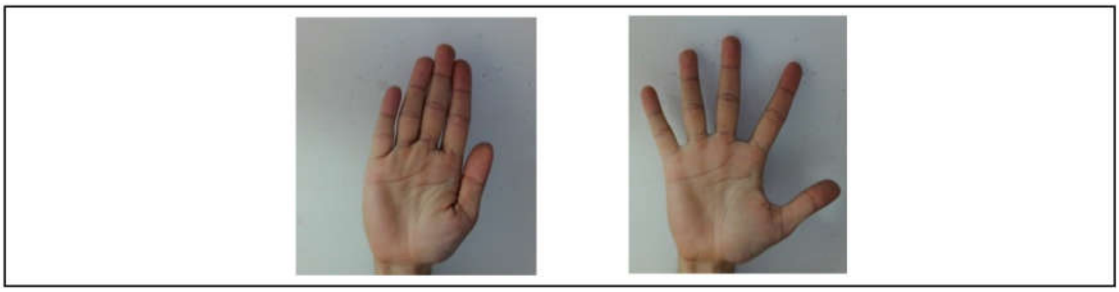


Figure 3.20 Finger shape feature classification using fuzzy logic.

B. Hand Appearance Features

Considering only finger shapes is not sufficient to estimate hand posture for finger-spelling. Generally, there are some hand postures that have similarity of finger shape, but have disparity of hand appearance. The hand appearance feature consists of finger relation, hand rotation and movement.



(a) (b)

Figure 3.21 Hand posture: (a) group (b) separate.

In the Figure 3.21 (a) and 3.21 (b), even though both hand postures have the same finger shape (all finger are open), but distinguish at finger relations. In Figure 3.21 (a), all finger relations are closed. In Figure 3.21 (b), all finger relations are separated. Thus, we have defined the finger relation as "Group", "Separate" and "Cross" by using distance (d) and fingertip's position (p) which covers for all set of hand postures in finger-spelling system as shown in Figure 3.22. The condition for finger relation can be defined as following:

- group finger relation:

$$d \leq d_{\tau} \tag{3.13}$$

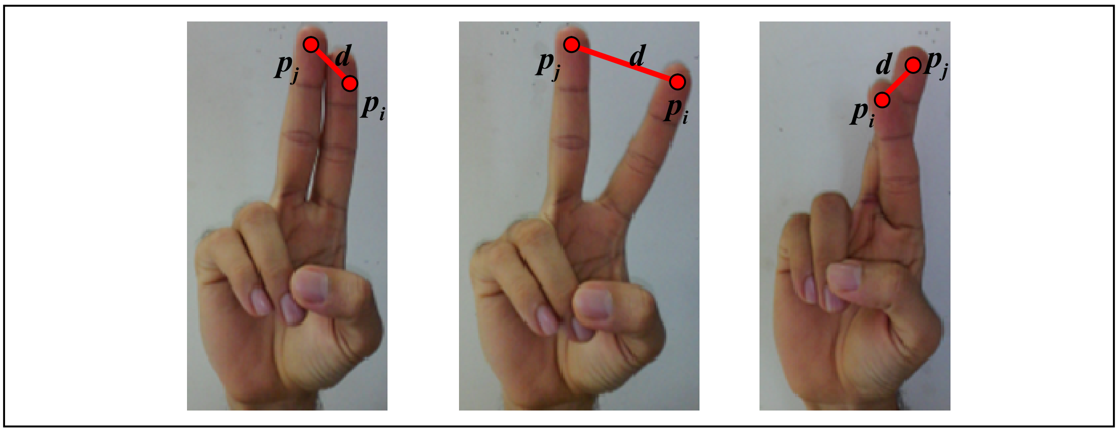
- separate finger relation:

$$d > d_{\tau} \tag{3.14}$$

- cross finger relation:

$$d \leq d_{\tau} \text{ and } p_i < p_j \tag{3.15}$$

where d is a distance between fingertip, and p is a fingertip's position.



(a) (b) (c)

Figure 3.22 Finger relations: (a) group (b) separate (c) cross.

Furthermore, hand rotation and hand movement will be used with another feature for represent difference of each hand posture in finger-spelling. The hand rotation can be calculated from orientation using image moment method. The hand movement computes from different of hand center's position in sequence. The hand rotation and movement feature will be shown in Figure 3.23.

Figure 3.23 Hand appearance: (a) Hand rotation (b) Hand movement.

C. Chain Codes Feature

To represent the hand posture, both finger shapes and hand appearance will be encoded as shown in Table 3.4.

Table 3.4 Finger shape and hand appearance feature codes.

Code	Finger shape	Code	Hand appearance	Code	Hand appearance
1	Open	5	Group	9	Movement
2	Bend	6	Separate	0	No-Rotation
3	Forward	7	Cross	0	No-Movement
4	Close	8	Rotation	-	-

Each key hand posture consists of features encoded as a feature array. The 17 feature array units are used to build hand posture chain codes which each in element is defined as follows:

f_1	f_2	f_3	f_4	f_5	f_{12}	f_{13}	f_{14}	f_{15}	f_{23}	f_{24}	f_{25}	f_{34}	f_{35}	f_{45}	h_r	h_m

In our system, the feature vector is composed of 17 values defined as shown in Figure 3.24. The first 5 elements (f_1,f_2,f_3,f_4,f_5) indicate the finger shape (thumb, index, middle, ring and pinky respectively). In Figure 3.24, (f_1,f_2,f_3,f_4,f_5) = (4,1,1,4,4) means that the thumb is "close", index and middle are "open", ring and pinky are "close", respectively. The next 4 elements (f_{12} ,f_{13},f_{14},f_{15}) describe the relation between thumb with index, middle, ring and pinky fingers. In Figure 3.24, (f_{12},f_{13},f_{14},f_{15}) = (6,6,6,6) indicates that all relations of thumb with other fingers are "separate". The next 3 elements (f_{23},f_{24},f_{25}) describe the relation of index with middle, ring and pinky fingers. The next 2 elements (f_{34},f_{35}) describe the relation of middle with ring and pinky fingers. In Figure 3.24, (f_{34},f_{35}) = (6,6) indicates that all relations of middle with other fingers are "separate". The element (f_{45}) describes the relation of ring and pinky. In Figure 3.24, (f_{45}) = (5) indicates that the relation of ring with pinky finger is "close". The next 2 elements (h_r,h_m) represent the rotation and movement of hand. In Figure 3.24, (h_r,h_m) = (0,0) indicates that there is no rotation or movement for this hand posture. Finally, the vector feature of the hand posture in Figure 3.24 have the chain codes as P = {4,1,1,4,4,6,6,6,6,6,6,6,6,6,5,0,0}.

Hand posture chain codes feature

f_1 = 4	f_2 = 1	f_3 = 1	f_4 = 4	f_5=4
f_{12} = 6	f_{13} = 6	f_{14} = 6	f_{15}= 6	
f_{23} = 6	f_{24} = 6	f_{25}= 6		
f_{34} = 6	f_{35} = 6			
f_{45} = 6				
h_r= 0	h_m= 0			

Figure 3.24 Hand posture encoding.

3.2.4 Recognition Model

In this step, we describe the about finger-spelling recognition process. Since Thai alphabet finger-spelling is based on American finger-spelling, the recognition model will consist of two major processes.

A. Scored Voting Recognition

The voting and scoring methods are combined and used in our recognition process. This method is simple and effective and can run in real time. Firstly, the *n* patterns of the hand posture chain codes (P_i | i = 1…N) are pre-defined because they were computed in off-line. Each pattern represents a finger-spelling that is the chain code of finger shape features and hand appearance features. Secondly, any hand posture which is captured from the camera will be detected and encoded into the same type of chain codes. Thirdly, the similarity between the detected pattern and the *n* pre-defined patterns is calculated by using a simple scored voting method. This technique will compare each element of the input chain codes with the corresponding element of the pattern chain code. The score will be given if any element has matched as shown in Figure 3.25. Thus, the hand posture will be recognized as the pattern which gives the maximum score. This technique is applied for the American finger-spelling recognition system. There are 31 template hand postures that consist of 26 hand postures for American-English letters (A-Z) and 5 hand postures for 1-5 numbers. Each template hand posture is shown in Figure 3.26 and chain code in Table 3.5.

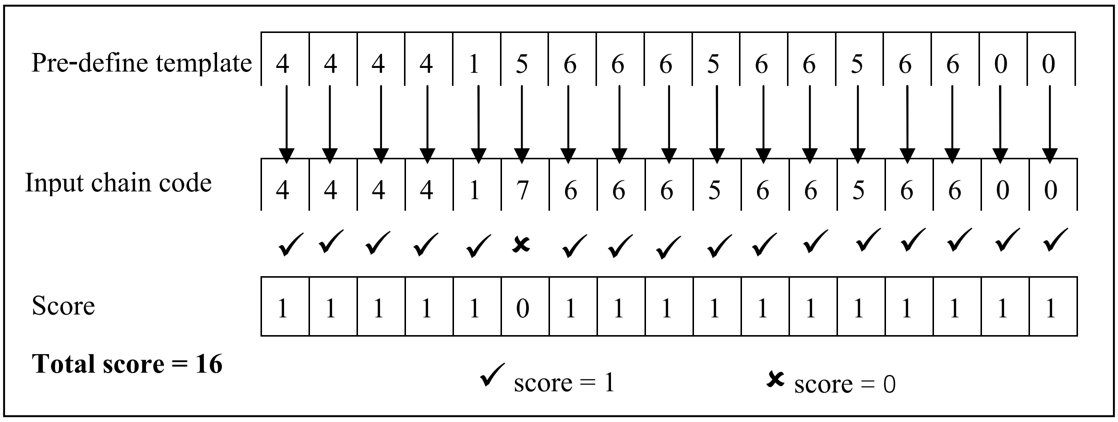


Figure 3.25 The scored voting.

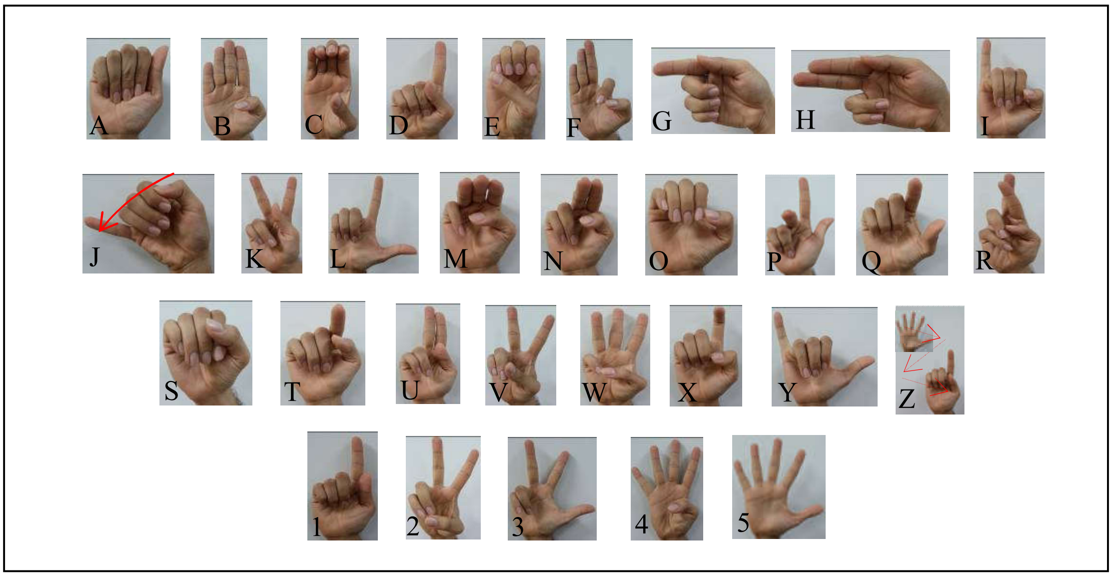


Figure 3.26 The 31 hand postures for American finger-spelling.

Table.3.5. hand posture feature codes

Hand posture	f_1	f_2	f_3	f_4	f_5	f_{12}	f_{13}	f_{14}	f_{15}	f_{23}	f_{24}	f_{25}	f_{34}	f_{35}	f_{45}	h_r	h_m
A	4	4	4	4	4	6	6	6	6	5	6	6	5	6	5	0	0
B	4	1	1	1	1	6	6	6	6	5	6	6	5	6	5	0	0
C	4	2	2	2	2	6	6	6	6	5	6	6	5	6	5	0	0
D	4	1	4	4	4	6	5	6	6	6	6	6	5	6	5	0	0
E	4	4	4	4	4	6	6	6	5	5	6	6	5	6	5	0	0
F	4	4	1	1	1	5	6	6	6	6	6	6	5	6	5	0	0
G	4	1	4	4	4	6	6	6	6	6	6	6	5	6	6	8	0
H	4	1	1	4	4	6	6	6	6	5	6	6	6	6	5	8	0
I	4	4	4	4	1	5	6	6	6	5	6	6	5	6	6	0	0
J	4	4	4	4	1	5	6	6	6	5	6	6	5	6	6	8	0
K	4	1	1	4	4	6	6	6	6	6	6	6	6	6	5	0	0
L	1	1	4	4	4	6	6	6	6	6	6	6	5	6	5	0	0
M	4	3	3	3	4	6	6	6	5	5	6	6	5	6	6	0	0
N	4	3	3	4	4	6	6	5	6	5	6	6	6	6	5	0	0
O	4	4	4	4	4	5	6	6	6	5	6	6	5	6	5	0	0

Hand posture	**f_1**	**f_2**	**f_3**	**f_4**	**f_5**	**f_{12}**	**f_{13}**	**f_{14}**	**f_{15}**	**f_{23}**	**f_{24}**	**f_{25}**	**f_{34}**	**f_{35}**	**f_{45}**	**h_r**	**h_m**
P	1	1	3	4	4	6	6	6	6	6	6	6	6	6	5	0	0
Q	3	3	4	4	4	6	6	6	6	6	6	6	5	6	5	0	0
R	4	1	1	4	4	6	6	6	6	7	6	6	6	6	5	0	0
S	4	4	4	4	4	7	5	6	6	5	6	6	5	6	5	0	0
T	4	3	4	4	4	6	5	6	6	6	6	6	5	6	5	0	0
U	4	1	1	4	4	6	6	6	6	5	6	6	6	6	5	0	0
V	4	1	1	4	4	6	6	5	6	6	6	6	6	6	5	0	0
W	4	1	1	1	4	6	6	6	5	6	6	6	6	6	6	0	0
X	4	2	4	4	4	6	5	6	6	6	6	6	5	6	5	0	0
Y	1	4	4	4	1	6	6	6	6	5	6	6	5	6	6	0	0
Z	4	1	4	4	4	6	5	6	6	6	6	6	5	6	5	0	9
1	4	1	4	4	4	6	6	6	6	6	6	6	5	6	5	0	0
2	4	1	1	4	4	6	6	5	5	6	6	6	5	6	5	0	0
3	1	1	1	4	4	6	6	6	6	6	6	6	6	6	5	0	0
4	4	1	1	1	1	6	6	6	6	6	6	6	6	6	6	0	0
5	1	1	1	1	1	6	6	6	6	6	6	6	6	6	6	0	0

B. HMM Learning Model

Thai finger-spelling derives from the hand posture of American finger-spelling. For example, “ก (ko kai)” use “K” (K), “ข (kho khai)” use a combination of “K” and “1” (K+1). The 42 hand posture sequences for each Thai letter finger-spelling can be shown in Figure 3.29 and Table.3.7. For this recognition step, a learning-based approach such as the Hidden Markov Model (HMM) [11] is used. The HMM is a stochastic processes which can be used to model any time series data. Therefore, HMM is useful for recognition of Thai finger-spelling, which can be viewed as a series of hand postures in American finger-spelling. The basic elements of HMM can be expressed as follows:

1). A set of learning states:

$$S = \{S_1, S_2, \ldots, S_N\}$$

The number of states is determined by using the maximum number of hand posture involved in performing a Thai finger-spelling (e.g. “ธ” (tho thong) in Table 3.7 use three hand posture) which is three states for hidden states and additional two states for initialization and finalization. Therefore, a five-state model with transitions was chosen for the system (N=5).

2). A set of observation symbols:

$$V = \{V_1, V_2, \ldots V_M\}$$

The observation symbols are represented by the thirty-one hand postures in American finger-spelling (M=31). All observation symbols are listed in Table.3.6. The hand posture sequence will be converted to an observation symbols sequence for using as input to the learning model. For instance, the “ก (ko kai)” alphabet uses “K”. This sequence is converted to symbol “11”. The “ข (kho khai)” alphabet use “K+1”. Thus, the sequence is converted to sequence symbol of “11 and 27” and so on.

3). The state transition matrix A = $\{a_{ij}\}$, where a_{ij} is the transition probability of taking the transition from state i to state j.

$$a_{ij} = P(q_t = S_j \mid q_{t-1} = S_i), \qquad 1 \le i, j \le N \tag{3.16}$$

$$A = \begin{bmatrix} 0.2 & 0.3 & 0.2 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.4 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.1 & 0.3 & 0.1 & 0.2 & 0.3 \end{bmatrix} \tag{3.17}$$

4). The observation symbol probability distribution matrix B = $\{b_j(k)\}$, where $b_j(k)$ give the probability of emitting observation symbol o_k from state j.

$$b_j(k) = P(o_t = V_k \mid q_t = S_j), \qquad 1 \le j \le N; \quad 1 \le k \le M \tag{3.18}$$

$$B=\left[\begin{array}{cccccccccccccccccccccccccccccccc}
0.03 & 0.05 & 0.03 & 0.05 \\
0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.05 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.05 \\
0.03 & 0.03 & 0.03 & 0.05 & 0.03 & 0.05 \\
0.03 & 0.03 & 0.03 & 0.03 & 0.05 & 0.03 & 0.05 \\
0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.05 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.03 & 0.05
\end{array}\right] \tag{3.19}$$

5). The initial state distribution matrix $\pi = \{\pi_i\}$, where π_i is initial probability of all states in the model.

$$\pi_i = P(q_1 = S_i), \qquad 1 \le i \le N \tag{3.20}$$

$$\pi_i = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix} \tag{3.21}$$

6). The HMM topology, Fully Connected topology, is applied for our system. In this topology, every state can be reached from any state in a finite number of steps as shown in Figure 3.27.

For training process, the HMM models the 42 Thai alphabet finger-spellings (λ_k) by the training data input. For recognition, a combination of hand posture in the American finger-spelling system are converted to be observation symbol sequence ($O = V_1, V_2, .., V_T$) and input to each HMM model for calculating the probability $P(O|\lambda_k)$. The models that give the maximum probability will be recognized as result of the input observation symbol sequence. The recognition process is shown in Figure 3.28.

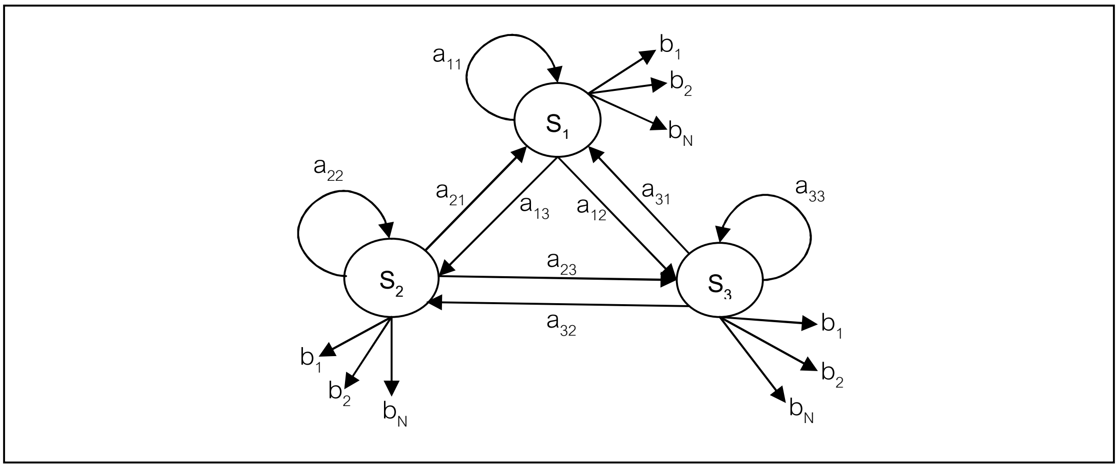


Figure 3.27 Fully connected HMM topology.

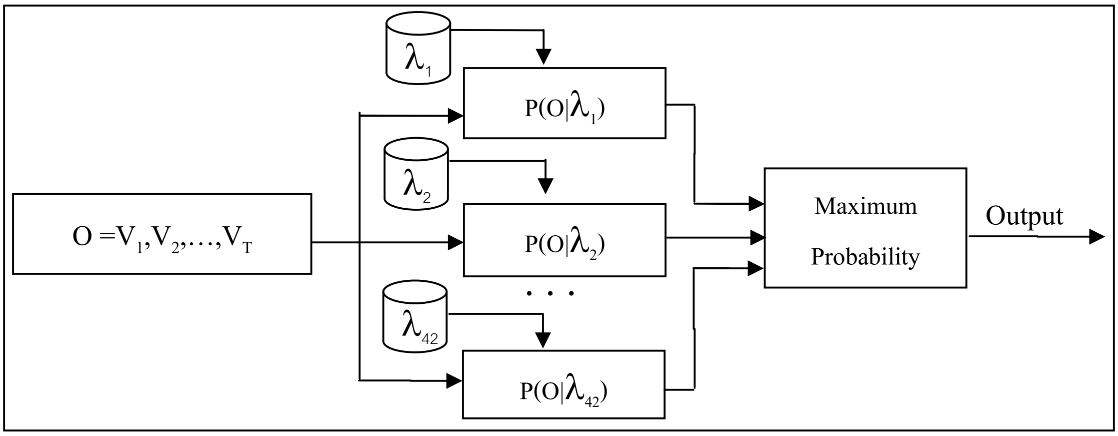


Figure 3.28 HMM recognition model process.

Table.3.6 Observation symbols.

