UNCONSTRAINT LAO HANDWRITTEN CHARACTER RECOGNITION USING MULTIPLE REPRESENTATIONS

KHAMPHETH BOUNNADY

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2007

หัวข้อวิทยานิพนธ์	การรู้จำพยัญชนะลายมือเขียนภาษาลาวแบบไม่บรรจงโดยใช้ตัวแทนหลายรูปแบบ
นักศึกษา	นายคำเพ็ด บุญนะดี
รหัสนักศึกษา	46060215
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2550
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร. บุญธีร์ เครือตราชู
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	Prof. Dr. Takenobu Matsuura

บทคัดย่อ

วิทยานิพนธ์นี้ได้นำเสนอระบบการรู้จำพยัญชนะลายมือเขียนภาษาลาวแบบไม่บรรจงโดยใช้ตัวแทนหลายรู้แบบ ลายมือเขียนภาษาลาวสามารถแสดงในรูปแบบของลำดับต่อเนื่องของเส้นโค้งที่มีทิศทาง ตามเข็มนาฬิกาและทวนเข็มนาฬิกาเป็นคุณลักษณะเด่นพื้นฐาน ในการที่จะนำคุณลักษณะเด่นเหล่านี้ไปใช้ อันดับแรกจะต้องสามารถแยกเส้นโค้งเล็กๆ ออกจากสัญญาณรบกวนก่อน เส้นโค้งเล็กๆ และสัญญาณรบกวนมีความเหมือนกัน เพราะสัญญาณรบกวนในลายมือเขียนโดยปกติแล้วจะเป็นเส้นโค้งเล็กๆ เช่นเดียวกันเส้นโค้งจริงเมื่อเขียนด้วยความเร็วสูงก็สามารถที่จะลดขนาดลงมาเป็นเส้นโค้งเล็กๆ ได้ และยากที่จะแยกระหว่างเส้นโค้งที่เป็นคุณลักษณะเด่นและเส้นโค้งที่เป็นสัญญาณรบกวนออกจากกัน ด้วยเหตุนี้ในการแก้ปัญหาได้นำเสนอทางเลือกของเส้นโค้งที่อยู่ในรูปแบบของต้นไม้ได้ถูกนำมาใช้เพื่อเป็นคุณลักษณะเด่น โดยที่เส้นโค้งที่เล็กจะให้เป็นส่วนที่ยังไม่แน่นนอน ซึ่งส่วนนี้สามารถที่จะเป็นได้ทั้งเส้นโค้งจริงและเป็นได้ทั้งสัญญาณรบกวนทุกๆ เส้นโค้งที่ยังไม่แน่นอนจะสามารถนำเสนอเป็นได้สองทางเลือก ดังนั้นแทนที่จะเป็นลำดับของเส้นโค้งเดียว จึงสร้างลำดับของเส้นโค้งที่อยู่ในรูปแบบของต้นไม้ และใช้เป็นคุณลักษณะเด่นหลักของตัวอักษร ในการรู้จำจะใช้การเปรียบเทียบกับต้นแบบ ผลการรู้จำที่ได้จากวิธีการที่นำเสนอการเปรียบเทียบแบบยืดหยุ่น การนำเสนอแบบลำดับของเส้นโค้งเดียว และวิธีการแบ่งเส้นโค้งโดยใช้จุดหักหลักๆของตัวอักษรจะถูกนำมาเปรียบเทียบกัน โดยใช้ข้อมูลตัวอักษรลาว 19,481 ตัว จากคนเขียน 18 คน วิธีการที่นำเสนอให้ประสิทธิภาพในการรู้จำสูงเมื่อเทียบกับวิธีการที่เป็นพื้นฐานทั่วๆ ไป เช่นวิธีการเปรียบเทียบแบบยืดหยุ่นและวิธีการอื่นๆ โดยผลการรู้จำลายมือเขียนภาษาลาวเท่ากับ 98.97%.