Hand posture	**Observation symbol**	**Hand posture**	**Observation symbol**
A	1	Q	17
B	2	R	18
C	3	S	19
D	4	T	20
E	5	U	21
F	6	V	22
G	7	W	23
H	8	X	24

Hand posture	Observation symbol	Hand posture	Observation symbol
I	9	Y	25
J	10	Z	26
K	11	1	27
L	12	2	28
M	13	3	29
N	14	4	30
O	15	5	31
P	16	Undistinguished	32

Table 3.7 42 Thai alphabets finger-spelling.

Alphabet	Posture	Symbol	Alphabet	Posture	Symbol
ก (ko kai)	K	11	ว (wo waen)	W	23
ข (kho khai)	K+1	11, 27	ด (do dek)	D	4
ค (kho khwai)	K+2	11, 28	ฎ (do cha-da)	D+1	4, 27
ฆ (kho ra-khang)	K+3	11, 29	ฟ (fo fan)	F	6
ต (to tao)	T	20	ฝ (fo fa)	F+1	6, 27
ถ (tho thung)	T+1	20, 27	ล (lo ling)	L	12
ฐ (tho than)	T+2	20, 28	ฬ (lo chu-la)	L+1	12, 27
ฒ (tho phu-thao)	T+3	20, 29	จ (cho chan)	J	10
ฑ (tho montho)	T+4	20, 30	ย (yo yak)	Y	25
ฏ (to pa-tak)	T+5	20, 31	ญ (yo ying)	Y+1	25, 27
ส (so suea)	S	19	ม (mo ma)	M	13
ศ (so sala)	S+1	19, 27	น (no nu)	N	14
ษ (so rue-si)	S+2	19, 28	ณ (no nen)	N+1	14, 27
ซ (so so)	S+P	19, 16	ง (ngo ngu)	N+G	14, 7

Alphabet	Posture	Symbol	Alphabet	Posture	Symbol
พ (pho phan)	p	16	ท (tho thahan)	T+H	20, 8
ป (po pla)	P+1	16, 27	ธ (tho thong)	T+H+1	20, 8, 27
ผ (pho phueng)	P+2	16, 28	ฉ (cho ching)	C+H	3, 8
ภ (pho sam-phao)	P+3	16, 29	ช (cho chang)	C+H+1	3, 8, 27
ห (ho hip)	H	8	ฌ (cho choe)	C+H+2	3, 8, 28
ฮ (ho nok-huk)	H+1	8, 27	อ (o ang)	A	1
บ (bo baimai)	B	2	Undistinguished	Undistinguished	32
ร (ro ruea)	R	18	-	-	-

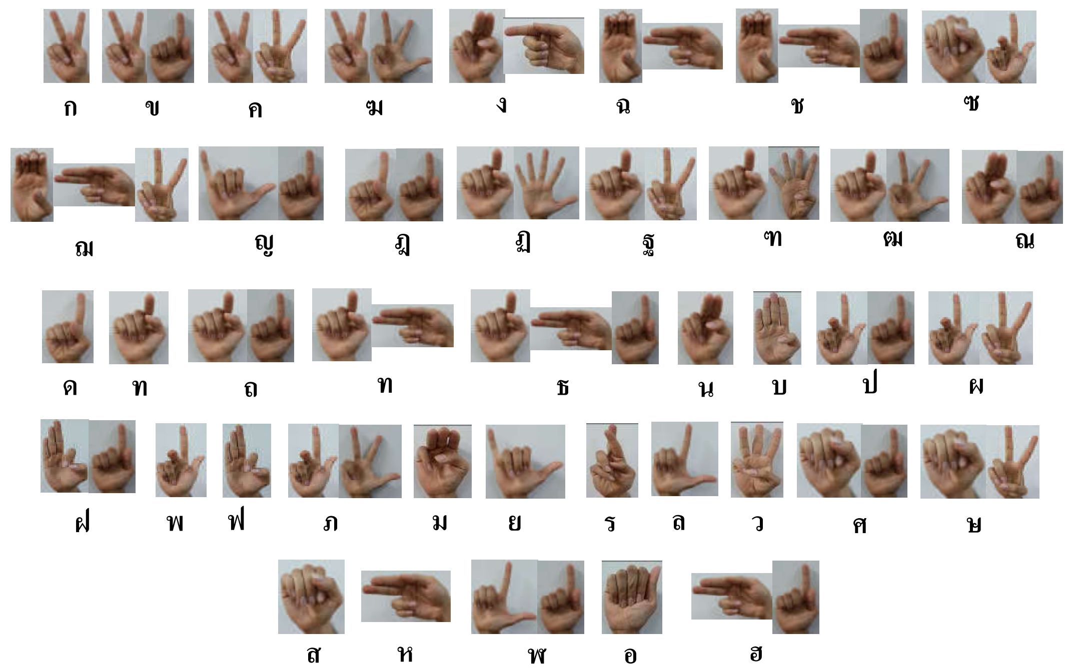


Figure 3.29 The 42 Thai letters finger-spelling hand postures.

3.3 Summary

In this chapter, we introduced a method for Thai alphabets finger-spelling recognition system based on a hand posture estimation method that uses fuzzy finger shape and hand appearance features. By using depth image, the hand region is extracted, and initial features such as fingertip, center point, and palm size have been calculated. The concept of active contour which uses an energy function is implemented, to track a fingertip's position in the frame image sequence. To discriminate the difference between a pair of elements in the hand posture sets, hand feature definitions have been established which are composed of fuzzy finger shapes and hand appearance features. The features have been defined as chain codes to represent hand posture. Since we focus on Thai alphabet finger-spelling, this recognition process will be performed in two steps. First, American finger-spelling is recognized by the chain code feature using scored voting method. Second, the Hidden Markov Model (HMM) method is used to build learning models to recognize sequences of American finger-spelling hand postures, to render Thai alphabet finger-spelling.

CHAPTER 4

EXPERIMENT RESULTS

In order to measure the performance of the system described in the previous section, three sets of experiments were conducted. For the first experiment, the performance of fingertips tracking using the active contour concept was evaluated. Since Thai alphabet finger-spelling is based on American finger-spelling, the two experiments for American as well as the single Thai alphabet finger-spelling will be evaluated in this chapter.

Implementation system:

- Software specification:

The proposed method was implemented in Opencv2.4.1 which runs with 30 frames per second on 320×240 depth image video sequences, and visual studio C++ 2008

- Hardware specification:

Our current computer hardware was an Intel® core(TM)2 Quad CPU RAM 4 GB, and Creative Senz3D depth camera. This web camera has a 720p HD depth map, where each pixel (the distance in millimeters from an object to the camera's X-Y plane, or Cartesian depth) is a 16-bit integer.

4.1 Datasets

The video sequence that user dose various hand posture is capture from 320x240 pixels depth image using depth sensor camera which used as input data to all experiments.

4.1.1 Dataset for fingertip tracking

The testing sequences of fingertip tracking consist of: bending finger, moving finger into the palm, crossing finger, hand movement (up, down, left and right)and 45^0 hand rotation (counterclockwise and clockwise). Each sequence is tested for 10 rounds, amounting to 50 sequences in total.

4.1.2 Dataset for American finger spelling

Hand postures for the American finger-spelling were collected, which consist of the letters from “A” to “Z” and hand postures of the numbers 1, 2, 3, 4 and 5 (31 hand postures). Each posture has 100 examples. The total examples make up 3,100 data point, which were tested with per-defined hand chain codes.

4.1.3 Dataset for Thai finger spelling

The 150 samples of each hand posture (“ก” (Ko kai) to “ฮ” (ho nok-huk); 42 letters) were collected, gathering 6,300 samples in total. For each hand posture data set, 50 of the samples were used to generate each alphabet model and the other 100 samples were used to test the performance of the recognition model.

4.2 Fingertips tracking precision

We evaluated the fingertip tracking precision on basic finger movements between the tracked fingertips and the ground truth. We defined the ground truth using the end point contour of each finger. The basic finger movements, Figure 4.1 to 4.5, include five sequences. Bending finger is sequence 1. Moving finger into the palm is sequence 2. Crossing finger is sequence 3. Hand movement (up, down, left and right) is sequence 4. Hand rotation (45^0 counterclockwise and clockwise) is sequence 5. Each sequence was tested for 10 rounds. Table 4.1 shows the precision in terms of the Euclidean distance.

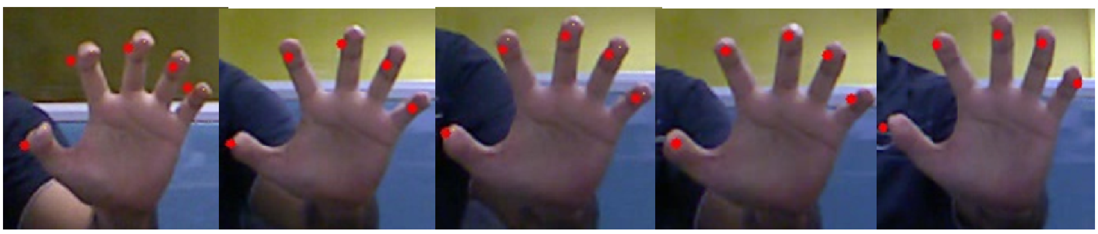


Figure 4.1 Bending finger sequence.

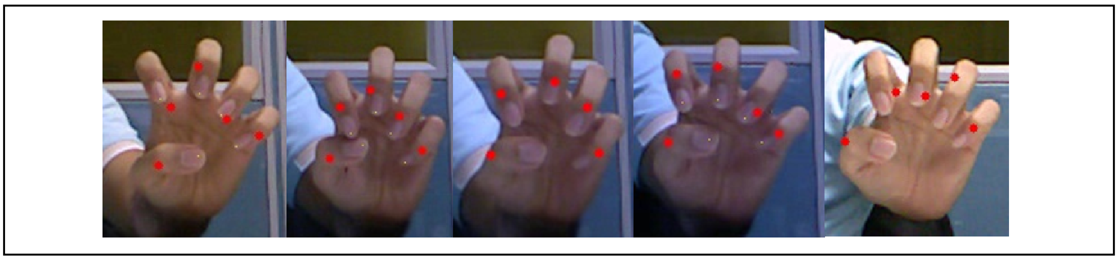


Figure 4.2 Moving finger into palm sequence.

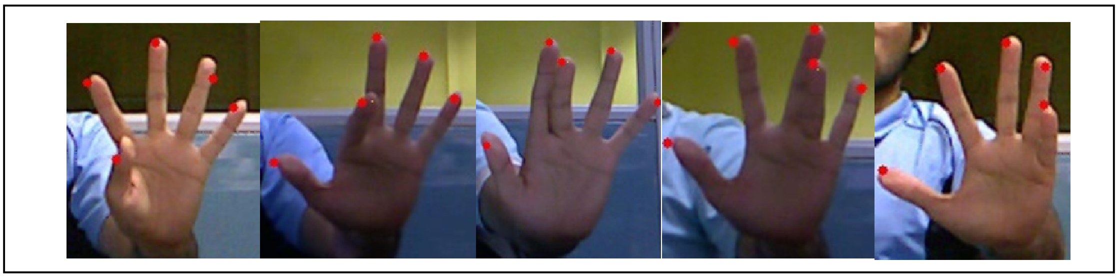


Figure 4.3 Crossing finger sequence.

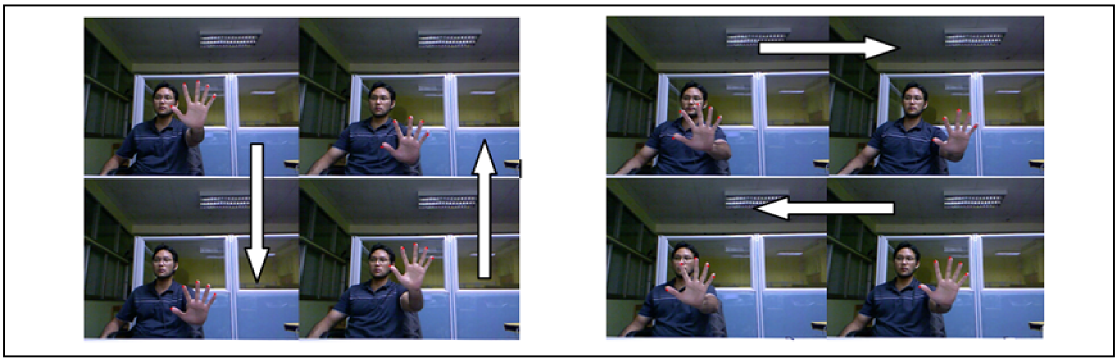


Figure 4.4 hand movement sequence.

Figure 4.5 Hand rotation sequence (45^0 degree).

Table.4.1. Fingertips tracking precision.

Seq.	Tracking Precision(Pixel)					Avg.
	Thumb	Index	Middle	Ring	Pinky	
1	7.94	7.65	8.50	9.25	8.51	8.37
2	25.35	14.28	14.96	23.12	15.69	18.68
3	7.47	6.06	5.12	5.60	5.10	5.87
4	5.60	4.16	3.06	5.05	5.77	4.72
5	4.17	3.44	5.80	3.76	7.19	4.87

From the experimentation results, we found that the tracking cases for crossing (sequence 3), rotation (sequence 4) and movement (sequence 5) give a good precision of about 4 to 6 pixels of error, because the candidate fingertips obtained the from polygon approximation algorithm are quite well located, to the fingertip ground truth. But the tracking cases for bending (sequence 1) and moving into palm (sequence 2) give a lower precision of about 9 to 20 pixels of error, due to the imperfect depth data received from sensor which was not enough level for near mode function. Hence, the depth data of the finger may have been lost in some frames. These factors may have produced some errors in the fingertip tracking process.

4.3 American finger-spelling recognition

Each hand posture for the American finger-spelling is tested for 100 rounds. The result of this recognition testing is shown in Table 4.2. The best cases of the system, 100% recognition rate, are the hand postures of the numbers “3”, “4” and “5”. The hand postures of these three cases are not much different from the initial hand posture of open hand. The movement of the finger does not change much. Thus, the finger tracking error will occur less. The worst case which has a less than 10% recognition rate is the hand posture of “Z” letter. This hand posture has rapid hand movement. Thus, the finger tracking error will occur more when the hand posture has an immediate change-direction movement. For the general cases about a 65% recognition rate for hand postures of “A” to “Y” and “1” to “2”, most of the hand postures have only a continuous finger movement, which does not change much from the current frame to the next frame. Hence finger tracking using the active contour concept can be quite effectively performed. However, there is misrecognition, which is shown in Table 4.2. For example, hand posture "S" can be erroneously recognized as hand posture "A" instead due to the similarity of both hand postures. Error may also occur if more than one model has a maximum score, and if score does not pass the threshold, which hand posture will be rejected since the process cannot distinguish the two hand postures. Example of this process can be shown in Figure 4.6 and at [37].

Figure 4.6 American finger-spelling recognition processes (2).

Figure 4.6 American finger-spelling recognition processes (2).

Table 4.2 American finger-spelling recognition result (confusion matrix).

	Result														
Hand Gesture	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A	69														23
B		76													
C	6		62												
D				68											
E	7				56										9
F						67									
G							62								
H								64							
I									75						
J										61					
K											63				
L												72			
M													67		3
N	6													64	2
O	13														55
P												15			
Q	4														8
R															
S	5														2
T															
U															
V											20				
W															
X														8	
Y										20					
Z															
1				20											
2											10				
3															
4															
5															

	Result																
Hand Gesture	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	undistiquish
A																	8
B																	24
C									14								18
D												23					9
E				13													15
F																	33
G																	38
H																	25
I										19							4
J																	39
K													17	8			12
L	17																11
M																	30
N																	28
O																	32
P	71													4			10
Q		64			12												12
R			68			18											14
S				58													35
T					68												32
U			10			70											20
V							65						5	2			8
W								78									22
X									69								23
Y										73							7
Z											8						92
1												72					8
2							20						68	2			-
3														100			-
4															100		-
5																100	-

4.4 Thai alphabet finger-spelling recognition

The Forward-Backward procedure (HMM) is used to calculate the probability of an input observation sequence. Since HMM does not have a fixed rule for the state number specification, we compare the recognition rate of the HMM process when it use 5 states, 10 states, and the 15 states. Table 4.3 shows the results of the 42 Thai letters recognition model. The aggregate recognition rate of 69.52% for 10 states of HMM is better than the aggregate recognition rate of 68.90% for 5 states of HMM, but the improvement (+0.62%) is marginal, not significant. 15 states of HMM give an even lower rate of 65.88%. We chose to use five states of HMM , give an average recognition rate at 68.90%.

Table 4.3 Thai alphabets finger-spelling recognition rate.

Alphabet	Recognition rate (%)			Alphabet	Recognition rate (%)		
	5 state	10 state	15 state		5 state	10 state	15 state
ก (Ko kai)	82	84	79	ธ (tho thong)	57	61	62
ข (kho khai)	72	75	71	น (no nu)	63	62	65
ค (kho khwai)	74	72	69	บ (bo baimai)	84	86	81
ฆ (kho ra-khang)	75	77	70	ป (po pla)	76	74	73
ง (ngo ngu)	59	62	56	ผ (pho phueng)	73	75	71
จ (cho ching)	73	74	71	ฝ (fo fa)	72	76	70
ฉ (cho ching)	70	69	67	พ (pho phan)	77	78	75
ช (cho chang)	68	69	65	ฟ (fo fan)	78	76	74
ซ (so so)	52	51	45	ภ (pho sam-phao)	74	75	72
ฌ (cho choe)	69	72	67	ม (mo ma)	64	61	62
ญ (yo ying)	77	75	69	ย (yo yak)	75	76	74
ฎ (do cha-da)	74	75	65	ร (ro ruea)	72	71	68
ฏ (to pa-tak)	57	59	61	ล (lo ling)	75	77	74
ฐ (tho than)	65	67	63	ว (wo waen)	72	71	70
ฑ (tho montho)	62	63	57	ศ (so sala)	54	57	46
ฒ (tho phu-	64	62	60	ษ (so rue-	52	51	43

Alphabet	Recognition rate (%)			Alphabet	Recognition rate (%)		
	5 state	10 state	15 state		5 state	10 state	15 state
thao)				si)			
ณ (no nen)	68	70	65	ส (so suea)	55	53	51
ด (do dek)	80	75	71	ห (ho hip)	73	76	71
ต (to tao)	61	59	55	ฬ (lo chu-la)	74	75	72
ถ (tho thung)	58	56	57	อ (o ang)	75	77	73
ท (tho thahan)	65	69	64	ฮ (ho nok-huk)	72	73	70
Avg. HMM 5 state = 68.904%, HMM 10 state = 69.528%, HMM 15 state = 65.880%							

As a result, we infer that the alphabet models that use only one hand posture get a quite better result such as “ก” (ko kai), “ข” (cho chan), “ด” (do dek), “บ” (bo baimai), “พ” (po han), “ฟ” (fo fan), “ย” (yo yak), “ร” (ro ruea), “ล” (lo ling), “ว” (wo waen), “ห” (ho hip), “อ” (o ang). There are some exceptions, such as “ต” (to tao) or “T”, “ม” (mo ma) or “M”, “น” (no nue) or “N” and “ส” (so suea) or “S”, which get a poor result (less than 70%) because the fingertip tracking cases for finger overlapping and for adjacency give less precision as show in Figure 4.7. Hence, the other letters that are based on these letter groups (“T”, “M”, “N” and “S”) will also give a poor result. These are groups of letters which are based on “N”, such as “ง” (ngo ngu) and “ณ” (no nen), the groups of alphabets based on “S” such as ศ” (so sala) and “ษ” (so rue-si), and groups of letters based on “T” such as “ฏ” (to pa-tak), “ฐ” (tho than), “ฑ” (tho montho), “ฒ” (tho phu-thao), “ถ” (tho thung), “ท” (tho thahan) and “ธ” (tho thong). The same is true of letters which are based on “C”, for example “ฉ” (cho ching) or “C+H”, ช (cho chang) or “C+H+1” and ฌ (cho choe) or “C+H+2”. These groups of letters have the same problem as mentioned above. HMM with many states can have an over-fitting model problem. An over-fitting model generally occurs when a model is excessively complex, such as having too many training cycles, or parameters relative to the number of observations. The model begins to memorize training data rather than learning to generalize from some trend. So its performance is good on the training examples while the performance on unseen data becomes worse. Therefore, the 15 states of HMM produce quite inferior results compared with HMMs that have fewer training cycles. Table 4.4, 4.5, and 4.6 show the detailed result of each of various state recognitions. The real-time application example can be shown in Figure 4.8 and at [38].

(a) (b) (c) (d) (e)

Figure 4.7 Hand postures: (a) “T” (b) “N” (c) “M” (d) “S” (e) “C”.