Thesis Title	Unconstraint Lao Handwritten Character Recognition using Multiple Representations.
Student	Mr. Khampheth Bounnady
Student ID	46060215
Degree	Doctoral of Engineering
Program	Electrical Engineering
Year	2007
Thesis Advisor	Assoc. Prof. Dr. Boontee Kruatrachue
Thesis Co-Advisor	Prof. Dr. Takenobu Matsuura

ABSTRACT

This dissertation proposes unconstraint Lao handwritten character recognition using multiple representations. Unconstraint Lao handwritten character can naturally be viewed as a sequence of curve segments in clockwise followed by counter clockwise directions as a main feature. In order to use this feature, one must be able to distinguish a small curvature from noises. Handwritten character usually contains a small curve segment, which is similar to a noisy curve. Also, some curve segment when written at fast speed can be decreased to small curve segment and it is hard to distinguish between a real curve and a noisy curve. To overcome this problem, an alternative curvature tree is used as a feature, where a small curve segment is considered as an uncertain. The uncertain curve can be both real curve and noisy curve. Each uncertain curve segment generates two possible representations. Therefore, instead of a single curve sequence, a tree of curve segments is formed and used as a main feature of the character, for recognition by prototype matching. The proposed multiple representation is compared with elastic matching, single representation, and dominant point method. In the experiments, 19,481 Lao handwritten characters from 18 writers are used. The proposed method performs very well in comparing to the comparison methods. The recognition rate is 98.97% for unconstraint Lao handwritten character.

Acknowledgments

I would like to take this occasion to express my appreciation to all people who have contributed their help and cooperation to the success of this dissertation.

Firstly, I would like to express my sincere thanks to my supervisor, Assoc. Prof. Dr. Boontee Kruatrachue, who gives me helpful suggestions, useful advises, teaching, and cooperation for researching during my study at King Mongkut's Institute of Technology Ladkrabang (KMITL).

Also, I would like to appreciate my co-advisor, Prof. Dr. Takenobu MATSUURA, from School of Information Technology and Electronics, Tokai University, Japan for his useful advises, and research cooperation.

I would like to gratefully thank AUN/SEED-Net and KMITL for scholarship and technical supports.

I thanks to all my friends, relatives, co-workers, who have provided me their handwritten characters used in my experiments.

Finally, this dissertation is dedicated to my grandfather, grandmother, my parents, my sisters and everyone in my family, who always give supports and encourages for all in my life.

Contents

Page

Abstract (Thai) I

Abstract (English) II

Acknowledgments III

Contents IV

List of Tables VI

List of Figures VII

Chapter 1 Introduction 1

1.1 Requirements and problems 1

1.2 Objective and scope of this dissertation 2

1.3 Device used in this dissertation 3

1.4 Overview of this dissertation 4

Chapter 2 Character Recognition 5

2.1 Off-line character recognition 6

2.2 On-line handwritten recognition 8

2.3 Character sets 16

2.4 Writing style variations 16

2.5 Lao character recognition 17

Chapter 3 Feature Extraction 21

3.1 Input data 21

3.2 Lao handwritten character feature extraction 22

3.3 Removing of hook at head and tail of character 29

3.4 Segmentation 33

3.4.1 Segmentation using dominant point 33

3.4.2 Segmentation using direction change of handwritten 35

3.5 Alternative curvature tree representation of character 40

Contents (Continued)

Page

Chapter 4 Handwritten Character Classification 53

4.1 Recognition process 53

4.2 Data normalization 54

4.3 Dissimilarity of characters 57

4.4 Prototype classification 63

Chapter 5 Selected Related Methods 64

5.1 Elastic matching 64

5.1.1 Data normalization 64

5.1.2 Dissimilarity by elastic matching 65

5.2 Single representation 68

5.3 Dominant point 71

5.4 Prototype selection 74

Chapter 6 Recognition Experiments and Results 76

6.1 Recognition of Lao handwritten character 76

6.2 Recognition results of Lao handwritten character 77

6.3 Analysis of recognition results 82

Chapter 7 Conclusions 86

7.1 Alternative curvature tree feature representation 86

7.2 Performance of alternative curvature tree 86

Reference 89

Appendix 93

Biography 124

List of Tables

Tables Page

3.1 Segmentation points 37

6.1. Experiment handwritten training and testing data sets randomly partition from 19481 handwritten data set 77

6.2. Recognition result 78

6.3 Comparison of handwritten error 79

6.4 Confusing matrix of Lao handwritten character recognition of the proposed method (5 sets) 80

6.5 Confusing matrix of Lao handwritten character recognition of the elastic matching (5 sets) 80

6.6 Confusing matrix of Lao handwritten character recognition of the single representation (5 sets) 81

6.7 Confusing matrix of Lao handwritten character recognition of the dominant point (5 sets) 81

6.8 Recognition of character has short tail and long tail from the proposed method 83

6.9 Recognition of character has short tail and long tail from elastic matching method 83

List of Figures

Fig Page

1.1 Lao character.......... 2

1.2 Curvature segment of Lao handwritten character.......... 3

2.1. Classification of character recognitions.......... 5

2.2 Off-line handwritten recognition, (A) input document, (B) its recognition result [12].......... 6

2.3 Handwritten address interpretation [13].......... 7

2.4 Word recognition for bank cheque [14].......... 7

2.5 Example of signature verification, (A) and (B) are cursive type signature, (C) and (D) are graphical type signature and on the right side (A) and (C) are original signature, (B) and (D) are forged signature [15].......... 8

2.6 (A) Direction value. (B) Example of representation of character [5].......... 9

2.7 Example deformation of T. (A) Relaxed version of T as result of applying type deformations. (B) Relaxed version of T as result of applying direction reformations [5].......... 9

2.8 Shown direction feature [6].......... 10

2.9 Shown eight direction change feature [6].......... 10

2.10 Clockwise counter clockwise direction change feature [6].......... 11

2.11 Character is divided into sub-stroke. (A) The angle of each data point. (B) Original character, (C) Divided character and (D) Merge character [7].......... 11

2.12 Five states HMM that use as model of recognition [7].......... 12

2.13 Example of primitive stroke [17].......... 12

2.14 Example of matching path [17].......... 13

2.15 Quantization of slope values [9].......... 13

2.16 Dominant points of character. (A) Curvature threshold=0. (B) Curvature threshold=1. (C) Curvature threshold=2 [9].......... 13

2.17 Example of reconstructed storks from a scanned image [11].......... 15

2.18 Example Segment separation for On-line [11].......... 15

2.19 Example application of on-line handwriting recognition. (A) PDA, (B) computer.......... 15

List of Figures (Continued)

Fig Page

2.20 Example of handwritten style Lao character. 17

2.21 Example of character with hook, (A) Lao character, (B) English character [22]. 17

2.22 Example of constrained writing style [22] 17

2.23 Binary image of character "ກ" [21]. 18

2.24 Graph of fuzzy membership function in each term [21] 20

3.1 Handwritten input as a sequence of X, Y coordinates of pen tracing 21

3.2 Handwritten speed effect on sampling point of pen tracing. 22

3.3 Hooks at head and tail of character 22

3.4 Lao handwritten character perceive as a sequence curvature 23

3.5 The directional code calculation form GCC 24

3.6 Examples input data translation to *DCs* 25

3.7 Length calculation between two sampling points 26

3.8 Angles calculation 27

3.9 Example of Lao handwritten character and its X,Y coordinates,
directional codes , lengths and angles 28

3.10 Angle threshold of 2.5 29

3.11 Show points index from starting and ending of character 30

3.12 (A) Boundary length for check hook. (B) Main dominance of character like as hook 31

3.13 Example of hooks locating at head and tail of character 32

3.14 Hook removing example 33

3.15 Segmentation points of character by using retrace point. 34

3.16 The same character with difference number and position of segmentation points 34

3.17 The threshold value is too big to find the segmentation points 34

3.18 Segmentation points using retracing point [31] 35

3.19 Segmentation using curvature rotation in clockwise and counter clockwise directions 35

3.20 The relation of angle sign and writing direction in clockwise and counter clockwise
direction. 36

3.21 (A) Quantized noise cancellation, (B) The threshold 0.6 curvature 38

List of Figures (Continued)

Fig Page

3.22 Examples of segmentation points of handwritten characters 38

3.23 Locate segmentation points by monitoring angle signs in the sequence 39

3.24 Sequence of curvature segments of character 40

3.25 Uncertain segment with small curvature 41

3.26 Uncertain segment and its effect in recognition 42

3.27 Examples of noise segment, uncertain segment, and real segment calculation 43

3.28 After removing of noises segments, only uncertain and real curve segment remain. 44

3.29 State 1 of tree generations at segment number 1 (real segment) 45

3.30 State 2 at segment number 2 (uncertain segment and assumes it is real segment) 45

3.31 State 3 at segment number 2 (uncertain segment with noise segment assumption). 46

3.32 State 4 at segment number 3 (real segment) 46

3.33 State 5 of tree generations at segment number 3 is real segment and merges with segment S1+S2+S3 that have the same rotation. 46

3.34 State 6 at segment number 4 (uncertain segment with real segment assumption) 47

3.35 State7 at segment number 4 (uncertain segment with noise segment assumption) 47

3.36 State 8 of tree generations at segment number 5 is real segment. 47

3.37 State 9 of tree generations at segment number 5 is real segment merges with segment S3+S4, and also merge with segment S1+S2+S3+S4, they have the same rotation 48