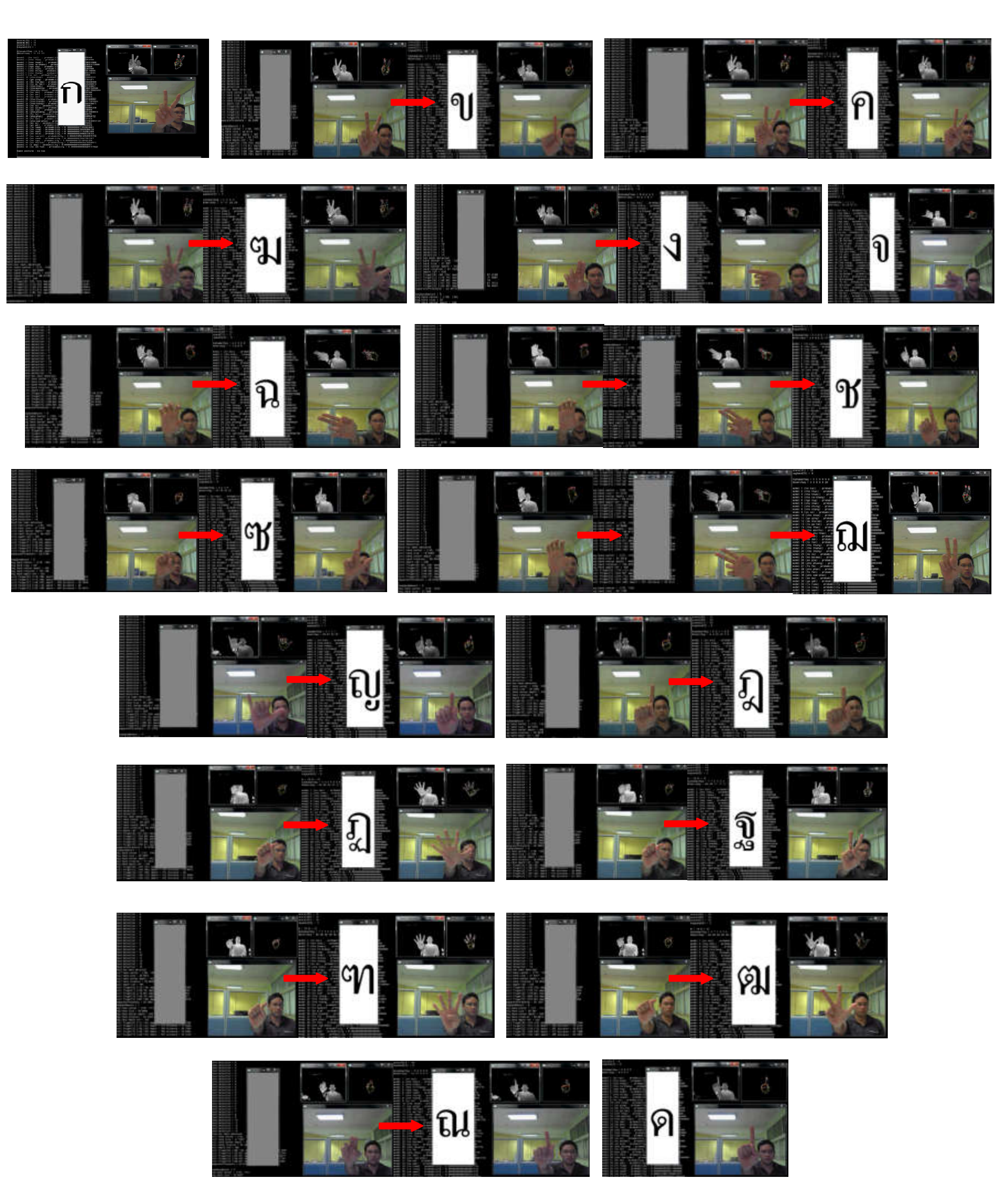


Figure 4.8 Example of Thai alphabets finger-spelling recognition (1).

Figure 4.8 Example of Thai alphabets finger-spelling recognition (2).

Table 4.4. HMM 5 stages confusion matrix (1).

Hand Posture \ Result	ก	ข	ค	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท
ก	82	2	3	4																	
ข	5	72	1									1						4			
ค	12	2	74											1							
ฆ	6	1		75												3					
ง					69																
จ						73					2										
ฉ							72	3		2											
ช							3	68		1											
ซ									52												
ฌ							4	2		69											
ญ						4					77						2				
ฎ		3										74						8		2	
ฏ													57	2		3			9		
ฐ			1											65		1			7		
ฑ															62	1			12	2	
ฒ				5												64			4	2	
ณ																	68	1			
ด		4																80		3	
ต																			61	1	
ถ		1															2		7	58	
ท																			5	1	65
ธ																			3	1	6
น					1												3				
บ															7						
ป		3															3			1	
ผ			3											2							
ฝ		2																		1	
พ									1												
ฟ																					
ภ				3																	
ม																					
ย						7					6										
ร	8	1	3																		
ล																					
ว															4						
ศ		2																			
ษ			5																		
ส																					
ห																					
ฬ																					
อ																			1		
ฮ		1																		3	

Table 4.4. HMM 5 stages confusion matrix (2).

Hand Posture	Result																					
	ธ	น	บ	ป	ผ	ฝ	พ	ฟ	ภ	ม	ย	ร	ล	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	error
ก																						9
ข																					17	17
ค																						11
ฆ									1													14
ง		6															5			3		27
จ											3									4		18
ฉ																				2	5	16
ช																					2	26
ซ							3								3		8					34
ฌ																		3				22
ญ											11						1					5
ฎ				1																		12
ฏ																						29
ฐ																						26
ฑ																						23
ฒ																						25
ณ		7																		3		21
ด																			3		2	8
ต																						38
ถ																						32
ท																						29
ธ	57																					33
น		63																		6		27
บ			84																			9
ป				76											2				1		1	13
ผ					73		3															19
ฝ				1		72		9														15
พ					2		77		2				5									13
ฟ			8			6		78														14
ภ				2			6		74													17
ม										64							2			3		31
ย											75											12
ร												72										16
ล				1			11						75						4			9
ว			6											72								18
ศ				1											54					4		38
ษ																52	3					40
ส															2	1	55					42
ห																		73			5	22
ฬ				5			1						8						74			12
อ																	3			75		21
ฮ																		5	1		72	18

Table 4.5. HMM 10 stages confusion matrix (1).

	Result																				
Hand Posture	ก	ข	ค	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท
ก	84		3	4																	
ข	3	75										2						3			
ค	9		72											4							
ฆ	5			77												3					
ง					62												1				
จ						74					1										
ฉ							69	7													
ช							3	69													
ซ									51												
ฌ							3			72											
ญ						4					75						2			1	
ฎ		2										75						10		1	
ฏ													59	1	1				7	3	
ฐ			1											67					6		
ฑ														1	63				12	1	
ฒ				4												62			4	3	
ณ																	70				
ด		4										5					1	75		2	
ต													2						59	1	
ถ		1															1	9		56	
ท														1					1		69
ธ																			2	4	
น					1										5		4				
บ																					
ป		4																		3	
ผ			4											1							
ฝ		1																			
พ																					
ฟ																					
ภ				1																	
ม																					
ย						7					5										
ร	8	2	3																		
ล															3						
ว																					
ศ																					
ษ			5											1						1	
ส										1											
ห																					
ฬ																					
อ																			1		
ฮ																				3	

Table 4.5. HMM 10 stages confusion matrix (2).

Hand Posture \ Result	ธ	น	บ	ป	ผ	ฝ	พ	ฟ	ภ	ม	ย	ร	ล	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	error
ก																						9
ข																						17
ค					1											3						11
ฆ									1													14
ง		2															5			3		27
จ												4								4		18
ฉ																				2	6	16
ช																					2	26
ซ							3								1	2	8					34
ฌ																		3				22
ญ												12							1			5
ฎ																						12
ฏ																						29
ฐ																						26
ฑ																						23
ฒ									2													25
ณ		6																		3		21
ด																			3		2	8
ต																						38
ถ																				1		32
ท																						29
ธ	61																					33
น		62																		6		27
บ			68																			9
ป				74											1						2	13
ผ					75		1															19
ฝ						76		7	3						1							15
พ				1			78						5									13
ฟ			8			2		76														14
ภ				2	2		5		75													17
ม										61							2			6		31
ย											76											12
ร												71										16
ล							11						77						2			9
ว			6											71								18
ศ															57					4		38
ษ																51	3					40
ส															4		53					42
ห																		76			2	22
ฬ				2			2						9						75			12
อ																	1			77		21
ฮ				1														5			73	18

Table 4.6. HMM 15 stages confusion matrix (1).

Result

Hand Posture	ก	ข	ค	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท
ก	79	7	2	3																	
ข	6	71	1															4			
ค	16		69											2							
ฆ	11		1	70												2					
ง					56												4				
จ						71					2										
ฉ							67	3		4											
ช							4	65		2											
ซ									45												
ฌ							4	3		67											
ญ						4					69						2				
ฎ		5										65						12		4	
ฏ													61			3			5	2	
ฐ													1	63					10		
ฑ														1	57				18		
ฒ				4										2		60			6	2	
ณ													2				65				
ด		12										3						71		6	
ต														1		1			55	2	
ถ		1										1							9	57	
ท																	1		3	2	64
ธ																				1	4
น																					
บ															10		2			5	
ป		3																			
ผ			6											1			3				
ฝ		1																			
พ									1												
ฟ																					
ภ				5																	
ม																					
ย						9					5										
ร	12	2	2																		
ล																					
ว															4						
ศ		5							3												
ษ			7											1							
ส									1												
ห																					
ฬ																					
อ																			4		
ฮ													1							1	

Table 4.6. HMM 15 stages confusion matrix (2).

Hand Posture \ Result	ธ	น	บ	ป	ผ	ฝ	พ	ฟ	ภ	ม	ย	ร	ล	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	error
ก																						9
ข																						17
ค					1											1						11
ฆ									2													14
ง		5															5			3		27
จ											5									4		18
ฉ																		2		2	6	16
ช																		1			2	26
ซ							3								2	4	12					34
ฌ																		3			1	22
ญ											18								2			5
ฎ				1		1																12
ฏ																						29
ฐ																1						26
ฑ																						23
ฒ									1													25
ณ		9		2																3		21
ด																					1	8
ต																						38
ถ																						32
ท	1																					29
ธ	62																					33
น		65																		6		27
บ			81																3			9
ป				73												1						13
ผ					71		2															19
ฝ				2		70		12														15
พ				3	2		75		1				5									13
ฟ			8			4		74														14
ภ							6		72													17
ม										62							2			5		31
ย											74											12
ร												68										16
ล				1	1		11						74						3			9
ว			8											70								18
ศ															46					6		38
ษ																43	9					40
ส															3	3	51					42
ห																		71			7	22
ฬ				3			1						12						72			12
อ																	2			73		21
ฮ																		9	1		70	18

Actually, there is much research related to the Thai sign language, word-level and finger-spelling, which we have discussed in the introduction. However, our work has focused only on the finger-spelling. Therefore, a general comparison can be performed between previous methods and our system, as show in Table 4.7.

Table 4.7 The general comparison.

work	Method	Device	Background	Outfit	Real time	No. alphabet	Recognition rate
[25]	G-B*	Sensor glove	No	No	Yes	16	94.44%
[29]	V-B*	No	Yes	No	No	15	72%
[30]	V-B*	Color glove	Yes	No	No	31	88.26%
[9]	V-B*	No	Yes	No	Yes	31	72.26%
[34]	V-B*	No	Yes	No	No	N/A*	72%
[3]	V-B*	No	Yes	No	No	42	81.43%
[23]	V-B*	No	Yes	Yes	N/A	15	79.90%
Our method	V-B*	Depth camera	No	No	Yes	42	68.90%
*G-B = Glove-Based, *V-B = Vision-Based *N/A = Not Available							

We have compared the general conditions not only for our method but also for some research that use other additional devices such as sensor glove, colour glove or depth sensor camera. Regarding the background, some researchers have to set it up to be a constant colour. In terms of the outfit, users are asked to wear long sleeves shirts. Concerning the number of the letters that can be recognized in the system, since we propose a vision-based method for hand posture estimation, the comparison result shows that it is hard to compare our system with a glove-based method as configured in [25], because the depth image is not as good as a signal from an electronic sensor, especially when fingers occlude or stick together, as we mentioned earlier. Although many proposed Thai finger-spelling recognition systems [3,9,23,29,30,34] are vision-based approaches, most of them use only appearance features, such as point, contour, edge etc. In contrast, our proposed system considers features of the hand posture that includes the finger shape, finger relation, hand movement and hand rotation features, which increases the capacity for hand posture discrimination. In addition, we do not need to segment the key frame before performing the letter recognition. Therefore, our process can be run in real-time situations. For the recognition rate, although our work does not yield a more significant result compare to other techniques, we do propose a method that recognizes all Thai alphabet finger-spelling. Meanwhile the other systems provide only some groups of the Thai alphabets recognition.

4.5 Summary

In this section, we examine three experiments which can be integrated into the proposed system of: fingertips tracking, American finger spelling and Thai alphabets finger-spelling. The fingertip tracking experiment is to locate the user's fingertip location. The fingertip tracking precision has been evaluated by comparison between the tracked fingertips using the active contour concept, and the ground truth which was defined at the end point contour of each finger. The basic five finger movement sequences have been performed: bending finger, moving finger into the palm, crossing finger, hand movement (up, down, left and right) and 45^0 hand rotation (counterclockwise and clockwise). From experimentation, we found that the tracking cases for crossing, rotation and movement give a good precision, because the candidate fingertips obtained from polygon approximation algorithm are quite well located, compared to the fingertip ground truth. But the tracking cases for bending and moving into palm give less precision, due to the imperfect depth data. Hence, the depth data of a finger may be lost in some frames. These factors might produce some error in the fingertip tracking process, because Thai alphabet finger-spelling are based on American finger-spelling. Then, performance for American finger-spelling was estimated. Score voting was used, which compares input feature sequences with pre-defined template sequences for representing American finger-spelling hand posture. The best cases of the system are the hand postures of the numbers “3”, “4” and “5”. The hand postures of these three cases are not much different from the initial hand posture, open hand. The worst case is the hand posture of the “Z” letter. This hand posture has rapid hand movement. Thus, the finger tracking errors will occur more when the hand posture has an immediate change-direction movement. For the general cases, hand postures of “A” to “Y” and “1 and 2”, most of the hand postures have only a continuous finger movement, which does not change much from the current frame to the next frame. Then, sequences of American finger-spelling will be recognized in Thai alphabet finger-spelling. The learning based model (HMM) has been used to build a recognition model for each Thai letter. The input sequence data is fed to each model for calculating probability. The models that give the maximum probability will be recognized as result of this input sequence. As the result, we infer that the alphabet models that use only one hand posture get a quite better result. However, there are some exceptions, such as the groups of alphabet which based on “T”, “M”, “N”, “S” and “C”, since the fingertip tracking cases for finger overlapping and adjacency give less precision. HMM with many states can cause an over-fitting model problem. An over-fitting model generally occurs when a model is excessively complex, such as having too many training cycles. The model begins to memorize training data, and its performance is good on the training examples but the performance on unseen data becomes worse. As a result, the HMM with many states produces quite inferior results, compared with HMM that have fewer training cycles. The aggregate recognition rate of 69.52% for 10 states of HMM is better than the aggregate recognition rate of 68.90% for 5 states of HMM, but the improvement (+0.62%) is marginal, not significant. 15 states of HMM give an even lower rate of 65.88%. We chose to use five states of HMM, give an average recognition rate at 68.90%.

CHAPTER 5

CONCLUSION

In this thesis, we presented a method that enables the estimation of hand posture for the Thai alphabet finger-spelling recognition system. The depth images are used for robust hand region segmentation, and for removing the complex background. The active contour concept is applied in order to calculate the energy function, and to track the fingertip's position in the frame sequence. The finger shapes and hand appearance features have been proposed to represent the different hand posture sets. The features of finger shapes are designed based on fuzzy logic. The hand appearance features consist of finger relation, hand rotation and hand movement. Since the Thai alphabet finger-spelling is based on the American finger-spelling, the recognition model will be performed in two steps. The first step is the American finger-spelling: each hand posture will be encoded into discrete chain codes, using finger shapes and hand appearance. The 31 templates of hand posture chain codes for the American finger-spelling are pre-defined, and are used to compare the similarities between pre-defined chain codes and input chain codes, by using simple score voting. The second step for Thai alphabet finger-spelling, the Hidden Markov Model, is used to build 42 Thai alphabet learning models that recognize the sequences of the American finger-spelling hand postures and interpret these more specifically to the Thai alphabet finger-spelling. The alphabet hand posture recognition result is performed with five states of HMM and provides an average recognition rate at 68.90%.

5.1 Further Improvements

In this section, we proposed some possible improvements in the further works. From the experiments, we found that the technique for fingertips tracking, which is an important process for locating finger position, could be better improved. Another possible improvement is an HMM training process for the learning model of hand posture classification. If these techniques are improved, then the error should be decreased and greatly impact on the final performance of the system.

5.1.1 Fingertips tracking

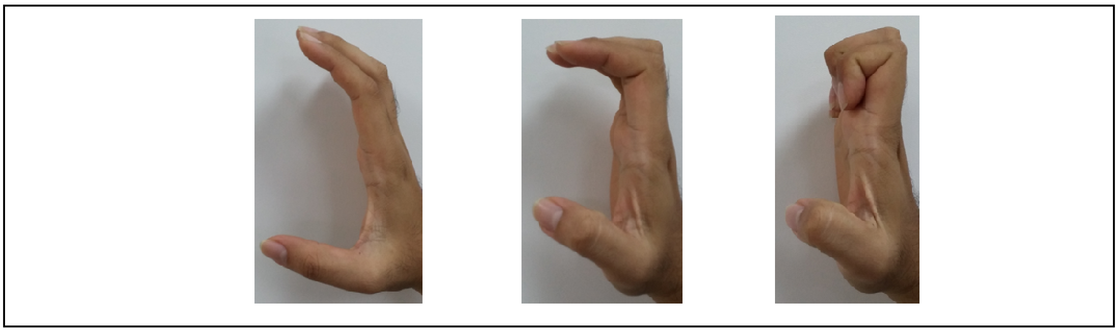
The method for fingertip tracking described in Section 2.2.2 is one of our main processes for the system. This procedure will locate the positions of fingertips, which are used for hand posture discrimination in each pattern. Therefore, the precision of fingertip position will directly effect to the performance of system. The tracking computation must be processed for the thumb first. After that, it is performed on the index, middle, ring and pinky fingers, respectively. The computational times for each finger are summed together. It may take much time to compute one frame. Because of this, discontinuity of frame sequence acquisition could happen. One among many solutions to improve this situation is to apply parallel computing. Parallel computing uses multiple independent threads to execute separately computational units without interfering with each other. The computation of every fingertip position could be performed at the same time to increase the speed and eventually the accuracy of fingertip positions.

Another possibility of accuracy improvement is to apply the post-tracking process as a low-pass filter using the conventional techniques such as Kalman filtering and particle filtering.

5.1.2 Hand feature definition

The hand features are used to discriminate the hand posture in each pattern, which consists of the finger shapes and hand appearance. The four basic patterns of finger shape are defined as “Open”, “Close”, “Bending” and “Point”. From our experiments we noticed that some additional patterns may improve the accuracy of system, due to the lack of continuity of patterns: for example, “Half-Bending”, “Half-Close” or “Half-Point”, as shown in Figure 5.1. Because of the limited dataset for the hand posture features used in our experimentation, the system is quite sensitive to the different characteristics of hand features, such as the length of fingers. So, our system needs to be characterized with more data. Previously, fuzzy logic could not classify finger shape correctly, because it provides a static mathematical model. Thus, the extended dynamic fuzzy sets were introduced.

For features of hand appearance, it is possible to add more features for detailing some physical characteristic of hand posture, such as looping, parallel relation and so on.



(a) (b) (c)

Figure 5.1 Additional finger shapes: (a) “Half-Bending” (b) “Half-Point” (c) “Half-Close”.

5.1.3 HMM training process

Among the machine learning methods, the Hidden Markov Model (HMM) is one of the most popular techniques. Its operation is described in the section 2.2.4. In our experiment, the fully connected HMM topology, where every state can be from every other state, has been used for the recognition. However, there are other topologies that could be investigated, such as: the linear HMM topology, where transitions are defined only to the consecutive states and to the current state itself; the Bakis HMM topology where transition are only allowed the two following states and no transition to states which are lower than current state, the left-to-right HMM topology which is similar to the Bakis model except the current state will be connected to all remaining states. These topologies could be applied to recognize the hand posture as well. Thus, a learning model should be trained with the different topologies and then the recognition rate should be compared. The HMM topologies are shown in Figure 5.2

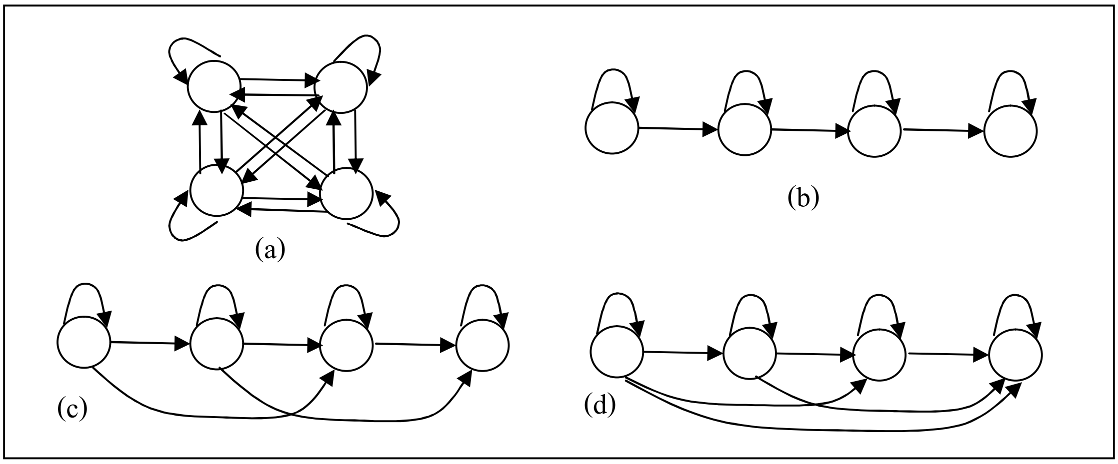


Figure 5.2 HMM topologies: (a) fully connected (b) linear (c) Bakis (d) left-to-right.

5.2 Summary

We presented a method that enables the estimation of the hand posture for a Thai alphabet finger-spelling recognition system. A depth image was used for robust hand region segmentation, and for removing the complex background. The active contour concept calculates the energy function to track the fingertip's position in the frame sequence. The finger shape and hand appearance features were proposed to represent different hand posture sets. The finger shape features are based on fuzzy logic. The hand appearance features consisted of finger relation, hand rotation and hand movement. Since Thai alphabet finger-spelling is based on American finger-spelling, therefore, the recognition model will be performed in two steps. For the American finger-spelling, each hand posture will be encoded to a discrete chain code based on finger shapes and hand configuration. The 31 template hand posture chain codes for the American finger-spelling (26 hand postures for A-Z and five hand postures for 1-5 numbers) were pre-defined and were used to compare the similarities between pre-defined chain codes and input chain codes by using simple score voting. For Thai alphabet finger-spelling, the learning-based method, Hidden Markov Model, is use to build 42 Thai letter models that recognize the sequence of the American finger-spelling hand postures and provide the Thai alphabet finger-spelling. The letter hand posture recognition result is performed with five states of HMM, and provides an average recognition rate of 68.90%. The method does not only apply to American finger-spelling or Thai alphabet finger-spelling. We expect that our method can be applied to other applications, as for example games, robot controlling or visual input devices. However, our main further work is to increase the speed and tracking accuracy of the fingertips. If the data of the fingertip's position is precise, then the error should be decreased and greatly impact on the final performance of the system.