3.38 Shown all representation of Lao handwritten character "ຕ" 48

3.39 Shape invariant. 49

3.40 Rotation invariant 49

3.41 Show difference character but have the same single clockwise curve segment 50

3.42 Segments in X and Y coordinate to assist easier separation of difference curve with the same rotation 50

3.43 Example calculation of segments in X and segments in Y 51

3.44 Example feature-vector of Lao handwritten character "ຕ" 52

4.1 Recognition system 53

List of Figures (Continued)

Fig — Page

4.2 Feature of Lao handwritten character "ຕ" 54

4.3 The two "head" angle are about the same size after normalization even with multiple loops 55

4.4 Data of character after normalization 56

4.5 Example of similarity calculation between two characters 57

4.6 Example of segment overlap 58

4.7 Calculation of dissimilarity of sequence of segment in X, with identical number segments and direction. 60

4.8 Calculation of dissimilarity of sequence of segment in X, with different number of segments but identical start direction 61

4.9 Calculation of dissimilarity sequence of segment in X, with different direction 62

4.10 Calculation dissimilarity sequence of segment in X, with numbers of segment and direction are difference 62

4.11 Prototype matching of Lao handwritten character recognition 63

5.1 Illustration of warping path of sequence A and B 66

5.2 Point to point matching of sequence A and B 67

5.3 Recognition system of elastic matching 67

5.4 Sequence of curvature segments of Lao handwritten character "ຕ". 69

5.5 Example of noise segments is removed 70

5.6 The effect when threshold is too big, small curves are miss interpret as noises 70

5.7 Segmentation points of character using retrace point 71

5.8 Locate segmentation points by monitoring angle value in the sequence 72

5.9 Example segments of character 73

5.10 (A) The threshold value is too small. (B) The threshold value is too big 73

5.11 Prototypes selection 75

6.1 (A) Lao handwritten characters and their printing character. (B) Unconstraint Lao handwritten character "ຊ" 76

List of Figures (Continued)

Fig		Page
6.2	Average recognition rate of Lao handwritten character from 5 data sets	79
6.3	Misrecognition of handwritten characters with the same shape but difference in tail length	82
6.4	Misrecognition of handwritten character with the same shape and rotation	82
6.5	Elastic matching misrecognition examples of Lao characters with same shape but difference direction	84
6.6	Misrecognition of Lao character in elastic matching due to curvature direction	84
6.7	Misrecognitions by single representation of Lao handwritten character	85
6.8	Misrecognition of Lao character because of missing the small real curve at tail	85
7.1	Elastic matching method misrecognized due to direction of curve at head	87
7.2	Elastic matching method missing dominant curve in the matching	87
7.3	Single representation misrecognition due to treating of small curve as a noise	87
7.4	Variant in dominant point of the same character	88

Chapter 1

Introduction

Handwritten character recognition has been an on-going research in many languages for many years. Main reasons for a long research due to each language need a different feature for recognition. Handwritten character recognition has been gaining more interest and has attracted more research since hand-held computers, PDAs, palms, digital notebooks, and advance mobile phones have been increasing popularity. These devices have small size, so keyboard is inconvenience to be used as their interface. They need another method for input.

1.1 Requirements and problems

Traditionally man-machine communications have based on keyboard and pointing devices. These methods can be very inconvenient when the machine has small size. Since, a keyboard is very difficult to integrate in a small device. It is also hard to use when the number of characters is very large such as in Chinese and Japanese character. Because of these problems, new methods for input have been developed, for example systems that recognize speech and handwriting, both of them are very natural ways to communicate and people can easily learn and use them.

There have been a lot of researches on handwritten. They have proposed their research using different techniques with many different features such as Structure method [1],[27],[28],[29],[30], Neural network [2],[35],[36],[37],[38], Hidden Markov Model [3],[23],[24],[25],[26], prototype matching [4],[22],[32],[33],[34] etc., Handwritten research has been also done in many languages such as English [5], Japanese [6], Thai [7], Chinese [8], Tamil [9], Arabic [10], Korean [11] , etc.. Different languages use different features and/or methods for recognition. There are no common methods or feature that can perform well for all languages. From the past to present there are many methods successfully recognizing of handwritten character with high accuracy and short recognition time. Very few have proposed on Lao handwritten, especially for unconstraint handwritten with high recognition rate.