REFERENCES

[1] A. Erol et al., Vision-based hand pose estimation: A review, Computer Vision and Image Understanding 108 (2007) 52-73.

[2] A. Park, S. Yun, J. Kim, S. Min and K. Jung, Real-Time Vision-based Korean Finger Spelling Recognition System, Engineering and Technology 34, 293-298**,** (2008)**.**

[3] B. Sakulsujirapa, M. Karnjanadecha, A. Choksuriwong, Thai Sign Language Recognition Based on Finger Pattern Analysis, In International Conference on Embedded Systems and Intelligent Technology (ICESIT2011), Phuket, Thailand, 9th - 11th February 2011, pages 16.

[4] D. J Williams, M. Shah, A Fast algorithm for active contours and curvature estimation, In: CVGIP: Image Understanding, vol. 55, no. 1, pp. 14-26 (January 1992)

[5] E. Cox, "Fuzzy fundamentals," Spectrum, IEEE, vol.29, no.10, pp.58-61, Oct. 1992.

[6] F. Isikdogan, S. Albayrak, Automatic recognition of Turkish fingerspelling, Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on , vol., no., pp.264-267, 15-18 June 2011.

[7] H. Cooper, B Holt, R. Bowden, "Sign Language Recognition", in Moeslund TB, Hilton A, Krüger V, Sigal L (eds.) Visual Analysis of Humans: Looking at People Springer Verlag, pp. 539-562.

[8] J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial," Proceeding of the IEEE, vol.83, no.3, pp.345-377, Mar 1995.

[9] J. Watthanatham, W. Jiratthitijaroen, T. Srinark, "ระบบจดจำตัวอักษรสำหรับการแปลตัวสะกดตัวอักษรภาษามือไทย", The 2nd ECTI-Conference on Application Research and Development (ECTI-CARD), Chonburi, Thailand, pages 572-577, May 10-12, 2010.

[10] K. Fujimura, L. Xia, Sign recognition using depth image streams, Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on, vol., no., pp.381-386, 2-6 April 2006.

[11] L.R Rabiner, "A tutorial on hidden markov models and selected application in speech recognition," Proc. IEEE 77, 1989, pp 267-293

[12] M. Kass, A. Witkin, and D. Terzopoulos: Snakes: Active contour models. In: International journal of computer vision, vol. 1, no. 4, pp. 321-331 (1988).

[13] M. Zhenyao, N. Ulrich, Real-time Hand Pose Recognition Using Low-Resolution Depth Images, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006 , vol.2, pp. 1499- 1505, 2006.

[14] P. Chanda, S. Auephanwiriyakul, N. Theera-Umpon, Thai sign language translation system using upright speed-up robust feature and dynamic time warping, Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on , vol.2, no., pp.70,74, 25-27 May 2012.

[15] P. Chanda, S. Auephanwiriyakul, N. Theera-Umpon, Thai sign language translation system using upright speed-up robust feature and c-means clustering, Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on , vol., no., pp.1,6, 10-15 June 2012.

[16] P. Goh, E.J. Holden, Dynamic Fingerspelling Recognition using Geometric and Motion Features, Image Processing, 2006 IEEE International Conference on , vol., no., pp.2741-2744, 8-11 Oct. 2006.

[17] R. Akmeliawati, et al., Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network, Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE, vol., no., pp.1-6, 1-3 May 2007.

[18] R. S. Feris, M. Turk, R. Raskar, T. Karhan; G. Ohashi, Exploiting Depth Discontinuities for Vision-Based Fingerspelling Recognition, Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on, vol., no., pp. 155, 27-02 June 2004.

[19] S. Liwicki, M. Everingham, “Automatic recognition of fingerspelled words in British Sign Language,” Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on, vol., no., pp.50-57, 20-25 June 2009.

[20] S. Min, et al., Simple Glove-Based Korean Finger Spelling Recognition System, Lecture Notes in Computer Science 4705, 1063. (2007).

[21] S.C.W. Ong, S. Ranganath, Automatic sign language analysis: a survey and the future beyond lexical meaning, Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.27, no.6, pp. 873- 891, Jun 2005.

[22] S. Auephanwiriyakul, et al., Thai sign language translation using Scale Invariant Feature Tranform and Hidden Markov Models, Pattern Recognition Letters 34(11): 1291-1298 (2013).

[23] S. Phitakwinai, S. Auephanwiriyakul, N. Theera-Umpon, Thai Sign Language Translation Using Fuzzy C-Means and Scale Invariant Feature Transform, Lecture Notes in Computer Science, Vol 5073, pp. 1107 – 1119, 2008.

[24] S. Ricco, C. Tomasi. Fingerspelling, Recognition through Classification of Letter-to-Letter Transitions, 9th Asian Conference on Computer Vision, pages 214--225, September 2009.

[25] S. Saengsri, V. Niennattrakul, C.A. Ratanamahatana, TFRS: Thai finger-spelling sign language recognition system, Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on , vol., no., pp.457,462, 16-18 May 2012.

[26] T. B. Dinh, et al., Hand gesture classification using boosted cascade of classifiers, Proceedings of the 4th IEEE International Conference on Research, Innovation and Vision for the Future, RIVF'06, p. 139-144, 2006.

[27] T. Ditcharoen, N. Cercone, K. Naruedomkul, B. Tipakorn, TSTMT: step towards an accurate Thai sign translation, Machine Learning and Applications, 2005. Proceedings. Fourth International Conference on , vol., no., pp.6 pp.,, 15-17 Dec. 2005.

[28] V.I. Pavlovic, R. Sharma, T.S. Huang, Visual interpretation of hand gestures for human-computer interaction: a review, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.19, no.7, pp.677, 695, Jul 1997.

[29] W. Kanjanapatmata, Sign language gesture recognition for Thai consonant using artificial neural Networks, King Mongkut's University of Technology North Bangkok, Bangkok, 2005.

[30] W. Veerasakulthong, Thai Hand Sign Recognition, M. Science. Thesis, National Institute of Development Administration, Bangkok, Thailand, 2007.

[31] W. Visarnkuna, Sign Language Recognition System by Hidden Markov Models, King Mongkut's University of Technology Thonburi, Bangkok, 2004.

[32] W. Jiangqin, G. Wen, The recognition of finger-spelling alphabets for Chinese sign language, the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction, 2001.

[33] X. Teng, et al., A hand gesture recognition system based on local linear embedding, Journal of Visual Languages & Computing Volume 16, Issue 5, October 2005, Pages 442–454.

[34] Y. Sribooruang, , P. Kumhom, K. Chamnongthai, Hand posture classification using wavelet moment invariant, Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2004. (VECIMS). 2004 IEEE Symposium on , vol., no., pp.78,82, 12-14 July 2004.

[35] Y. Tabata, T. Kuroda, Finger spelling recognition using distinctive features of hand shape, In: 7th ICDVRAT with Art Abilitation. Maia, Portugal: 2008:287-292.

[36] Ramer-Douglas-Peucker algorithm, June 25, 2014, Available at (online): http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorith.

[37] Finger-Spelling Recognition System using Fuzzy Finger Shape and Hand Appearance Features, May 10, 2014, Available at (online): http://www.youtube.com/watch?v=qW7i6TyVBaM.

[38] Thai Alphabets finger spelling, July 15, 2014, Available at (online): http://www.youtube.com/watch?v=tn7J0k1KZis&feature=youtu.be.

APPEDIX

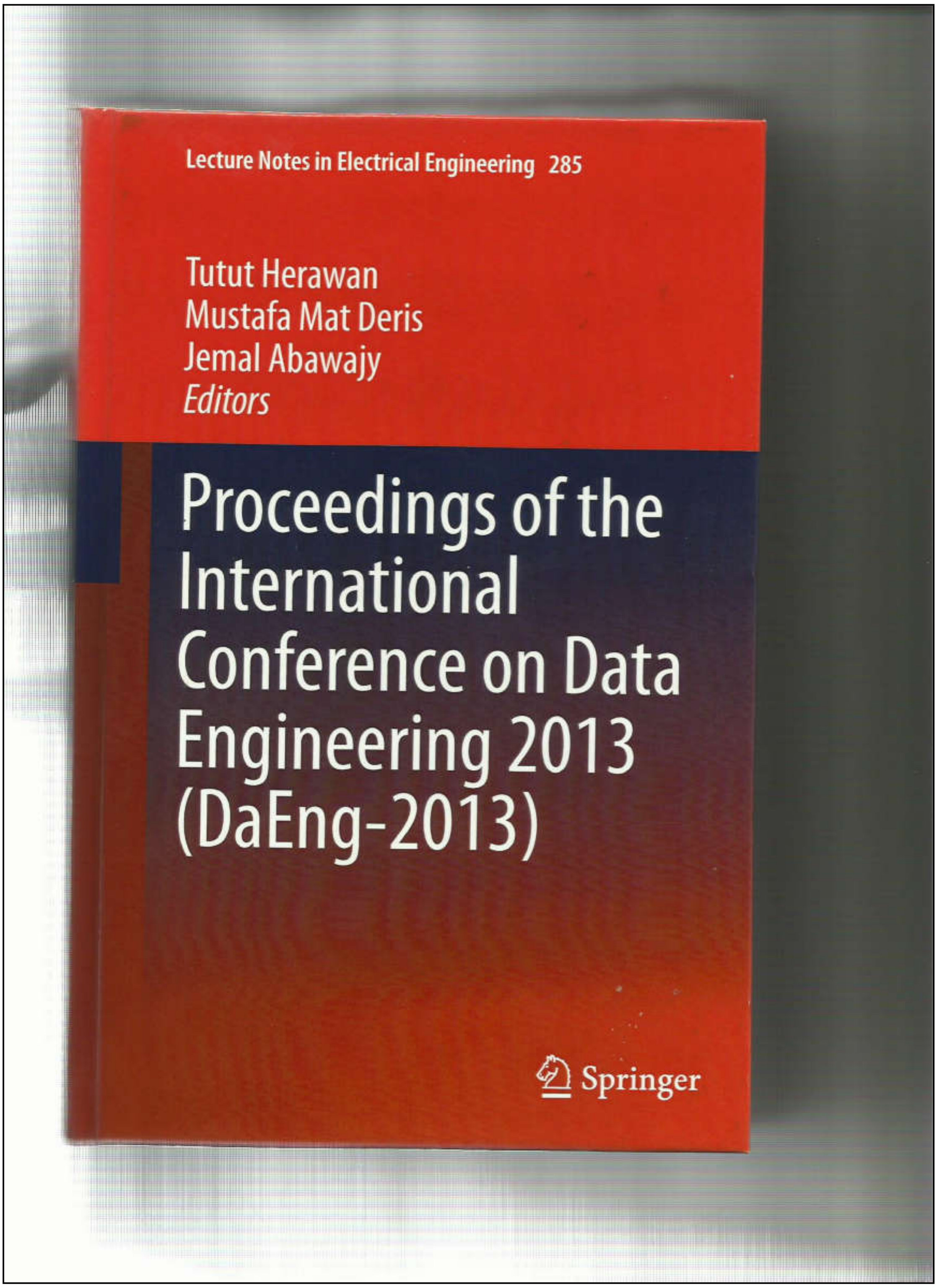
APPENDIX A.

PAPERS PUBLICATION

1. K. Silanon and N. Suvonvorn, “Fingertip Tracking Based Active Contour for General HCI Application,” Proceedings of First International Conference on Advanced Data and Information Engineering (DaEng-2013), Lecture Notes in Electrical Engineering, Vol.285, 2014, pp. 309-316.

2. K. Silanon and N. Suvonvorn, “Finger Spelling Recognition System using Fuzzy Finger Shape and Hand Appearance Features,” In the Fourth International Conference on Digital Information and Communication Technology and its Application (DICTAP2014), Thailand, May 6-8, 2014.

3. K. Silanon and N. Suvonvorn, “Fuzzy Finger Shape and Hand Appearance Feature for Thai Alphabet Finger-Spelling Recognition System,” Multimedia Tools and Applications (Submitted to Journal of Multimedia Tools and Applications).


Lecture Notes in Electrical Engineering 285
Tutut Herawan
Mustafa Mat Deris
Jemal Abawajy
Editors
Proceedings of the International Conference on Data Engineering 2013 (DaEng-2013)
Springer

...lysis by the Proposed Jackknife-
...ss-Validation.

...R (%)
...e-One-Out Cross-Validation (LOO)* *The higher the better
43.75
45.83
65.63
76.00
58.33
65.63
76.00

...ing soft biometrics. Information ...6-415 (2010)
..., Grother, P. J., Quinn, G. W., & ...gnition. In Automatic Face & ...EEE International Conference on,
...ing and retrieval in forensics
... M.: A group of facial normal ...trics: Theory, Applications and ...ce on, pp. 271-277 (2012)
... R. W.: Analysis of Facial Marks ...ns on Information Forensics and
...Modal Identification System in ...sing (IJIP), 7(1), 72 (2013)
... American Journal of Physical
...sis of gait as a cue to identity.
...Psychology: Human Perception
...m recognition by walking and ...37(5), pp. 1057--1072 (2004)
... and temporal features of gait ...ologies, 2005. Fourth IEEE
...ness of fit of autoregressive ...23--129 (1947)

Fingertips Tracking Based Active Contour for General HCI Application

Kittasil Silanon[1], Nikom Suvonvorn[1]

[1] Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Songkhla, Thailand 90112
{kittasil.silanon, nikom.suvonvorn}@gmail.com

Abstract. This paper presents a real time estimation method for 3D trajectory of fingertips. Our approach is based on depth vision, with Kinect depth sensor. The hand is extracted using hand detector and depth image from sensor. The fingertips are located by the analysis of the curvature of hand contour. The fingertips detector is implemented using concept of active contour which combine the energy of continuity, curvature, direction, depth and distance. The trajectory of fingertips is filtered to reduce the tracking error. The experiment is evaluated on the fingers movement sequences. Besides, the capabilities of the method are demonstrated on the real-time Human-Computer Interaction (HCI) application.

Keywords: We Fingertips Detection and Tracking, Hand Posture Estimation, Human-Computer Interaction (HCI).

1 Introduction

Hand gesture recognition has been a popular research in recent year. It provides more natural human-computer interaction. Many researches in this field related to real-time hand gesture recognition are proposed. Most of these system use trajectories of hand motion to recognize the commands [1,2,3,4,5]. However, the most important aspect of hand gesture recognition is to recognize commands accurately can be done with the accurate position of fingertips. Thus, some works have introduced on contours-based method for 2D fingertips tracking [6,7,8,9]. However, this approach cannot track fingertips robustly and usually design to track only stretched fingertips. Other systems for tracking the fingertips are using finger kinematic model [10,11,12] which search for the special form of the fingertips. These systems can work robustly. However, the computation cost is still too high. Using stereo vision has proposed to analyze the 3D fingertip positions [13,14,15,16]. The most problems in 3D fingertips are failure tracking when the fingertips are bending into the palm or overlapped each other. Therefore, this paper presented system to deal with such situations by using the depth image from Kinect. The hand region is segmented from the depth image and initial hand features are detected. The 3D fingertips are tracked using concept of active contour which occurs from internal and external energy that use features of

T. Herawan et al. (eds.), *Proceedings of the International Conference on Data Engineering 2013 (DaEng-2013)*, Lecture Notes in Electrical Engineering 285,
DOI: 10.1007/978-981-4585-18-7_35, © Springer Science+Business Media Singapore 2014

continuity, curvature, depth and distance from candidate fingertips that are detected at hand region in each frame. The energy represents the possible of candidate fingertips to be the fingertip in next frame. The tracking experiments are tested on basic finger movement. In addition, we develop an HCI application based on the fingertips tracking result. The rest of paper is organized in four sections by the following: initial hand segmentation, finger detection and tracking, experimentation results and conclusion respectively.

2 Initial Hand Segmentation

2.1 Hand Segmentation

From our previous work [5], we proposed hand detector by using object detection method which provides accurate result (Fig.1.a). The hand detector will be used to search hand's region in image. From experimentation, we found that hand detector failed when other parts of body such as face, arm move close to hand's region (Fig.1.b).

(a) (b)

Fig. 1. Hand segmentation: (a) hand detector (b) wrong segmentation.

Therefore, depth image (Fig.2.c) is used in the system. The depth is a 3D position vector (x,y,z) which obtained from the Kinect camera, where the x and y are the rows and columns in an image and z is the depth readings that are stored in the pixels, for detecting how pixel far away from camera.

(a) (b) (c)

Fig. 2. Depth information: (a) complex background (b) depth image (c) hand's region.

The depth can be solved the problem of complex background (Fig.2.a) by setting initial depth value from hand detector to remove any object behind the hand. Thus, our system takes depth information to extract only hand's region in image. After extraction hand's region, the initial hand's features will be estimated. For example, hand center, fingertips position, palm size etc. All initial hand features are used to be reference value to compare changing of hand gesture in command recognition system.

2.2 Hand Center Point

We obtained the center p
moments of pixels in hand's

In the above equations,
and y are range over the
(Fig.3.a). The palm size is
closest pixel on hand contour

Fig. 3. Initial

2.3 Fingertips Position

Since the user is required
"open" right hand facing the
curvature of boundary point
consider all the boundary
approximation method to extr
point P1,…,Pn (Fig.4.a). Each
slope. The angle can be estim
of key point by two vectors
(Fig.4.b). The key point, with
is an initial fingertip (Fig.4.c).

(a)

Fig. 4. Fingertips position: (a)

The initial positions of the fi
a label that corresponds to the
control initial open hand posture
on sorting the five points by clo
can only detect fingertip in ope
fingertips which can be changed

2.2 Hand Center Point

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:

$$X_c = \frac{M_{10}}{M_{00}},\ Y_c = \frac{M_{01}}{M_{00}} \quad \left(M_{ij} = \sum_x \sum_y x^i y^j I(x,y)\right) \tag{1}$$

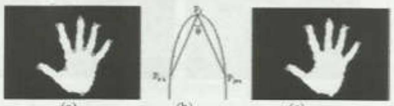
In the above equations, $I(x,y)$ is the pixel value at the position (x,y) of the image, x and y are range over the hand's region. The center point is calculated as Xc *and* Yc (Fig.3.a). The palm size is defined as the distance between the center point and the closest pixel on hand contour (Fig.3.b).

(a) (b)

Fig. 3. Initial hand feature: (a) hand center (b) palm size.

2.3 Fingertips Position

Since the user is required to initialize the system by producing the pose of an "open" right hand facing the camera. Therefore, it is simple to locate fingertips from curvature of boundary point of hand's region. However, it may not be necessary to consider all the boundary points of hand's region. Thus, we use the polygon approximation method to extract key point [18] and is stored in a new series of key point P1,…,Pn (Fig.4.a). Each key point Pi has two parameters, the angle (θ) and slope. The angle can be estimated by using k-curvature [6] which calculates the angle of key point by two vectors [$P(i\text{-}k)P(i)$] and [$P(i)P(i+k)$] with the same range (k) (Fig.4.b). The key point, with curvature value is in the threshold and slope is positive, is an initial fingertip (Fig.4.c).



(a) (b) (c)

Fig. 4. Fingertips position: (a) key point (b) curvature calculation (c) fingertip points.

The initial positions of the five fingertips are detected. Each of them will be given a label that corresponds to the thumb, index, middle, ring and pinky fingers (f_j). As we control initial open hand posture to frontal view. We can simply label fingers based on sorting the five points by clockwise arranging around palm center. Nevertheless, it can only detect fingertip in open hand. Thus, tracking method is used for tracking fingertips which can be changed position at all time in hand gesture sequence.

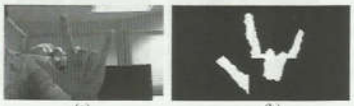
3 Finger Detection and Tracking

3.1 Finger Location

Most movements in hand gesture are finger movements (Fig.5.a). Therefore, we defined two conditions to segment finger locations (Fig.5.b). For a stretching finger, use distance condition and a bending finger, use depth conditions follow the rules:

$$D(p,C_0) > k.R \quad \| \quad |Z(p) - Z(C_0)| < \tau_D \tag{2}$$

Where k is a scaling factor, D (p, C_0) is distance between considering pixel and hand center, Z is depth value and τ_D is a pre-defined depth threshold.



(a) (b)

Fig. 5. Finger locations: (a) origin hand posture (b) extraction finger location.

3.2 Candidate Fingertips

We define searching area to locate the candidate fingertip positions (Cf_i). We assume that fingertip positions should be points on hand contour. Therefore, it may not be necessary to consider all points on hand contour. Hence, the polygon approximation algorithm is used again to find candidate fingertips. Finger.6.b shows points on contour for stretching finger. In addition, candidate fingertip positions can be found by assuming that they are closest to the camera in each finger region. Thus, we use depth to find point which has minimum depth to be candidate fingertips. Fig. 6.b shows these points inside contour.

(a) (b)

Fig. 6. Candidate fingertips: (a) origin hand posture (b) candidate fingertips.