1.2 Objective and scope of this dissertation

This dissertation proposes a technique to improve recognition accuracy and recognition time of unconstraint Lao handwritten character. Some examples of unconstraint Lao handwritten are shown in Fig. 1.1(A). These handwritten characters are written out of proportion and varied in amount of rotation (slant) and writing style. Most zone features are less effective due to being out of proportion since some characters will have big, small or omitted head and cause shifting in zone as shown in Fig. 1.1(B). Unconstraint Lao handwritten character can be represented with a less variation as a continuous drawing of curves. As shown in Fig. 1.1(A), it is natural to represent the characters as a sequence of curve segments with clockwise and counter clockwise directions as the main feature. This agrees with the way Lao learn to write a character as a sequence of clockwise and counter clockwise curvature. Obviously this representation is not suitable for Chinese characters, since they consist of a sequence of stroke.

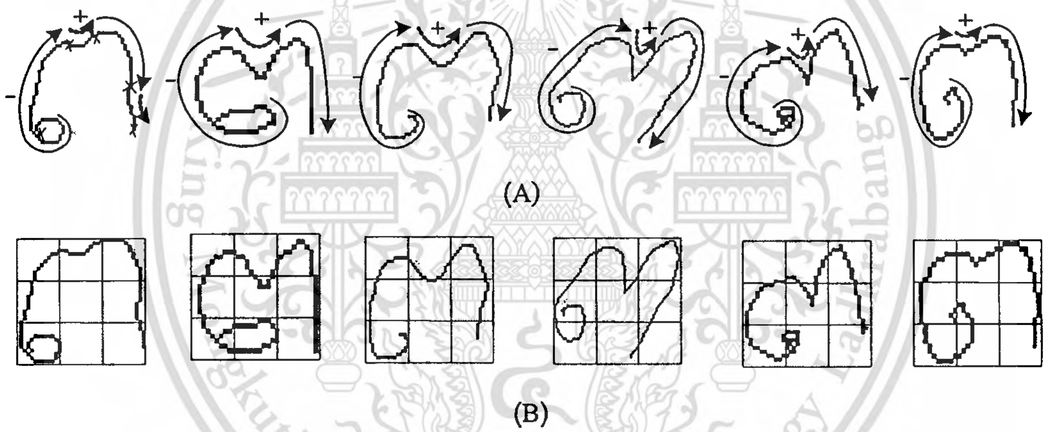


Fig 1.1 Lao character.

Even though this feature quite coincides with the way writers learn to write as in Fig. 1.2(A), it may consist of noisy curves. Usually, a noise curve segment can be distinguished from a real curve by the high amount of curvature of the real signal. But when the written speed is fast, the real curvature is reducing closed to the noisy curve as shown in Fig 1.2(C). These noisy curves are hard to distinguish with normal noise reduction technique such as Fourier expansion or Fourier descriptor due to their similarity in frequency and amplitude. In order to successfully use the curvature sequence in the recognition, one may have to identify the real curvature from the noisy one. As shown in Fig 1.2, the character ຕ consist of three curvatures in sequence. If all small curvature segments are treated as noise and omitted, the character will match another character ດ that has one clockwise curvature. On the other hand, if all curvatures are treated as real curve the character will have 4 curvatures and it can not match to the correct character with three curvature segments.

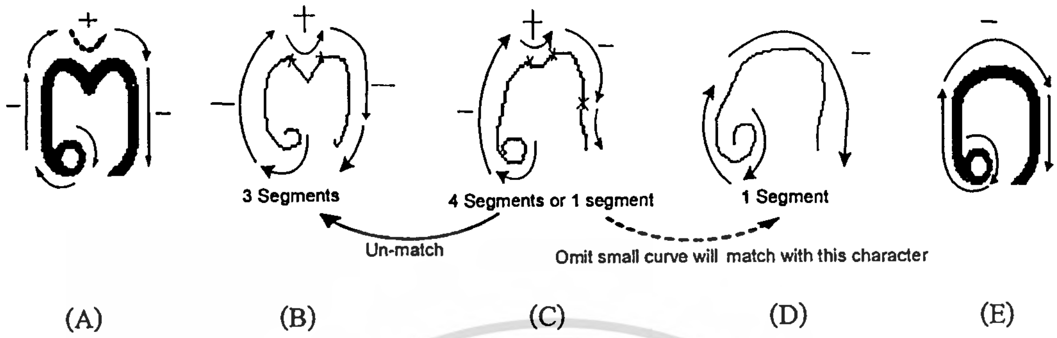


Fig 1.2 Curvature segment of Lao handwritten character.