3.3 Fingertip Tracking

The tracking of the fingertip position between successive frames is built by concept of active contour [18,19]. The possible candidate fingertips of each fingertip will be assigned energy and then the maximum energy point is chosen to be the fingertip in the next frame. The discrete formulation of energy function which can be written as:

$$E(Cf_i) = E_{internal}(Cf_i) + E_{external}(Cf_i) \quad (3)$$

The energy for each candidate fingertips can be decomposed into two basic energy term. $E_{internal}$ represents the internal energy of the candidate due to bending or stretching of finger and $E_{external}$ is the external constraint introduced by user. The internal Energy of candidate fingertips is defined as:

$$E_{internal} = E_{continuity} + E_{curvature} + E_{depth} \quad (4)$$

The $E_{continuity}$. It forces the candidate fingertip points to be continuous, because the fingertip should not change much from the current point to the next one. Therefore, this term tries to keep point which has appropriate distance between the average distance ($\bar{d}$) and candidate fingertip point. The form for $E_{continuity}$ is the following:

$$E_{continuity} = \bar{d} - \left\| f_j - Cf_i \right\|^2 \quad (5)$$

The $E_{curvature}$, this term will find smoothness of the candidate fingertips by considering contour curvature between the two vectors, A = $(x_i - x_{i-k}, y_i - y_{i-k})$ and B = $(x_{i+k} - x_i, y_{i+k} - y_i)$, k is constraint. If the candidate fingertips have the curvature value that fall under a threshold, these points will be kept to be possible candidate fingertips. The formula for $E_{curvature}$ is given by:

$$E_{curvature} = \cos^{-1} \frac{\mathbf{A.B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \quad (6)$$

Because we have use property of depth, the E_{depth} has been established. The E_{depth} is distance between candidate fingertips to camera that represent in 16-bit depth data units of millimeter. As we have assumed, the fingertips should be found at the closest point to the camera. Therefore, the E_{depth} becomes maximum value when the candidate fingertips get close to a camera. The closest point will be given more priority than the rest points in order of depth. As we mentioned previously, The $E_{external}$ is the external constraint. Thus, in our system, the distances from all fingertips to considering candidate fingertip in image are used to describe about the $E_{external}$. For instance, if we are considering the movement of index fingertip, the distances from other fingertips to considering candidate fingertip point are equivalent to the external energy which will be used to estimate suitability for choosing the considering candidate fingertip to be the index fingertips. The distance from index fingertip to considering candidate fingertip should be shorter than other fingertip. On the other hand, if the distances from other fingertips are shorter than index fingertip distance, the considering candidate fingertip should be assigned to another fingertip. The energy function represents the importance of candidate fingertips relative to each fingertip. The candidate fingertip with maximum energy is selected to be new location of fingertip in next frame. In order to reduce the tracking error due to losing depth value, a simple low-pass filter is applied for the smoothness trajectory of fingertip tracking. The average point between selected point and current fingertip point will be estimated.

4 Experimentation Result

4.1 Fingertips Tracking Precision

We evaluate the fingertip tracking precision on basic finger movements between the tracked fingertips and the ground truth. We have defined the ground truth using the end point contour of each finger. The basic finger movements have included five sequences (Fig.7), bending finger (seq.1), moving finger into the palm (Seq.2), for crossing finger (Seq.3), hand movement (up, down, left and right) (Seq.4). 45^0 hand rotation (counterclockwise and clockwise) (Seq.5). Each sequence is tested at 10 rounds. Table1 shows the precision in terms of the Euclidean distance.

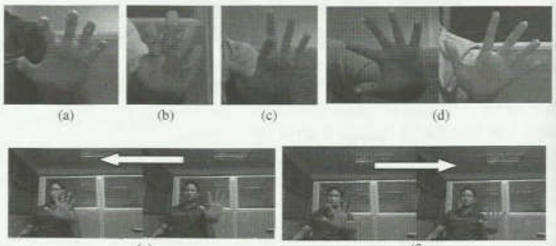


Fig. 7. Basic fingertip movements: (a) bending finger (b) moving finger into the palm (c) crossing finger (d) 45^0 hand rotation (e-f) hand movement.

Table 1. Fingertips Tacking Precision

Seq.	Tracking Precision(Pixel)					Avg.
	Thumb	Index	Middle	Ring	Pinky	
1	7.94	7.65	8.50	9.25	8.51	8.37
2	25.35	14.28	14.96	23.12	15.69	18.68
3	7.47	6.06	5.12	5.60	5.10	5.87
4	5.60	4.16	3.06	5.05	5.77	4.72
5	4.17	3.44	5.80	3.76	7.19	4.87

From experimentation result, we found that the tracking cases for crossing, rotation and movement (Seq.3, Seq.4 and Seq.5) give a good precision (≈ 4-6 pixels error) because the candidate fingertips obtained from polygon approximation algorithm are quite well located to the fingertip ground truth. But the tracking cases for bending and

moving into palm (Se
the imperfect depth
function. Hence, dept
there might produce s

4.2 Human-Comput

We carried out e
commands to interfa
example, pressing up,
time interactive at: htt

Fig. 8. HC

5 Conclusion

Fingertips and palm
The most of previo
complexity of finger
these issues by using
concept of active con
shows good performan
Human-Computer Int
limitation in tracking
neighborhood candida
are too fast. For the
Human-Computer Int
controlling, visual inpu

Acknowledgments. W
of Thailand's Office of

References

1. Deng, J.W., Tsui, H.T
In: 15th International
(2000)

moving into palm (Seq.1 and Seq.2) give less precision (≈ 9-20 pixels error) due to the imperfect depth data received from Kinect that is not suitable for near mode function. Hence, depth data of finger may lose in some frames. From these factors, there might produce some errors in the fingertips tracking process.

4.2 Human-Computer Interaction Application

We carried out experiments on a general HCI Application that used as the commands to interface with Window Media Center on Window 7 (Fig.8.). For example, pressing up, down, left, right button. You can see video showing the real-time interactive at: http://www.youtube.com/watch?v=OnQra4We-4o

Fig. 8. HCI Application for controlling Window Media Center.

5 Conclusion

Fingertips and palm positions are significant features for hand gesture recognition. The most of previous works cannot track 3D fingertip positions because the complexity of finger movement. In this paper, we present the method to deal with these issues by using depth data feature for correctly hand segmentation and apply concept of active contour to track fingertips over finger movement. Our method shows good performance in term of real-time and also has capability to expansion to Human-Computer Interaction application. However, our method still has some limitation in tracking fingertips. The fingertips tracking procedure fails if the neighborhood candidate fingertips are lost, which is the case of the finger movements are too fast. For the possible applications, our system can be combined with other Human-Computer Interaction applications, such as finger-spelling process, robot controlling, visual input device and etc.

Acknowledgments. We would like to thank the National Research University Project of Thailand's Office of the Higher Education Commission for financial support.

References

1. Deng, J.W., Tsui, H.T.: An HMM-based approach for gesture segmentation and recognition. In: 15th International Conference on Pattern Recognition Proceedings, vol. 3, pp. 679-682 (2000)

2. Ho-Sub Yoon, Jung Soh, Younglae J. Bae, Hyun Seung Yang: Hand gesture recognition using combined features of location, angle and velocity. In: Pattern Recognition, vol. 34, pp. 1491–1501 (2001)
3. Feng-Sheng Chen, Chih-Ming Fu, Chung-Lin Huang: Hand gesture recognition using a real-time tracking method and hidden Markov models. In: Image and Vision Computing, vol. 21, pp. 745–758 (2003)
4. Elmezain M., Al-Hamadi: Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Model. In: IEEE International Symposium on Signal Processing and information Technology, pp.1192-1197 (2007)
5. Kittasil Silanon, Nikom Suvonvorn: Hand Motion Analysis for Thai Alphabet Recognition using HMM. In: International Journal of Information and Electronics Engineering vol. 1, pp. 65-71 (2011)
6. Wei Du, Hua Li: Vision based gesture recognition system with single camera. In: 5th International Conference on Signal Processing Proceedings WCCC-ICSP, vol. 2, pp.1351-1357 (2000)
7. Antonis A. Argyros, Manolis I. A. Lourakis: Vision-based interpretation of hand gestures for remote control of a computer mouse. In: Computer Vision in Human-Computer Interaction, pp. 40-51 (2006)
8. Ko-Jen Hsiao, Tse-Wei Chen, Shao-Yi Chien: Fast fingertip positioning by combining particle filtering with particle random diffusion. In: IEEE International Conference on Multimedia and Expo, pp. 977-980 (2008)
9. J. Ravikiran, Mahesh Kavi, Mahishi Suhas, R. Dheeraj, S. Sudheender, Pujari Nitin V.: Finger Detection for Sign Language Recognition In: International MultiConference of Engineers & Computer Scientists, pp. 489 (2009)
10. Lee, J., Kunii, T.L.: Model-based analysis of hand posture. In: IEEE Computer Graphics and Applications, vol.15, no.5, pp.77-86 (1995)
11. Cheng-Chang Lien, Chung-Lin Huang: Model-based articulated hand motion tracking for gesture recognition. In: Image and Vision Computing, vol. 16, Issue 2, pp. 121-134, (February 1998)
12. Lathuiliere, F., Herve, J. Y.: Visual tracking of hand posture with occlusion handling. In: 15th International Conference on Pattern Recognition Proceedings, vol.3, pp.1129-1133 (2000)
13. Dung Duc Nguyen, Thien Cong Pham, Jae Wook Jeon: Fingertip detection with morphology and geometric calculation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1460-1465 (2009)
14. M. Do, T. Asfour, R. Dillman: Partical filter-based fingertips tracking with circular hough transform feature. In: Proceedings of the 12th IAPR Conference on Machine Vision Application (2011)
15. Raheja, J.L., Chaudhary, A., Singal, K.: Tracking of Fingertips and Centers of Palm Using KINECT Computational Intelligence. Third International Conference on Modeling and Simulation (CIMSiM), pp. 248-252 (2011)
16. Hui Liang, Junsong Yuan, Daniel Thalmann: 3D fingertip and palm tracking in depth image sequences. In: Proceedings of the 20th ACM international conference on Multimedia (MM '12). ACM, New York, NY, USA, pp.785-788 (2012)
17. Ramer-Douglas-Peucker algorithm, http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm
18. Michael Kass, Andrew Witkin, Demetri Terzopoulos: Snakes: Active contour models. In: International journal of computer vision, vol. 1, no. 4, pp. 321-331 (1988)
19. Donna J. Williams, Mubarak Shah: A Fast algorithm for active contours and curvature estimation. In: CVGIP: Image Understanding, vol. 55, no. 1, pp. 14-26 (January 1992)


The Fourth International Conference on Digital Information and Communication Technology and its Applications (DICTAP2014)
มหาวิทยาลัยหอการค้าไทย
THE UNIVERSITY OF THE THAI CHAMBER OF COMMERCE
Bangkok, Thailand
May 6-8, 2014
IEEE
THAILAND SECTION
SDIWC
UTCC
50 YEARS OF LEADERSHIP WISDOM
The Society of Digital Information and Wireless Communications (SDIWC)
ISBN: 978-1-4799-3723-3 ©2014 IEEE

Finger-Spelling Recognition System using Fuzzy Finger Shape and Hand Appearance Features

Kittasil Silanon
Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University
Hatyai, Songkhla, Thailand 90112
Kittasil.silanon@gmail.com

Nikom Suvonvorn
Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University
Hatyai, Songkhla, Thailand 90112
Nikom.SUVONVORN@gmail.com

***Abstract*— In this paper, we introduce a method for finger-spelling recognition system. The objective is to help the deaf or non-vocal persons to improve their skills on the finger-spelling. Many researches in this field have proposed methods mostly based on hand posture estimation techniques. We propose an alternative flexible method based on fuzzy finger shape and hand appearance analysis. By using depth image, the hand is extracted and tracked using an active contour like method. Its features, such as, finger shape, and hand appearance, have been defined as chain code, which are input to the American finger-spelling recognition system by using a vote method. The performance of the system is tested in real-time environment, which results in around 70% recognition rate.**

Keywords—Finger-Spelling; Hand Posture Estimation; American Finger-Spelling

I. Introduction

In this paper, we present our research on sign language, especially related to the finger-spelling. The finger-spelling is a basic communication method for deaf and non-vocal persons, in which the hand posture as symbol will represents the alphabets of words of spoken language, such as, names, places, technical words and etc. However, most of these people, especially children, have problems with finger-spelling skills. Usually, the word-level vocabulary signs have been used for communicating with each other, and only 7% to 10% of the finger-spelling is used in the daily life. Evidently, the finger-spelling skills lag far behind the sign language skills. Our research goal to the field is to develop an automatic recognition system for the finger-spelling, in order to help these people to improve their skills. Actually, many systems specific to a language are proposed, for examples, finger-spelling of American (ASL) [11,13,15], British (BSL) [12], Australian (Auslan) [10], Chinese (CSL) [18], Japanese (JSL) [17] etc. Various researches have been proposed, but most of them cannot achieve the critical criteria, such as, accuracy, flexibility, and real time constraint. There are two principle approaches: glove-based and vision-based. The gloves-based methods [14,16] use electronic sensor devices for digitizing hand joint and finger motion, which give the precision of the hand posture that result in high recognition rate in real time, but these methods are very limited by the environment configuration. The vision-based approach consists of two groups of techniques. Firstly, the model-based method [1,5] uses a kinematic hand model to estimate the articulated hand (i.e., joint angle, finger position), leading to a full reconstruction of the articulated hand posture. Secondly, the appearance-based method [4,9,19] uses computer vision techniques to extract important features from images, such as, point, edge, contour or silhouette, for reconstructing the hand posture, and then, for recognizing the finger-spelling.

In this paper, we proposed a vision-based method for hand posture estimation. The method combines both model and appearance-based method using finger shape and hand appearance features, to finally recognize the American finger-spelling. The system consists of four main parts: 1) hand segmentation, to segment the region of interest of the hand form image sequence, 2) key hand posture selection, to determine the key frame representing the hand posture of finger-spelling from image sequences, 3) hand feature definition, to define the finger shapes and hand appearance features as chain code sequence, 4) finger-spelling recognition, to recognize the finger-spelling from hand features by simply using a scored voting method. The paper details the four parts of our method, then the experimentation results and conclusion, respectively.

II. Hand Segmentation

In this step, the method is focused on the segmentation of the region of interest of the hand from the rest of the image. In our experimentation, the hand is simply defined by the closest object to the camera. Since we used depth image, as shown in Fig.1.b, thus the segmentation of the hand from the complex background can be done by using predefined threshold to obtain the hand's region in image, as shown in Fig.1.c.

(a) (b) (c)

Fig. 1. Depth information: (a) image (b) depth image (c) hand region.

ISBN: 978-1-4799-3724-0/14/$31.00 ©2014 IEEE

After hand's region extraction, the hand's features will be estimated and used as hand initialization, for examples, hand's center, fingertips position, palm size, and etc. These initial features of the hand will be used as reference values to compare with the hand gesture changing in finger-spelling recognition system. For the center of the hand, we used the image moment technique to obtain the center point, as shown in Fig.2.a. The palm size is defined by the distance between hand's center and the closest pixel on hand's contour, as shown in Fig.2.b. The fingertip's positions are points located at the curvature of hand's region, as shown in Fig.2.c. We simply label the five fingers by clockwise sorting of points around the palm's center. The detail of our method in this section is published in [7].

(a) (b) (c)

Fig. 2. Initial features: (a) hand center (b) palm size (c) fingertip position.

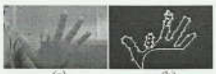
III. Key Hand Posture Selection

In this section we detail how to select the key frames of the hand posture that represent the finger-spelling alphabets in the image sequences. Note that not every frame in the image sequences is necessary to the recognition system, due to the transition frames (from one alphabet to another) that cannot clearly define the alphabet. The selection condition is that the key frame is selected if the fingertips' position is not changed from the last frame. So, the fingertips must be tracked though the image sequences to determine its changing status. Firstly, the finger area is separated from the palm area. Due to the highest movement of finger during finger-spelling, the finger is then segmented in order to reduce the tracking errors, using its characteristics, stretching and bending fingers, as shown in Fig.3.b.

(a) (b)

Fig. 3. Fingers locating: (a) original image (b) finger location.

We assume that the corresponding fingertips in the next frame must be the points located on the contour of the segmented area, which locations do not change much from the current point. The Fig.4.b shows the candidate fingertips, which are calculated with the polygon approximation approach.



(a) (b)

Fig. 4. Candidate fingertip: (a) original image (b) finger location.

The correspondence of any fingertip between successive frames is then determined by using the concept of active contour [2,8] and the limited searching area. The active contour energy of each candidate fingertip is defined by the energy of continuity, curvature, depth, direction, and distance. The candidate fingertip which has the maximum energy is chosen as its corresponding fingertip in the next frame. The examples of tracking fingertips are shown in Fig.5.

(a) (b) (c) (d) (e)

Fig. 5. Finger tracking: (a) bending (b) closing (c) crossing (d-e) rotation.

Finally, the stable frame will be selected as the key frame of the hand posture. The different position of the fingertip as in (1) between successive frame images is computed in order to detect the temporal difference. The minimum difference is selected as the key hand posture in video. More details of the method are explained in our publication [7].

$$D_k(x,y) = \sum_{i=0}^{n} \left| P_{k-1}(x,y) - P_k(x,y) \right|_i \le \tau_\theta \qquad (1)$$

IV. Hand Feature Definition

The finger shape and hand appearance features have been defined. All of features are converted as feature chain code to discriminate the difference between a set of hand postures.

A. Fuzzy Finger Shape Feature

The finger shape is used to discriminate the difference between a set of hand postures. We define the four basic finger shapes as the following: open, bend, point, and close, as shown in Fig.6.

(a) (b) (c) (d)

Fig. 6. Finger shape: (a) open (b) bend (c) point (d) close.

In this step, the fuzzy logic [3,6] is applied in order to classify the finger shapes by using the fingertips characteristics obtained from the last section. The process of fuzzy logic is explained in Fig.7. Firstly, a crisp set of input data are gathered and converted to fuzzy set using linguistic variable, fuzzy linguistic terms and membership functions (μ). This step is called as fuzzification. Subsequently, an inference is made for the evaluation result based on a set of rules. Finally, the resulting fuzzy output is mapped to a crisp output using membership functions, in the defuzzification step. In our system, the linguistic variables are defined as depth (D), distance (Dist) and shape (S). The depths at fingertip positions, and distances between fingertip's positions and center of the hand are the input variables, and the shape is the output variable. For linguistic terms, depth and distance variables are decomposed into a set of "Near", "Middle" and "Far" terms, $D(t)$ = *{Near, Middle, Far}* and $Dist(t)$ = *{Near, Middle, Far}*. The range of each term can be shown in Fig.8. For shape variable, linguistic terms are open, bend and point $S(t)$ = *{Open, Bend, Point}*. We do not include "Close" into shape linguistic terms because "Close" shape will be classified immediately when fingertip position is in the palm region. We use the trapezoidal curve as the type of the membership function of each linguistic term. The fuzzification will be performed to convert all crisp inputs to a fuzzy input set.

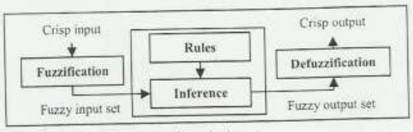
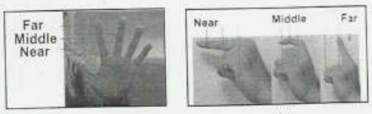


Fig. 7. Fuzzy logic system.



(a) (b)

Fig. 8. Lingusitic terms: (a) distance (b) depth.

The rule is defined for determining the output variables. Our fuzzy rule is defined as a simple IF-THEN rule with a condition and conclusion. Table.1 shows the fuzzy rules for the finger shape classification. Table.2 shows the matrix representation of fuzzy rules. The rows are the depth values, the columns are the distance values, and each cell is the shape output, which corresponds to the specific row and column. For example, the cell (3,2) of the matrix can be described as follows: IF distance is middle AND depth is near THEN shape is point.

TABLE I. FUZZY LOGIC RULE EXAMPLE

Fuzzy Rules
1. IF Distance is Far AND Depth is Far THEN Shape is Open
2. IF Distance is Middel AND Depth is Middle THEN Shape is Bend
3. IF Distance is Near AND Depth is Near THEN Shape is Point

TABLE II. FUZZY MATRIX EXAMPLE

Distance/Depth	Near	Middle	Far
Near	Point	Point	Point
Middle	Point	Bend	Bend
Far	Open	Open	Open

The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations (AND, OR and NOT), this process is called inference. After evaluating the result of each rule and obtaining the final result, the defuzzification will be performed according to the membership function of the fuzzy output variable. For instance, we have the result in Fig.9. at the end of inference.

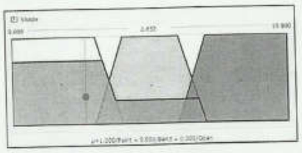


Fig. 9. Fuzzy result.

In this example, all shaded areas belong to the fuzzy result, which must be defuzzified to obtain a final crisp output. There are different algorithms for defuzzification. In our experimentation, we used the center of gravity (COG) in order to map the fuzzy output to a crisp output, represented with a dot in the Fig.9. The COG algorithm can be written as (2).

$$COG = \frac{\sum_{i=1}^{N} x\mu(x)}{\sum_{i=1}^{N} \mu(x)} \tag{2}$$

B. Hand Appearance Fetaure

Considering only the finger shapes is not sufficient to estimate the correct hand posture used in the finger-spelling. We found that, in general, there are some hand postures having the similar finger shape, but having a different hand appearance. Thus, we define the additional hand appearance features for increasing robustness of the hand features. The hand appearance features are defined by the relative distance and position between fingers in three statuses: "close", "separate" and "cross". Two more features are defined as the hand rotation and hand movement. The hand appearance feature will be combined with finger shape features in order to increase the hand posture discrimination. The set of the hand appearance features in finger-spelling system are shown in Fig.10.

(a) (b) (c) (d) (e)

Fig. 10. Hand Appearance: (a) group (b) separate (c) cross (d) rotation (e) movment

ISBN: 978-1-4799-3724-0/14/$31.00 ©2014 IEEE

C. Feature Codes

To represent the hand posture, the finger shape and hand appearance features are encoded as shown in table.3. In our system, the feature vector is composed of 17 values defined as showed in Fig.11. The first 5 elements (f_1,f_2,f_3,f_4,f_5) indicate the finger shape (thumb, index, middle, ring and pinky respectively). In Fig.11, $(f_1,f_2,f_3,f_4,f_5) = (4,1,1,4,4)$ means that the thumb is "close", index and middle are "open", ring and pinky are "close", respectively. The next 4 elements $(f_{12},f_{13},f_{14},f_{15})$ describe the relation between thumb with index, middle, ring and pinky fingers. In Fig.11, $(f_{12},f_{13},f_{14},f_{15}) = (6,6,6,6)$ indicates that all relations of thumb with other fingers are "separate". The next 3 elements (f_{23},f_{24},f_{25}) describe the relation of index with middle, ring and pinky fingers. The next 2 elements (f_{34},f_{35}) describe the relation of middle with ring and pinky fingers. In Fig.11, $(f_{34},f_{35}) = (6,6)$ indicates that all relations of middle with other fingers are "separate". The element (f_{45}) describes the relation of ring and pinky. In Fig.11, $(f_{45}) = (5)$ indicates that the relation of ring with pinky finger is "close". The next 2 elements (H_r,H_m) represent the rotation and movement of hand. In Fig.11, $(H_r,H_m) = (0,0)$ indicates that there is no rotation or movement for this hand posture. Finally, the vector feature of the hand posture in Fig.11 have the chain codes as P = {4,1,1,4,4,6,6,6,6,6,6,6.6,6.5,0,0}.

TABLE III. FEATURE CODES

Finger shape	Hand appearance
Open = 1	Group = 5
Bend = 2	Separate = 6
Point = 3	Cross = 7
Close = 4	Rotation = 8, No-Rotation = 0
	Movement = 9, No-Movement = 0

Hand posture codes

$f_1 = 4$	$f_2 = 1$	$f_3 = 1$	$f_4 = 4$	$f_5 = 4$
$f_{12} = 6$	$f_{13} = 6$	$f_{14} = 6$	$f_{15} = 6$	
$f_{23} = 6$	$f_{24} = 6$	$f_{25} = 6$		
$f_{34} = 6$	$f_{35} = 6$			
$f_{45} = 6$				
$H_r = 0$	$H_m = 0$			

Fig. 11. Hand posture encoding.

V. SCORED VOTING RECOGNITION

The voting and scoring methods are combined and used in our recognition process. This method is simple and effective and can run in real time. Firstly, the n patterns of the hand posture chain codes (P_i | i = 1...N) are the pre-defined and the off-line computation. Each pattern represents a finger-spelling that is the chain code of finger shape features and hand appearance features. Secondly, any hand posture, which is captured from the camera, will be detected and encoded into the same type of chain codes. Thirdly, the similarity between the detected pattern and the n pre-defined patterns are performed by using a simple scored voting method. This technique will compare each element of the input chain codes with the corresponding element of the pattern chain code. The score will be given if any element has matched. Thus, the hand posture will be recognized to the pattern which gives the maximum score. The method can be used to recognize the different sets of hand posture by using different pre-defined corresponding chain codes for each of the hand posture. The scored voting recognition process can be shown in Fig.12.

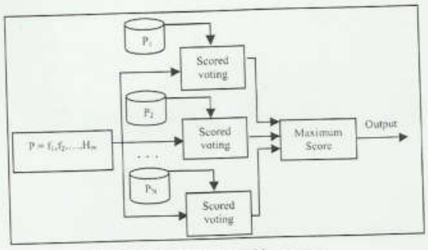


Fig. 12. Scored voting recognition process.

VI. EXPERIMENT RESULT

We applied our method to the American finger-spelling. There are 31 patterns of the pre-defined chain code of hand postures, consisting of 26 American-English alphabets (A-Z) and 5 hand postures for 1-5 numbers. Each hand posture is tested at 100 rounds. All hand postures and the feature chain code are shown in Fig.15 and table 4. The result of the recognition is shown in Fig.14, and the real-time example of our system in shown in Fig.13, available at: www.youtube.com/watch?v=qW7i6TyVBaM&feature=youtu.be

Fig. 13. Real-time American finger-spelling recognition system.

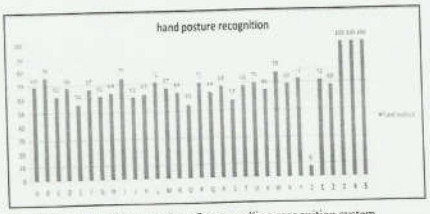


Fig. 14. Real-time American finger-spelling recognition system.

ISBN: 978-1-4799-3724-0/14/$31.00 ©2014 IEEE

As the result shown in Fig.15, the best cases (100% recognition rate) of the system are the hand postures of the numbers "3", "4" and "5". The hand postures of these 3 cases are not much different from initial hand posture (Open hand). The movement of finger does not much change. Thus, finger with immediate changing direction that increases the tracking error. For the general cases (65% recognition rate), "A-Y" and "1-2" hand postures, most of these hand postures have only some finger movements continuously, and does not change much from the previous frame. Hence, our finger tracking using active contour concept can perform more accurately tracking process gives less error. The worst case (less than 10% recognition rate) is the hand posture of "Z" alphabet. We found that this hand posture has the rapid hand movement

VII. CONCLUSION

We proposed an automatic recognition system for finger-spelling using hand posture estimation. Depth image is used for robust hand region segmentation and removing the complex background. The active contour concept is applied for fingertip tracking. The finger-shape and hand appearance features are proposed. The scored voting method is introduced for the n-pattern hand posture chain codes recognition. Our system is tested with the American finger-spelling including 26 hand postures for the A-Z alphabets and 5 hand postures for the 1-5 numbers. The recognition rate (not including "Z" alphabet hand posture) is performed in average at around 70%. The system can be performed in real time. We expect that our method can be applied to the other hand posture applications, such as, game, robotic controlling, visual input device, another language finger-spelling, and etc. However, our main future work is to increase the tracking accuracy of the fingertips that greatly effect to the final performance of the system.

ACKNOWLEDGMENT

We would like to thank the National Research University Project of Thailand's Office of the Higher Education Commission for financial support.

REFERENCES

[1] C. C. Lien, "A scalable model-based hand posture analysis system," Machine Vision and Applications, 2005, Volume 16, Number 3, Page 157.

[2] D.J. Williams and M. Shah, "A Fast algorithm for active contours and curvature estimation," In: CVGIP:Image Understanding, vol.55, no.1, pp.14-26(January 1992).

[3] E. Cox, "Fuzzy fundamentals," Spectrum, IEEE, vol.29, no.10, pp.58-61, Oct. 1992

[4] G. Panin, S. Klose, A. Knoll, "Real-Time Articulated Hand Detection and Pose Estimation," Advances in Visual Computing, Lecture Note in Computer Science, vol 5876, pp.1131-1140,2009.

[5] J. Lee and T.L Kunii, "Model-based analysis of hand posture," Computer Graphics and Applications, IEEE , vol.15, no.5, pp.77,86, Sep 1995.

[6] J.M. Mendel, "Fuzzy logic systems for engineering: a tutorial," Proceeding of the IEEE, vol.83, no.3, pp.345-377, Mar 1995.

[7] K. Silanon and N. Suvonvorn, "Fingertips Tracking Based Active Contour for General HCI Application," (DaEng-2013) Lecture Note in Electrical Engineering, vol.285, 2014, pp 309-316.

[8] M. kass, A. Witkin, D. Terzopoulos, "Snake: Active contour models," In: International journal of computer vision, vol.1, no.4, pp.321-331 (1988).

[9] N. Shimada, K., Kimura, Y. Shirai, "Real-time 3D hand posture estimation based on 2D appearance retrieval using monocular camera," Proc. ICCV workshop on Recognition, Analysis, and Tracking of Face and Gesture in Real-Time System, Vancouver, Canada, pp.23-30.

[10] P. Goh and E.J. Holden, "Dynamic Fingerspelling Recognition using Geometric and Motion Features," Image Processing, 2006 IEEE International Conference on , vol., no., pp.2741-2744, 8-11 Oct. 2006.

[11] R. Feris, M. Turk, R. Raskar, K. Tan, G. Ohashi, "Exploiting Depth Discontinuities for Vision-Based Fingerspelling Recognition," Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on , vol., no., pp. 155, 27-02 June 2004.

[12] S. Liwicki and M. Everingham, "Automatic recognition of fingerspelled words in British Sign Language," Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on , vol., no., pp.50-57, 20-25 June 2009.

[13] S. Ricco and C. Tomasi, "Fingerspelling Recognition through Classification of Letter-to-Letter Transitions," 9th Asian Conference on Computer Vision, pages 214--225, September 2009.

[14] S.S. Fels and G.E. Hinton, "Glove-talk: A neural network interface between a data-glove and a speech synthesizer," IEEE Transactions on Neural Networks, vol.4, pp.2-8, 1993.

[15] T.B. Dinh, V.B. Dang, D.A. Duong, T.T. Nguyen, D.D. Le, "Hand gesture classification using boosted cascade of classifiers," Proceedings of the 4th IEEE International Conference on Research, Innovation and Vision for the Future, RIVF'06, p. 139-144, 2006.

[16] T. Baudel and M.B. Lafon, "CHARADE: Remote Control of Object using FreeHand Gesture," Communications of the ACM, vol 36, pp.29-35, 1993.

[17] Y. Tabata and T. Kuroda, "Finger spelling recognition using distinctive features of hand shape," In: 7th ICDVRAT with Art Abilitation. Maia, Portugal: 2008:287-292.

[18] W. Jiangqin and G. Wen, "The recognition of finger-spelling alphabet for chinese sign language," the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction.

[19] Z. Pan, Y. Li, M. Zhang, C. Sun, K. Guo, X. Tang, S.Z. Zhou, "A real-time multi-cue hand tracking algorithm based on computer vision," Virtual Reality Conference (VR), 2010 IEEE, vol.,no.,pp.219-222, 20-24 March 2010.

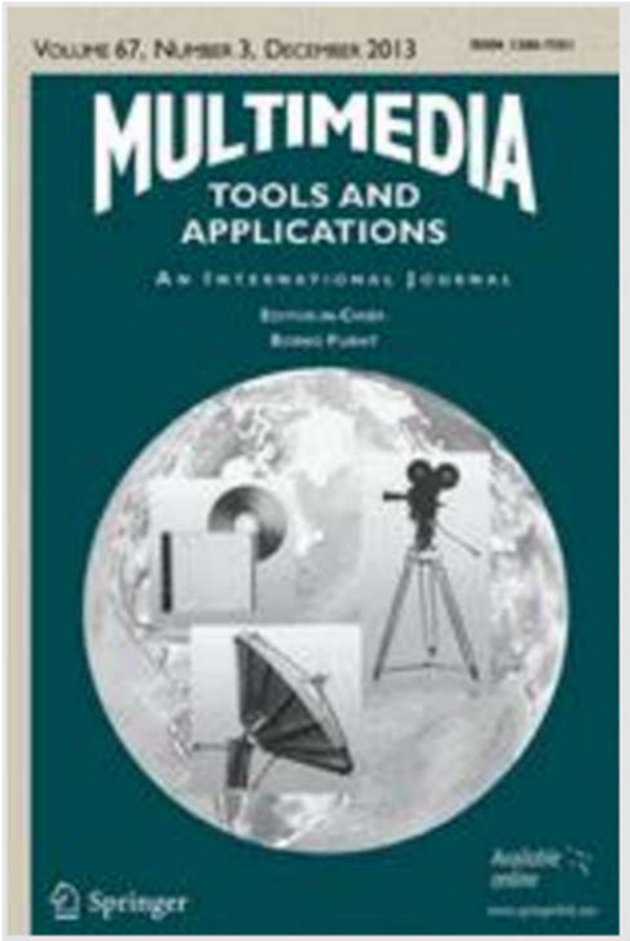
ISBN: 978-1-4799-3724-0/14/$31.00 ©2014 IEEE

Fig. 15. Hand posture sets.

TABLE IV. FEATURE CODES

Hand posture	F_1	F_2	F_3	F_4	F_5	F_{12}	F_{13}	F_{14}	F_{15}	F_{23}	F_{24}	F_{25}	F_{34}	F_{35}	F_{45}	H_r	H_m
A	4	4	4	4	4	6	6	6	6	5	6	6	5	6	5	0	0
B	4	1	1	1	1	6	6	6	6	5	6	6	5	6	5	0	0
C	4	2	2	2	2	6	6	6	6	5	6	6	5	6	5	0	0
D	4	1	4	4	4	6	5	6	6	6	6	6	5	6	5	0	0
E	4	4	4	4	4	6	6	6	5	5	6	6	5	6	5	0	0
F	4	4	1	1	1	5	6	6	6	6	6	6	5	6	5	0	0
G	4	1	4	4	4	6	6	6	6	6	6	6	5	6	6	8	0
H	4	1	1	4	4	6	6	6	6	5	6	6	6	6	5	8	0
I	4	4	4	4	1	5	6	6	6	5	6	6	5	6	6	0	0
J	4	4	4	4	1	5	6	6	6	5	6	6	5	6	6	8	0
K	4	1	1	4	4	6	6	6	6	6	6	6	6	6	5	0	0
L	1	1	4	4	4	6	6	6	6	6	6	6	5	6	5	0	0
M	4	3	3	3	4	6	6	6	5	5	6	6	5	6	6	0	0
N	4	3	3	4	4	6	6	5	6	5	6	6	6	6	5	0	0
O	4	4	4	4	4	5	6	6	6	5	6	6	5	6	5	0	0
P	1	1	3	4	4	6	6	6	6	6	6	6	6	6	5	0	0
Q	3	3	4	4	4	6	6	6	6	6	6	6	5	6	5	0	0
R	4	1	1	4	4	6	6	6	6	7	6	6	6	6	5	0	0
S	4	4	4	4	4	7	5	6	6	5	6	6	5	6	5	0	0
T	4	3	4	4	4	6	5	6	6	6	6	6	5	6	5	0	0
U	4	1	1	4	4	6	6	6	6	5	6	6	6	6	5	0	0
V	4	1	1	4	4	6	6	5	6	6	6	6	6	6	5	0	0
W	4	1	1	1	4	6	6	6	5	6	6	6	6	6	6	0	0
X	4	2	4	4	4	6	5	6	6	6	6	6	5	6	5	0	0
Y	1	4	4	4	1	6	6	6	6	5	6	6	5	6	6	0	0
Z	4	1	4	4	4	6	5	6	6	6	6	6	5	6	5	0	9
1	4	1	4	4	4	6	6	6	6	6	6	6	5	6	5	0	0
2	4	1	1	4	4	6	6	5	5	6	6	6	5	6	5	0	0
3	1	1	1	4	4	6	6	6	6	6	6	6	6	6	5	0	0
4	4	1	1	1	1	6	6	6	6	6	6	6	6	6	6	0	0
5	1	1	1	1	1	6	6	6	6	6	6	6	6	6	6	0	0

ISBN: 978-1-4799-3724-0/14/$31.00 ©2014 IEEE


Volume 67, Number 3, December 2013
MULTIMEDIA
TOOLS AND
APPLICATIONS
An International Journal
Springer
Available online

Thai alphabets finger-spelling recognition system

Kittasil Silanon[a]*, Nikom Survornvon[b]

Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Songkhla, Thailand 90112

Abstract

In this paper, we introduce a method for the Thai alphabets finger-spelling recognition system. The main objective is to help the deaf and non-vocal persons to improve their skill on finger-spelling. Several researches have been proposed. Most of them are based on hand posture estimation techniques. Therefore, we propose an alternative flexible hand posture estimation method using fuzzy finger shape and hand appearance features. By using depth image, the hand region is first extracted, and then the initialized features, such as fingertip, centre point, and palm size, have been calculated. The concept of active contour using energy function is implemented in order to track fingertip's position in the frame image sequence. To discriminate the hand posture sets, a hand feature definition have been established, which is composed of fuzzy finger shape and hand appearance features. The features are defined as chain code to represent hand posture. Since we focus on Thai alphabet finger-spelling, thus recognition process will be performed in two steps. Firstly, the American finger-spelling is recognized by the chain code feature using a scored voting method. Secondly, the Hidden Markov Model (HMM) method is used to build a learning model to recognize the sequence of the American finger-spelling hand postures and to provide the Thai alphabet finger-spelling. The performance of the recognition system can be measured in real time at around 68% recognition rate.

Keywords: Finger-Spelling; Hand Posture Estimation; American finger-spelling; Thai alphabet finger-spelling

1. Introduction

In this paper, we emphasize on the Thai sign language recognition system. For the sign language system, there are two main categories as follow: 1) word-level vocabulary signs, which are the signs of the hands' shape, orientation and movement of the hands, arms or body, and facial expressions simultaneously to convey meanings, 2) finger-spellings, which use fingers' pose to spell the alphabets of the word in a spoken language for communicating names, places, technical terms and etc. However, most of deaf and non-vocal persons, especially children, have problems with finger-spelling skills. Usually, the word-level vocabulary signs have

* Corresponding author. Tel.: +6681 698 9318
E-mail address: kittasil.silanon@gmail.com

been used to communicate each other, but the finger-spelling is used only 7% to 10% of daily communications. Evidently, the finger-spelling skills lag far behind the sign language skills. In order to help these people improve their skills, many systems specific to a language are proposed, as for example the American (ASL) [16,22,24], British (BSL) [17], Australian (Auslan) [14], Chinese (CSL) [30,31], Japanese (JSL) [8,33], Thai (ThSL) [3,21,23,27,28,32] finger-spellings and etc [2,6,15,18]. Therefore, in this paper, we have focused on the Thai alphabet finger-spelling. Actually, various researches on this topic have been proposed and most of them, including the Thai alphabets finger-spelling and other language finger-spellings, will be based on hand posture estimation techniques [1,19,26]. The hand posture estimation technique has two main approaches: 1) using the signal from a glove sensor device, and 2) using a vision-based method to estimate the hand posture. For the Thai alphabets finger-spelling, both methods have been introduced. Saengsri [23] proposed a Thai alphabets finger-spelling by using the data glove, motion tracker and Neural Network theory to improve the accuracy of the system. Kanjanapatmata [27] presents an image recognition method for the Thai alphabet using a polar orientation histogram of the hand image and an artificial Neural Network. Veerasakulthong [28] introduces a simple colour hand glove and appearance features. Sakulsujirapa [3] presents an appearance features lookup table to analyze the hand posture pattern for identifying Thai alphabets in finger-spelling. Sirboonruang [32] proposed a method combining the Zernike moment and wavelet moment to capture hand's features and using fuzzy classification algorithm to classify Thai finger-spelling hand postures. Phitakwinai [21] developed the 15 Thai finger-spelling alphabets and 10 words of the Thai sign language translation system using the scale invariant feature transform (SIFT). Moreover, other world-level researches have proposed more methods. Visarnkuna [27] proposed a Thai sign language recognition system using the Hidden Markov Model (HMM) for transcribing the human sign language into text or speech. The system estimates hand postures using an electronic glove called "CyberGlove". Auephanwiriyakul [20] proposed a Thai word sign language translation using SIFT with HMM. Ditcharoen [25] proposed the Thai Sign to Thai Machine (TSTM), for translating the Thai sign language to text. Chanda [12,13] build the dynamic Thai sign language translation system using upright speed-up robust feature and dynamic time warping, as well as the C-Mean clustering.

Although several approaches are proposed for the Thai alphabets finger-spelling recognition, however, they cannot achieve the critical criteria, such as, accuracy, flexibility, device constraints, and to cover all Thai alphabets. Moreover, they are mostly offline systems that cannot run in real-time. Thus, our research goal is to develop an automatic real-time recognition system for the complete set of Thai alphabets finger-spelling. We applied a vision-based method for hand posture estimation, which uses both finger shape and hand appearance features, to finally recognize the Thai alphabet finger-spelling. The system consists of four main parts: 1) hand segmentation, to segment the region of interest (ROI) of the hand from image sequences and calculate initial features 2) key hand posture selection, to determine the key frame representing the hand posture of finger-spelling from image sequences, 3) hand feature definition, to define the finger shapes and hand appearance features and represent it as a chain code sequence, 4) recognition model, to recognize the Thai alphabet finger-spelling from hand features by simply using a scored voting method and the Hidden Markov Model. The system overview can be shown in Fig.1.

2. Hand segmentation

In this section, the method is focused on the segmentation of the region of interest of the hand from the depth image and extracts the hand features necessary for system initialization.

2.1. Hand segmentation

In this step, the RGB-D image sequence is acquired and used as input to our system, shown in Fig.2.a (color), Fig.2.b (depth). The depth information can be defined simply by 3D vector *(x,y,z)*, where the *x* and *y* values are the spatial image coordinates that correspond to the row and column respectively, and *z* is the relative depth value. In our experimentation, the right hand side executes the finger spelling sign language. The depth image in Fig.2.b provides a strong cue for differentiating between the hand and the complex background (the closer object is of darker intensity). Thus, our system takes advantage of the depth information to extract only hand's region in image as show in Fig.2.c. After the ROI of hand is segmented, its initial features are then computed, such as, hand centre, fingertips position, palm size, and etc. The initialized features are used as referenced values for the incoming hand gesture processing in the finger-spelling recognition system.

2.2. Hand centre position

We compute the centre of hand's region using its moments M_{ij} which is defined as:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \qquad (1)$$

Where $I(x,y)$ is the intensity value at pixel (x,y) of the image, x and y are range over the hand's region. The center point of hand $C_0(X_c, Y_c)$ is calculated as the result shown in Fig.3.a.

$$Xc = \frac{M10}{M00}, \quad Yc = \frac{M01}{M00} \qquad (2)$$

The palm size *(R)* is defined by a cycle where its radius is the distance (*D*) between $C_0(X_c, Y_c)$ and the closest pixel (*P*) on hand's contour (*B*), as shown in Fig.3.b.

$$R = \underset{P \in B}{argmin}\{D(C_0, P)\} \qquad (3)$$

2.3. Fingertip position

In this stage, the fingertip and its valley points are detected, since the user is required to initialize the system by performing the pose of an "open" right hand facing to the camera, as shown in Fig.4.a. Certainly, it is effective to locate the fingertips from the curvature of boundary points from hand's region. Each point P_i has two parameters, the angle (θ_i) and direction (D_i). The angle can be estimated by using k-curvature profile [11] which is a method for calculating the angle of key point using two vectors with the same range *(k)*. The angle will be used to represent the curvature point as depicted in Fig.4.b. The candidate fingertip is defined with appropriated curvature values of threshold and positive direction $(D_i = 1)$. Inversely, the valley points are defined by its negative direction $(D_i = -1)$, which are located between fingertips, as shown in Fig.4.c. Next, the initialized fingertips are labelled, corresponding to the thumb, index, middle, ring, and pinky fingers respectively.

3. Key hand posture selection

In this section, we detail how to select the key frames of the hand posture that represents the finger-spelling alphabets in the image sequences. The main concept is that the initialled fingertips are tracked through the candidate fingertips of consecutive frames. The key hand posture is selected whenever the tracking fingertips are quasi still. Four steps are necessary in order to accomplish this concept.

3.1. Finger segmentation

In finger-spelling, the finger's movement is the key feature for recognition that must be segmented, as depicted in Fig.5.a. In our method, two conditions are defined in order to segment the finger locations from the depth image, as shown in Fig.5.b. Firstly, a stretching finger is segmented by distance condition (D),

$$D(p,C_0) > \tau_D \tag{4}$$

and secondly, bending finger is segmented by depth condition (Z),

$$|Z(p) - Z(C_0)| < \tau_z \tag{5}$$

Where τ_D and τ_Z are the pre-defined distance and depth threshold.

3.2. Candidate fingertips

By using the segmented image region of fingers, the candidate fingertips (Cf_i), as key points, are extracted. For the stretching fingers, we assume that the positions of fingertips should be located on its contour. Accordingly, the polygon approximation technique [34] is applied to the contour to extract the key points, as shown in Fig.6.b (point on contour). In addition, the candidate fingertips can be defined as the closest points to the camera. Thus, the depth value is used to extract the additional candidate fingertips, shown in Fig.6.b (point in contour).

3.3. Fingertip tracking

The fingertips tracking is a process for following the initialled fingertips, defined in the last section, through the current frame via its candidate fingertips. We introduced an active contour like-method [4,10] as an optimal tracking technique. For each fingertip, the cost functions with its possible candidate fingertips are established by trying to maximize its energy cost as following:

$$E(Cf_i) = E_{internal}(Cf_i) + E_{external}(Cf_i) \tag{6}$$

Where $E_{internal}$ represents the internal energy of the candidate due to bending or stretching of the finger and $E_{external}$ is the external constraint introduced by user. The internal energy is defined as:

$$E_{internal} = E_{continuity} + E_{curvature} + E_{depth} + E_{direction} \tag{7}$$

The $E_{continuity}$ is the energy of continuity that forces the candidate fingertip to be continuous, meaning it should not change much from the current point. Therefore, it tries to equalize the distance between the average distance ($\bar{d}$) and candidate fingertip, defined as the following equation:

$$E_{continuity} = \bar{d} - \left\| f_j - Cf_i \right\|^2 \tag{8}$$

The $E_{curvature}$ is the energy of smoothness which considers the contour's curvature, defined by angles between the any two closest contour vectors, $\mathbf{A} = (x_i - x_{i-k}, y_i - y_{i-k})$ and $\mathbf{B} = (x_{i+k} - x_i, y_{i-k} - y_i)$, given by:

$$E_{curvature} = \cos^{-1} \frac{\mathbf{A}.\mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|} \tag{9}$$

where k is the constraint. Any candidate fingertips which have the curvature value under a predefined threshold will be considered as the possible candidate fingertips. The E_{depth} is the energy of depth value representing the distance between candidate fingertips and camera. Its value increases when the candidate fingertips get close to camera. The $E_{direction}$ is the energy of direction that favorites the fingertips movement between consecutive frames that have the same direction. Supposing that $\mathbf{f}_j$ and $\mathbf{c}_{fi}$ are the vector of the current and candidate fingertip respectively, the energy is defined as:

$$E_{direction} = \overrightarrow{\mathbf{f}_j \mathbf{c}_{fi}} \tag{10}$$

The $E_{external}$ is an external energy that is defined by the distances between a considering fingertip to any candidate fingertips. A shorter distance will increase the external energy of that corresponding fingertip matching. In order to reduce the tracking error due to losing depth value, a low-pass smoothness filter is applied, so the distance between the current to the selected fingertip is then estimated. The pseudo-code of the tracking process for the basic finger's movements (bending, closing, crossing, rotation) is shown in table.1 and Fig.7.

3.4. Key hand posture

The main idea for the key hand posture selection is that the fingertip's displacement is quasi still. The frame differencing method is applied in order to detect the temporal difference between fingertips in successive images, as described by the following equation:

$$D_k(x,y) = \sum_{i=0}^{n} \left| P_{k-1}(x,y) - P_k(x,y) \right|_i \le \tau \tag{11}$$

A frame with minimal different value will be selected as the key hand posture.

4. Hand feature definition

The definition of hand features is detailed in this section. Two types of characteristics are used in order to obtain the invariant and robust features: finger shape feature and hand appearance feature. All features will be encoded as a chain code feature sequence to represent the hand posture.

4.1. Fuzzy finger shape feature

Finger shapes will be used to discriminate between different sets of hand postures. For reducing the computation complexity, there are four basic finger shapes (Open, Bend, Point and Close), as shown in Fig.8. In this step, fuzzy logic [5,7] is used to classify finger shapes. The process of fuzzy logic is explained in Fig.9. Firstly, a crisp set of input data are gathered and converted to fuzzy set using linguistic variable, fuzzy linguistic terms and membership functions (μ). This step is called the fuzzification. Subsequently, an inference is made for the evaluation result based on a set of rules. Finally, the resulting fuzzy output is mapped to a crisp output using membership functions, in the defuzzification step. In our system, the linguistic variables are defined as depth (D), distance (Dist) and shape(S). The depth (depth value at fingertip position) and distance (distance from fingertip position to hand center point) are input variables, but shape is an output variable. For linguistic terms, depth and distance variables are decomposed into a set of "Near", "Middle" and "Far" terms, *D(t) = {Near, Middle, Far} and Dist(t) = {Near, Middle, Far}*. The range of each term is shown in Fig.10. For the shape variable, linguistic terms are open, bend and point *S(t) = {Open, Bend, Point}*. We do not include "close" into shape linguistic terms because "close" shape will be classified immediately when fingertip position is in the palm region. We use the trapezoidal curve as the type of the membership function of each linguistic term. The fuzzification will be performed to convert all crisp input to fuzzy set input as shown in Fig.11. The rules are defined in order to control the output variable. The fuzzy rule is a simple IF-THEN rule with a condition and conclusion. In table.2 there is a sample fuzzy rule for the finger shape classification. Table.3 shows the matrix representation of the fuzzy rules. Row captions contain the value that depth can take, column captions contain the value for distance, and each cell is the resulting shape when the input variables take the values in that row and column. For example, the cell (3,2) in matrix can be read as follows: IF distance is middle AND depth is near THEN shape is point. The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations, this process is called inference. The mostly used operations for fuzzy logic are OR*(max)*, AND*(min)* and NOT*(complement)*, respectively. After evaluating the result of each rules and combining them to obtain a final result, the defuzzification is performed according to membership function of fuzzy output variable. For instance, we have the result in Fig.12 at the end of inference. In this Fig.12., the shaded areas are belonging to the fuzzy result. This result has to be defuzzified to obtain a final crisp output. There are different algorithms for the defuzzification. We have used the center of gravity (COG) to map the fuzzy output to be a crisp output, represented with a dot in the Fig.12. The COG algorithm can be written as:

$$COG = \frac{\sum_{i=1}^{N} x\mu(x)}{\sum_{i=1}^{N} \mu(x)} \tag{12}$$

4.2. Hand Appearance Feature

Considering only the finger shapes is not sufficient to estimate the hand posture for finger-spelling. Generally, there are some hand postures that have similarities in finger shape, but different configurations.

The hand configuration consists of finger relation, hand rotation and movement. Considering the Figs.13.a and 13.b, even though both hand postures have the same finger shape (all finger are open), they have, however, distinct finger relations. In the Fig.13.a, the finger relation is closed. In the Fig.13.b., the finger relation is separated. Thus, we have defined the finger relation as "close", "separate" and "cross". Moreover, hand rotation and hand movement will be considered to discriminate different hand posture sets. All sets of hand configuration in finger-spelling system as show in Fig.14.

4.3. Chain Codes Feature

To represent the hand posture, each of the finger shapes and finger relation will be encode as shown in table.4. In our system, the feature vector is composed of 17 values defined as shown in Fig.15. The first five elements (f_1,f_2,f_3,f_4,f_5) indicate the finger shape (thumb, index, middle, ring and pinky respectively). In Fig 15, (f_1,f_2,f_3,f_4,f_5) = (4,1,1,4,4) means that the thumb is "close", index and middle are "open", ring and pinky are "close", respectively. The next four elements (f_{12} ,f_{13},f_{14},f_{15}) describe the relation between thumb with index, middle, ring and pinky fingers. In Fig.15., (f_{12},f_{13},f_{14},f_{15}) = (6,6,6,6) indicates that all relations of thumb with other fingers are "separate". The next three elements (f_{23},f_{24},f_{25}) describe the relation of index with middle, ring and pinky fingers. The next two elements (f_{34},f_{35}) describe the relation of middle with ring and pinky fingers. In Fig.15., (f_{34},f_{35}) = (6,6) indicates that all relations of middle with other fingers are "separate". The element (f_{45}) describes the relation of ring and pinky. In Fig.15., (f_{45}) = (5) indicates that the relation of ring with pinky finger is "close". The next two elements (H_r,H_m) represent the rotation and movement of hand. In Fig.15., (H_r,H_m) = (0,0) indicates that there is no rotation or movement for this hand posture. Finally, the vector feature of the hand posture in Fig.15. have the chain codes as P = {4,1,1,4,4,6,6,6,6,6,6,6,6,6,5,0,0}.

5. Recognition model

In this section, we mention about finger-spelling recognition process. Since the Thai alphabet finger-spelling is based on the American finger-spelling, therefore, the recognition model will consist of two major processes.

5.1. Vote Recognition

The vote method is used in our recognition process. This method is simple and effective and can run in real time. Firstly, the *n* patterns of the hand posture chain codes (P_i | i = 1...N) are defined with an off-line computation. Each pattern represents a finger-spelling that is the chain code of finger shape features and hand appearance features. Secondly, any hand posture, which is captured from the camera, will be detected and encoded into the same type of chain codes. Thirdly, the similarity measurement between the detected pattern and the *n* pre-defined patterns are performed by using a simple vote method. This technique will compare each element of the input chain codes with the corresponding element of the pattern chain code. A score will be given if any elements have matched. Thus, the hand posture will be recognized to the pattern which gives the maximum score. This technique is applied for the American finger-spelling recognition system. There are 31 template hand postures that consist of 26 hand postures for the American-English alphabets (A-Z) and five hand postures for 1-5 numbers. Each template of hand posture is shown in Fig.16., and the chain code in table.5.

5.2. HMM learning model

The Thai finger-spelling derives from the hand postures of the American finger-spelling. For example, "ก (ko kai)" use "K" (K), "ข (kho khai)" use combination of "K" and "1" (K+1). The 42 hand posture sequences for each Thai alphabet finger-spelling can be shown in Fig.21. and table.9. For this recognition step, a learning-based approach Hidden Markov Model (HMM) [9] is used. HMM are stochastic processes which can be used to model any time series data. Therefore, HMM is useful for the recognition of the Thai finger-spelling, which can be viewed as a series of hand postures in the American finger-spelling. The basic elements of HMM can be expressed as following:

1). A set of learning states $S = \{S_1, S_2, \ldots, S_N\}$: the number of states is determined by using the maximum number of hand postures involved in performing a Thai finger-spelling (e.g. "ธ" (tho thong) in table.9 uses three hand postures), which consist of three states for hidden states and additional two states for initialization and finalization. Therefore, a five-state model with transitions was chosen for the system (N=5).

2). A set of observation symbols $V = \{V_1, V_2, \ldots V_M\}$: observation symbols are represented by 31 hand postures in the American finger-spelling (M=31). All observation symbols are listed in table.6. The hand posture sequence will be converted to observation symbols sequence for using as input of learning model. For instance, the "ก (ko kai)" alphabet uses "K". This sequence is converted to symbol "11". The "ข (kho khai)" alphabet use "K+1". Thus, the sequence is converted to sequence symbol of "11 and 27" etc.

3). The state transition matrix $A = \{a_{ij}\}$, where a_{ij} is the transition probability of taking the transition from state "i" to state "j".

$$a_{ij} = P(q_t = S_j \mid q_{t-1} = S_i), \quad 1 \le i, j \le N \tag{13}$$

4). The probability distribution matrix of the observation symbol $B = \{b_j(k)\}$, where $b_j(k)$ gives the probability of emitting observation symbol o_k from state "j".

$$b_j(k) = P(o_t = V_k \mid q_t = S_j), \quad 1 \le j \le N; \; 1 \le k \le M \tag{14}$$

5). The initial state distribution matrix $\pi = \{\pi_i\}$, where π_i is the initial probability of all states in model.

$$\pi_i = P(q_1 = S_i), \quad 1 \le i \le N \tag{15}$$

6). The HMM topology, fully connected topology (Ergodic model), where any state can be reached from other states, is applied for our system.

The HMM is trained to the model 42 Thai alphabets finger-spelling (λ_k) by the training data input. A combination of hand postures in the American finger-spelling are converted to be observation symbol sequence ($O = V_1, V_2, \ldots, V_T$) and input to each HMM model for calculating the probability $P(O|\lambda_k)$. The models that give the maximum probability will be recognized as result of the input observation symbol sequence. The recognition process can be shown in Fig.17.

6. Experiment result

In this section, we discuss about recognition process result. There are two main recognition results. Those are American finger-spelling recognition result and alphabet finger-spelling recognition result.

6.1. American finger-spelling recognition

Each hand posture for the American finger-spelling is tested at 100 rounds. The result of recognition is shown in Fig.18. As the result in Fig.18, the best cases (100% recognition rate) of the system are the hand postures of numbers "3", "4" and "5". The hand postures of these three cases are not much different from the initial hand posture (Open hand). The movement of the finger does not change much. Thus, the finger tracking error will happen less. The worst case (less than 10% recognition rate) is the hand posture of "Z" alphabet. This hand posture has the rapid hand movement. Thus, the finger tracking error will occur when the hand posture has an immediate change-direction movement. For the general cases (≈65% recognition rate), including the "A-Y" and "1-2", the most of the hand postures have only a continuously finger movement, which does not change much from the current frame to the next frame. Hence, the finger tracking from the active contour concept can be quite effectively performed. The example of this process can be shown in Fig.19 and at [35].

6.2. Thai alphabets finger-spelling recognition

The 150 samples of each hand posture are collected, gathering 6,300 in total. For each hand posture data set, the 50 samples are used to generate each alphabet model and the 100 samples are used to test the performance of the recognition model. Forward-Backward procedure is used to calculate the probability of the input observation sequence. Since HMM does not have a fixed rule for the state number specification, therefore, we evaluate the number of HMM state with 5 and its variations (10 and 15 states) for the recognition rate comparison. The table.7 shows the results of 42 Thai alphabets recognition model. Although, in global the 10 states of HMM give in better result (69.52%) than 5 states. But, its improvement is insignificant (+0.62%). For the 15 states, HMM gives less recognition rate at 65.88%. We assume to use the five states that give average recognition rate at 68.90%. As the result, we infer that the alphabet models that use only one hand posture get a quite better result such as "ก" (ko kai), "จ" (cho chan, "ด" (do dek), "บ" (bo baimai), "พ" (po han), "ฟ" (fo fan), "ย" (yo yak), "ร" (ro ruea), "ล" (lo ling), "ว" (wo waen), "ห" (ho hip), "อ" (o ang). There are some exceptions, such as "ต" (to tao) that equal to "T", "ม" (mo ma) that equal to "M", "น" (no nue) that equal to "N" and "ส" (so suea) that equal to "S", which get a poor result (less than 60%) because the fingertip tracking cases for finger overlapping and adjacency give less precision. Hence, the other letters that are based on these alphabet groups ("T", "M", "N" and "S") will give a less good result as well, as the groups of alphabet which are based on "N" such as "ง" (ngo ngu) and "ณ" (no nen), the groups of alphabets based on "S" such as ศ" (so sala) and "ษ" (so rue-si), and the groups of alphabets based on "T" such as "ฏ" (to pa-tak), "ฐ" (tho than), "ฑ" (tho montho), "ฒ" (tho phu-thao), "ถ" (tho thung), "ท" (tho thahan) and "ธ" (tho thong). The real-time application example can be shown in Fig.20 and at [36]. Actually, there are many researches related to the Thai sign language, word-level and finger-spelling, which we have discussed in the introduction. However, our work has focused only on the finger-spelling. Therefore, the general comparison can be performed between previous methods and our system as show in table.7. We have compared the general conditions not only for our method but also for some researches that use other additional devices such as sensor glove, colour glove or depth sensor camera etc. Regarding the background, some researchers have to setup it to be a constant colour. In terms of the outfit, users are asked to wear long sleeves shirts. Concerning the number of the alphabets that can be recognized in the system, since we propose a vision-based method for hand posture estimation, the comparison result shows that it is hard to compare our system with a glove-based method that is propose in [23], because the depth image is not as good as signal from electronic sensor, especially when fingers occlude or stick together as we mentioned earlier. Although many Thai finger-spelling recognition systems [3,21,27,28] proposed are vision-based approaches, most of them use only appearance features such as point, contour, edge etc. In contrast, our proposed system considers features of the hand posture that includes the fingers shape, finger relation, hand movement and hand rotation features, which increase the capacity for hand posture discrimination. Besides, we do not need to segment the key frame before performing the alphabet recognition. Therefore, our process can be run in real-time situations. For the recognition rate, although our work yields a less significant result comparing with other techniques, however, we propose a method that recognizes all Thai alphabets finger-spelling; meanwhile the other systems provide only some groups of the Thai alphabets recognition.

7. Conclusion

In this paper, we presented a method that enables the estimation of the hand posture for the Thai alphabets finger-spelling recognition system. The depth image will be used for robust hand region segmentation and for removing the complex background. The active contour concept calculates the energy function to track the fingertip's position in the frame sequence. The finger shapes and hand appearance have been proposed to represent different hand posture sets. The finger shapes features are based on fuzzy logic. The hand appearance features consist of finger relation, hand rotation and hand movement. Since the Thai alphabets finger-spelling is based on the American finger-spelling, therefore, the recognition model will be performed in two steps. For the American finger-spelling, each hand posture will be encoded to a discrete chain codes based on finger shapes and hand configuration. The 31 templates hand posture chain codes for the American finger-spelling (26 hand postures for A-Z and five hand postures for 1-5 numbers) will be pre-defined and used to compare the similarities between pre-defined chain codes and input chain codes by using simple score voting. For the Thai alphabets finger-spelling, the learning-based method (Hidden Markov Model) is use to build 42 Thai alphabet models that recognizes the sequence of the American finger-spelling hand postures and provide the Thai alphabet finger-spelling. The alphabet hand posture recognition result is performed with five states HMM and provides an average recognition rate at 68.90%. The method does not only apply to the American finger-spelling or Thai alphabets finger-spelling. We expect that our method can be applied to other applications, as for example games, robot controlling or visual input devices etc. However, our main future work is to increase the speed and tracking accuracy of the fingertips. If the data of fingertip's position is precise, then the error should be decreased and greatly impact on the final performance of the system.

Acknowledgements

We would like to thank the National Research University Project of Thailand's Office of the Higher Education Commission for the financial support.

References

[1] A. Erol, et al., Vision-based hand pose estimation: A review , Computer Vision and Image Understanding 108 (2007) 52-73.

[2] A. Park, S. Yun, J. Kim, S. Min and K. Jung, Real-Time Vision-based Korean Finger Spelling Recognition System, Engineering and Technology 34, 293-298, (2008).

[3] B. Sakulsujirapa, M. Karnjanadecha, A. Choksuriwong, Thai Sign Language Recognition Based on Finger Pattern Analysis, In International Conference on Embedded Systems and Intelligent Technology (ICESIT2011), Phuket, Thailand, 9th - 11th February 2011, pages 16.

[4] D. J. Williams, M. Shah, A Fast algorithm for active contours and curvature estimation, In: CVGIP: Image Understanding, vol. 55, no. 1, pp. 14-26 (January 1992)

[5] E. Cox, "Fuzzy fundamentals," Spectrum, IEEE, vol.29, no.10, pp.58-61, Oct. 1992

[6] F. Isikdogan, S. Albayrak, Automatic recognition of Turkish fingerspelling, Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on , vol., no., pp.264-267, 15-18 June 2011.

[7] J.M. Mendel, "Fuzzy logic systems for engineering: a tutorial," Proceeding of the IEEE, vol.83, no.3, pp.345-377, Mar 1995.

[8] K. Fujimura, L. Xia, Sign recognition using depth image streams, Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on , vol., no., pp.381-386, 2-6 April 2006.

[9] L.R. Rabiner, "A tutorial on hidden markov models and selected application in speech recognition," Proc. IEEE 77, 1989, pp 267-293.

[10] M. Kass, A. Witkin, D. Terzopoulos: Snakes: Active contour models. In: International journal of computer vision, vol. 1, no. 4, pp. 321-331 (1988).

[11] M. Zhenyao, N. Ulrich, Real-time Hand Pose Recognition Using Low-Resolution Depth Images, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006 , vol.2, pp. 1499- 1505, 2006.

[12] P. Chanda, S. Auephanwiriyakul, N. Theera-Umpon, Thai sign language translation system using upright speed-up robust feature and dynamic time warping, Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on , vol.2, no., pp.70,74, 25-27 May 2012.

[13] P. Chanda, S. Auephanwiriyakul, N. Theera-Umpon, Thai sign language translation system using upright speed-up robust feature and c-means clustering, Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on , vol., no., pp.1,6, 10-15 June 2012.

[14] P. Goh, E.J. Holden, Dynamic Fingerspelling Recognition using Geometric and Motion Features, Image Processing, 2006 IEEE International Conference on , vol., no., pp.2741-2744, 8-11 Oct. 2006.

[15] R. Akmeliawati, et al., Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network, Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE , vol., no., pp.1-6, 1-3 May 2007.

[16] R. S. Feris, M. Turk, R. Raskar, T. Karhan; G. Ohashi, Exploiting Depth Discontinuities for Vision-Based Fingerspelling Recognition, Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on , vol., no., pp. 155, 27-02 June 2004.

[17] S. Liwicki, M. Everingham, , "Automatic recognition of fingerspelled words in British Sign Language," Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on , vol., no., pp.50-57, 20-25 June 2009.

[18] S. Min, et al., Simple Glove-Based Korean Finger Spelling Recognition System, Lecture Notes in Computer Science 4705, 1063. (2007).

[19] S.C.W. Ong, S. Ranganath, Automatic sign language analysis: a survey and the future beyond lexical meaning, Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.27, no.6, pp. 873- 891, Jun 2005.

[20] S. Auephanwiriyakul, et al., Thai sign language translation using Scale Invariant Feature Tranform and Hidden Markov Models, Pattern Recognition Letters 34(11): 1291-1298 (2013).

[21] S. Phitakwinai, S. Auephanwiriyakul, N. Theera-Umpon, Thai Sign Language Translation Using Fuzzy C-Means and Scale Invariant Feature Transform, Lecture Notes in Computer Science, Vol 5073, pp. 1107 – 1119, 2008.

[22] S. Ricco, C. Tomasi. Fingerspelling, Recognition through Classification of Letter-to-Letter Transitions, 9th Asian Conference on Computer Vision, pages 214--225, September 2009.

[23] S. Saengsri, V. Niennattrakul, C.A. Ratanamahatana, TFRS: Thai finger-spelling sign language recognition system, Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on , vol., no., pp.457,462, 16-18 May 2012.

[24] T. B. Dinh, et al., Hand gesture classification using boosted cascade of classifiers, Proceedings of the 4th IEEE International Conference on Research, Innovation and Vision for the Future, RIVF'06, p. 139-144, 2006

[25] T. Ditcharoen, N. Cercone, K. Naruedomkul, B. Tipakorn, TSTMT: step towards an accurate Thai sign translation, Machine Learning and Applications, 2005. Proceedings. Fourth International Conference on , vol., no., pp.6 pp.., 15-17 Dec. 2005

[26] V.I. Pavlovic, R. Sharma, T.S. Huang, Visual interpretation of hand gestures for human-computer interaction: a review, Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.19, no.7, pp.677,695, Jul 1997

[27] W. Kanjanapatmata, Sign language gesture recognition for Thai consonant using artificial neural Networks, King Mongkut's University of Technology North Bangkok, Bangkok, 2005.

[28] W. Veerasakulthong, Thai Hand Sign Recognition, M. Science. Thesis, National Institute of Development Administration, Bangkok, Thailand, 2007.

[29] W. Visarnkuna, Sign Language Recognition System by Hidden Markov Models, King Mongkut's University of Technology Thonburi, Bangkok, 2004.

[30] W. Jiangqin, G. Wen, The recognition of finger-spelling alphabets for Chinese sign language, the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction, 2001.

[31] X. Teng, et al., A hand gesture recognition system based on local linear embedding, Journal of Visual Languages & Computing Volume 16, Issue 5, October 2005, Pages 442-454.

[32] Y. Sriboonuang, P. Kumhom, K. Chamnongthai, Hand posture classification using wavelet moment invariant, Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2004. (VECIMS). 2004 IEEE Symposium on , vol., no., pp.78,82, 12-14 July 2004.

[33] Y. Tabata, T. Kuroda, Finger spelling recognition using distinctive features of hand shape, In: 7th ICDVRAT with Art Abilitation. Maia, Portugal: 2008:287-292.

[34] Ramer-Douglas-Peucker algorithm, June 25, 2014, Available at (online): http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorith

[35] Finger-Spelling Recognition System using Fuzzy Finger Shape and Hand Appearance Features, May 10, 2014, Available at (online): http://www.youtube.com/watch?v=qW7i6TyVBaM

[36] thai alphabets finger spelling, July 15, 2014, Available at (online): http://www.youtube.com/watch?v=tn7J0k1KZis&feature=youtu.be

Figures.

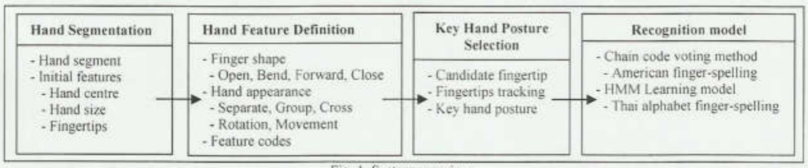
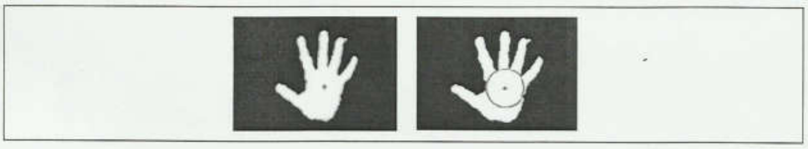


Fig. 1. System overview.

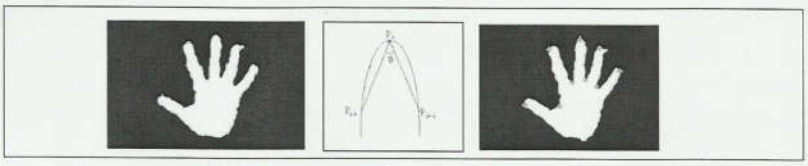
(a) (b) (c)

Fig.2. RGB-D Information: (a) colour image (b) depth image (c) hand segmented



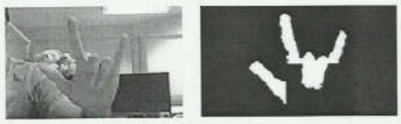
(a) (b)

Fig.3. Hand feature initialization: (a) hand center (b) palm size



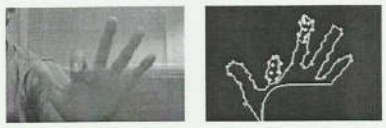
(a) (b) (c)

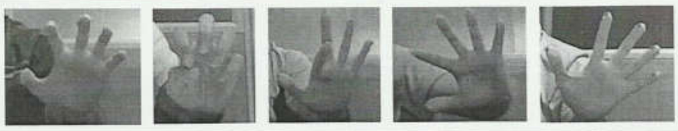
Fig.4. Fingertips positions: (a) open hand (b) curvature calculation (c) fingertip and valley points between fingertips.



(a) (b)

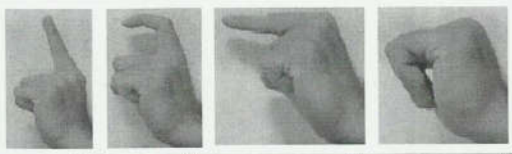
Fig.5. Locating fingers: (a) origin hand posture (b) extraction finger location using pre-defines conditions.





(a)* (b) (c) (d) (e)

Fig.7. Basic fingertip movements: (a) bending finger (b) moving finger into the palm (c) crossing finger (d) 45^0 hand rotation.



(a) (b) (c) (d)

Fig.8. Four basic finger shapes: (a) Open (b) Bend (c) Point (d) Close

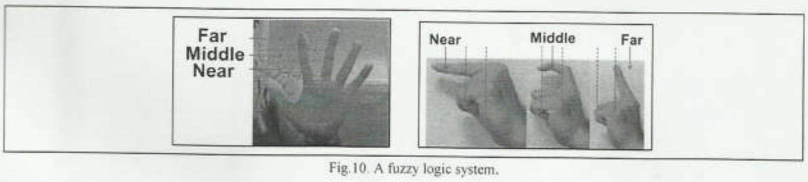
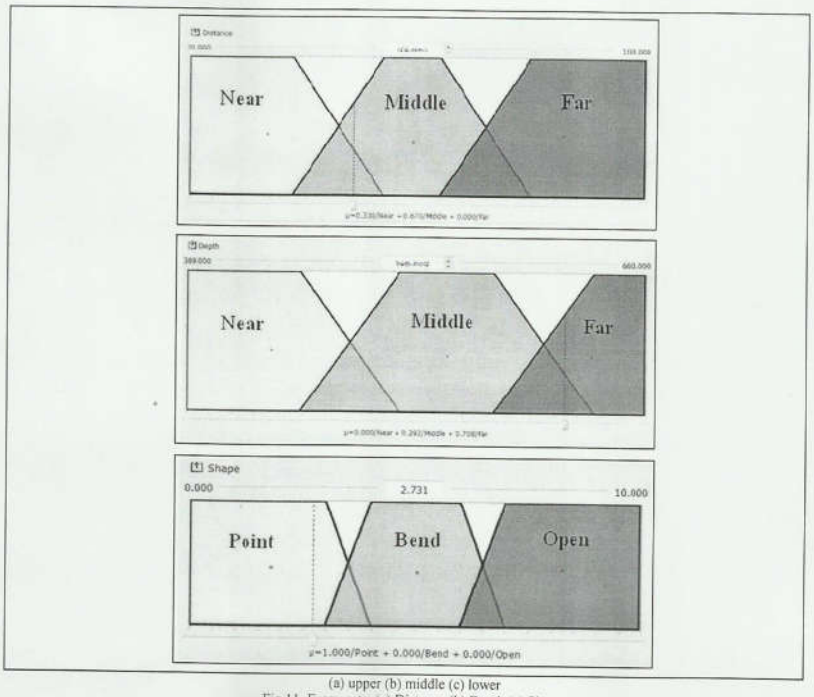


Fig.10. A fuzzy logic system.



(a) upper (b) middle (c) lower

Fig.11. Fuzzy sets: (a) Distance (b) Depth (c) Shape.

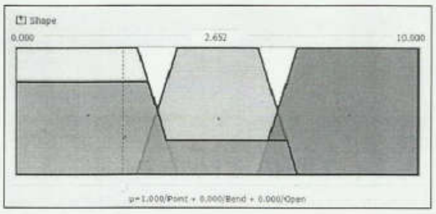
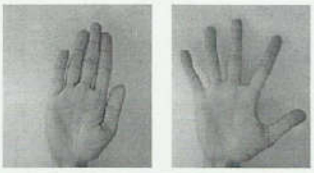
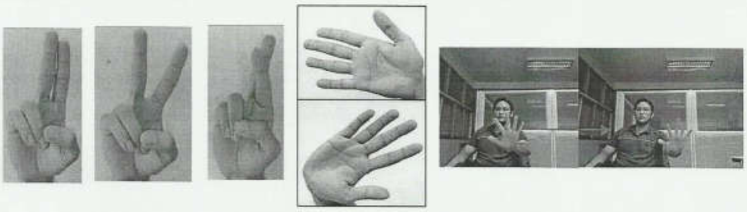


Fig.12. Fuzzy set result.



(a) (b)

Fig.13. Finger relation: (a) Close (b) Separate.



(a) (b) (c) (d) (e)

Fig.14. Hand appearance: (a) Close (b) Separate (c) Cross (d) Rotation (e) Movement.

Hand posture feature chain code

f_1	f_2	f_3	f_4	f_5	f_{12}	f_{13}	f_{14}	f_{15}	f_{23}	f_{24}	f_{25}	f_{34}	f_{35}
4	1	1	4	4	6	6	6	6	6	6	6	6	6

f_{45}	H_r	H_m
5	0	0

Fig.15. Hand posture feature chain code.

Fig.16. Hand posture feature chain code.

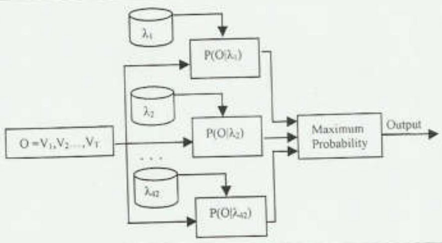


Fig.17. HMM recognition process

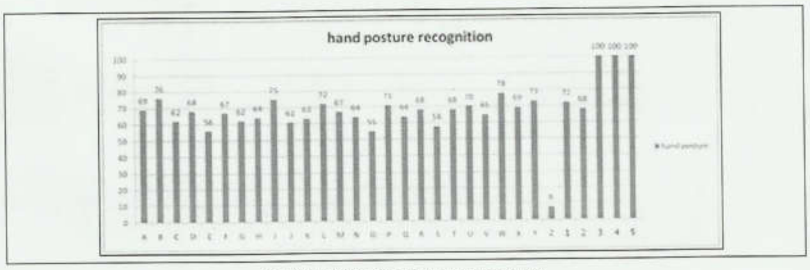


Fig.18. American finger-spelling recognition rate.

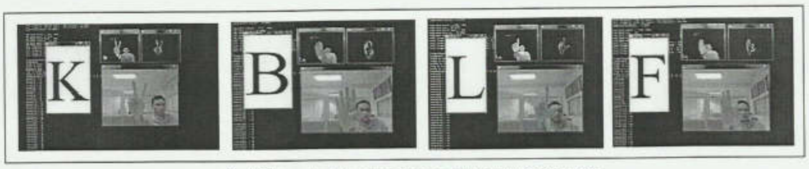


Fig.19. Example of American finger-spelling recognition system.

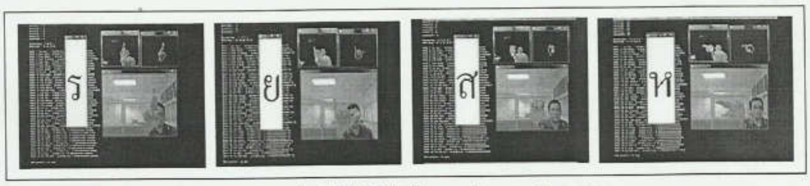


Fig.20.Example of Thai alphabet finger-spelling recognition system.

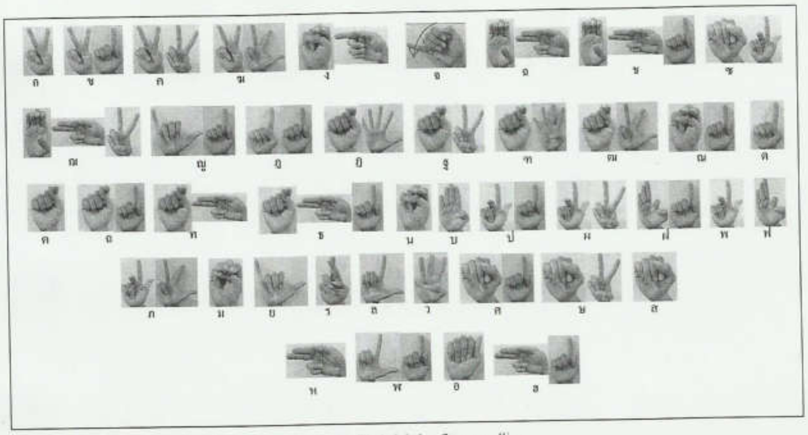


Fig.21. Thai alphabet finger-spelling.

Table 4.

Feature codes

Finger shape	Hand appearance	Hand appearance
Open = 1	Group = 5	Movement = 9
Bend = 2	Separate = 6	No-Movement = 0
Point = 3	Cross = 7	No-Rotation = 0
Close = 4	Rotation = 8	-

Table 5.

Hand posture feature chain code

Hand posture	Feature chain codes	Hand posture	Feature chain codes
A	4,4,4,4,4,6,6,6,6,5,6,6,5,6,5,0,0	Q	3,3,4,4,4,6,6,6,6,6,6,6,5,6,5,0,0
B	4,1,1,1,1,6,6,6,6,5,6,6,5,6,5,0,0	R	4,1,1,4,4,6,6,6,6,7,6,6,6,6,5,0,0
C	4,2,2,2,2,6,6,6,6,5,6,6,5,6,5,0,0	S	4,4,4,4,4,7,5,6,6,5,6,6,5,6,5,0,0
D	4,1,4,4,4,6,5,6,6,6,6,6,5,6,5,0,0	T	4,3,4,4,4,6,5,6,6,6,6,6,5,6,5,0,0
E	4,4,4,4,4,6,6,6,5,5,6,6,5,6,5,0,0	U	4,1,1,4,4,6,6,6,6,5,6,6,6,6,5,0,0
F	4,4,1,1,1,5,6,6,6,6,6,6,5,6,5,0,0	V	4,1,1,4,4,6,6,5,6,6,6,6,6,6,5,0,0
G	4,1,4,4,4,6,6,6,6,6,6,6,5,6,6,8,0	W	4,1,1,1,4,6,6,6,6,5,6,6,5,6,6,0,0
H	4,1,1,4,4,6,6,6,6,5,6,6,6,6,5,8,0	X	4,2,4,4,6,5,6,6,6,6,6,6,5,6,5,0,0
I	4,4,4,4,1,5,6,6,6,5,6,6,5,6,6,0,0	Y	1,4,4,4,1,6,6,6,6,5,6,6,5,6,6,0,0
J	4,4,4,4,1,5,6,6,6,5,6,6,5,6,6,8,0	Z	4,1,4,4,4,6,5,6,6,6,6,6,5,6,5,0,9
K	4,1,1,4,4,6,6,6,6,6,6,6,6,6,5,0,0	1	4,1,4,4,4,6,6,6,6,6,6,6,5,6,5,0,0
L	1,1,4,4,4,6,6,6,6,6,6,6,5,6,5,0,0	2	4,1,1,4,4,6,6,5,5,6,6,6,5,6,5,0,0
M	4,3,3,3,4,6,6,6,5,5,6,6,5,6,6,0,0	3	1,1,1,4,4,6,6,6,6,6,6,6,6,6,5,0,0
N	4,3,3,4,4,6,6,5,6,5,6,6,6,6,5,0,0	4	4,1,1,1,1,6,6,6,6,6,6,6,6,6,6,0,0
O	4,4,4,4,4,5,6,6,6,5,6,6,5,6,5,0,0	5	1,1,1,1,1,6,6,6,6,6,6,6,6,6,6,0,0
P	1,1,3,4,4,6,6,6,6,6,6,6,6,6,5,0,0	-	-

Table 6.

Hand posture feature chain code

Hand posture	Observation symbol	Hand posture	Observation symbol
A	1	Q	17
B	2	R	18
C	3	S	19
D	4	T	20
E	5	U	21
F	6	V	22
G	7	W	23
H	8	X	24
I	9	Y	25
J	10	Z	26
K	11	1	27
L	12	2	28
M	13	3	29
N	14	4	30
O	15	5	31
P	16	-	-

Table 7.

Hand posture feature chain code

Alphabet	Recognition rate (%)			Alphabet	Recognition rate (%)		
	5 state	10 state	15 state		5 state	10 state	15 state
ก (Ko kai)	82	84	79	ธ (tho thong)	57	61	62
ข (kho khai)	72	75	71	น (no nu)	63	62	65
ค (kho khwai)	74	72	69	บ (bo baimai)	84	86	81
ฆ (kho ra-khang)	75	77	70	ป (po pla)	76	74	73
ง (ngo ngu)	59	62	56	ผ (pho phueng)	73	75	71
จ (cho ching)	73	74	71	ฝ (fo fa)	72	76	70
ฉ (cho ching)	72	73	70	พ (pho phan)	77	78	75
ช (cho chang)	68	69	65	ฟ (fo fan)	78	76	74
ซ (so so)	52	51	45	ภ (pho sam-phao)	74	75	72
ฌ (cho choe)	69	72	67	ม (mo ma)	64	61	62
ญ (yo ying)	77	75	69	ย (yo yak)	75	76	74
ฎ (do cha-da)	74	75	65	ร (ro ruea)	72	71	68
ฏ (to pa-tak)	57	59	61	ล (lo ling)	75	77	74
ฐ (tho than)	65	67	63	ว (wo waen)	72	71	70
ฑ (tho montho)	62	63	57	ศ (so sala)	54	57	46
ฒ (tho phu-thao)	64	62	60	ษ (so rue-si)	52	51	43
ณ (no nen)	68	70	65	ส (so suea)	55	53	51
ด (do dek)	80	75	71	ห (ho hip)	73	76	71
ต (to tao)	61	59	55	ฬ (lo chu-la)	74	75	72
ถ (tho thung)	58	56	57	อ (o ang)	75	77	73
ท (tho thahan)	65	69	64	ฮ (ho nok-huk)	72	73	70
Avg. 5 state = 68.904%, 10 state = 69.528%, 15 state = 65.880%							

Table 8.

The general comparison.

work	Method	Device	Background	Outfit	Real time	#. for alphabet	Recognition rate
[23]	G-B	Sensor glove	No	No	Yes	16	94.44%
[27]	V-B	No	Yes	No	No	15	72%
[28]	V-B	Color glove	Yes	No	No	31	88.26%
[32]	V-B	No	Yes	No	No	N/A	72%
[3]	V-B	No	Yes	No	No	42	81.43%
[21]	V-B	No	Yes	Yes	N/A	15	79.90%
Our method	V-B	Depth camera	No	No	Yes	42	68.90%

*G-B = Glove-Based, *V-B = Vision-Based *N/A = Not Available

Table 9.

Thai alphabet hand posture sequence.

Alphabet	Posture	Alphabet	Posture	Alphabet	Posture
ก (ko kai)	K	ฒ (tho phu-thao)	T+3	ม (mo ma)	M
ข (kho khai)	K+1	ณ (no nen)	N+1	ย (yo yak)	Y
ค (kho khwai)	K+2	ด (do dek)	D	ร (ro ruea)	R
ฆ (kho ra-khang)	K+3	ต (to tao)	T	ล (lo ling)	L
ง (ngo ngu)	N+G	ถ (tho thung)	T+1	ว (wo waen)	W
จ (cho chan)	J	ท (tho thahan)	T+H	ศ (so sala)	S+1
ฉ (cho ching)	C+H	ธ (tho thong)	T+H+1	ษ (so rue-si)	S+2
ช (cho chang)	C+H+1	น (no nu)	N	ส (so suea)	S
ซ (so so)	S+P	บ (bo baimai)	B	ห (ho hip)	H
ฌ (cho choe)	C+H+2	ป (po pla)	P+1	ฬ (lo chu-la)	L+1
ญ (yo ying)	Y+1	ผ (pho phueng)	P+2	อ (o ang)	A
ฎ (do cha-da)	D+1	ฝ (fo fa)	F+1	ฮ (ho nok-huk)	H+1
ฏ (to pa-tak)	T+5	พ (pho phan)	P		
ฐ (tho than)	T+2	ฟ (fo fan)	F		
ฑ (tho montho)	T+4	ภ (pho sam-phao)	P+3		

VITAE

Name Kittasil Silanon

Student ID 5410130021

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Engineering (Computer Engineering)	Prince of Sonkla University	2009
Master of Engineering (Computer Engineering)	Prince of Sonkla University	2011

Scholarship Awards during Enrolment

Scholarship of Nation Research University, 2011, Faculty of Engineering, Prince of Songkla University, Songkla, Thailand

List of Publication and Proceeding

K. Silanon and N. Suvonvorn, "Fingertip Tracking Based Active Contour for General HCI Application," Proceedings of First International Conference on Advanced Data and Information Engineering (DaEng-2013), Lecture Notes in Electrical Engineering, Vol.285, 2014, pp. 309-316.

K. Silanon and N. Suvonvorn, "Finger Spelling Recognition System using Fuzzy Finger Shape and Hand Appearance Features," In the Fourth International Conference on Digital Information and Communication Technology and its Application (DICTAP2014), Thailand, May 6-8, 2014, pp.419-424.

K. Silanon and N. Suvonvorn, "Fuzzy Finger Shape and Hand Appearance Feature for Thai Alphabet Finger-Spelling Recognition System," Multimedia Tools and Applications (Submitted to Journal of Multimedia Tools and Applications).