In this dissertation, curvature segment is classified into three groups, noise, real and uncertain segments. The noise group consists of the noise curves that can be identified and removed. The uncertain group consists of noise and real signals that can not be distinguished. The real group has high curvature enough to be certain as a real signal. Once all curvature segments in a handwritten character are identified, the noisy segments group is removed. Then, alternative curvature tree is created from real group and uncertain group. The branches in the tree are all combination of each uncertain segment. So a feature of a Lao handwritten character is not just a single sequence but a tree representation of curve segments.

This dissertation proposes unconstraint Lao handwritten character recognition using multiple representations with the following assumptions:

1. Input character is 27 handwritten alphabet Lao characters.
2. Handwritten characters are all written in a single stoke only.
3. Handwritten characters are isolated unconstraint to any particular style.

1.3 Device used in this dissertation

1. Tablet with sampling rate 100 sampling per second.
2. Computer PC Pentium 4 with 1.4 GHz and RAM 512.
3. Program Microsoft Visual C ++ 6.0.

1.4 Overview of this dissertation

This dissertation discusses the essential background knowledge of the research field of handwritten character recognition and makes a literature survey of the related and recent recognition methods. The dissertation explains the theoretical recognition process from feature extraction, prototype selection, and concludes with recognition experiments with comparison to selected methods. The Chapter2 mainly consists of the literature survey of researches from articles in the various journals and conference proceedings about pattern recognition and artificial intelligence. The Chapter3 describes our proposed method applied to isolated Lao handwritten recognition. Start with the description of input and its encoding, the hook removing at the head and tail of a character. Locate the segmentation points, which divide character into segment in clockwise and counter clockwise direction. After that, the alternative curvature tree of character is generated. The Chapter 4 discusses about recognition system and prototype selection. All the experiments and their recognition results are discussed in the Chapter 5. The major conclusion drawn from the results are discussed in the Chapter 6.

Chapter 2

Character Recognition

Character recognition system can be divided into two categories. It divides from the type of input data. The first category is Off-line character recognition and the second one is On-line character recognition. The Off-line character recognition system has two subtypes. The first type is printed character recognition and the second is handwritten character recognition. The inputs of these methods are characters that are already printed or written on a paper respectively. They are then scanned and converted into image. Features of a character are extracted from the image. Off-line character recognition does not recognize in real time. In on-line character recognition system, its input is a sequence of data points sampling from electronic pen device or tablet while drawing. This has much more information than the Off-line method including writing speed, direction, pressure, etc.. In on-line method the recognition can also perform in real time. Fig 2.1 Shows the flowcharts of character recognition systems.

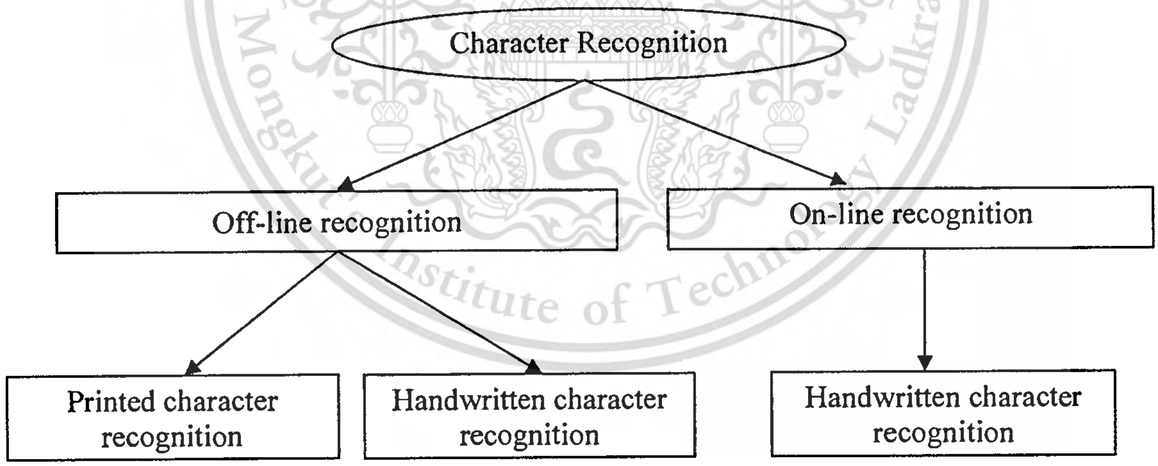
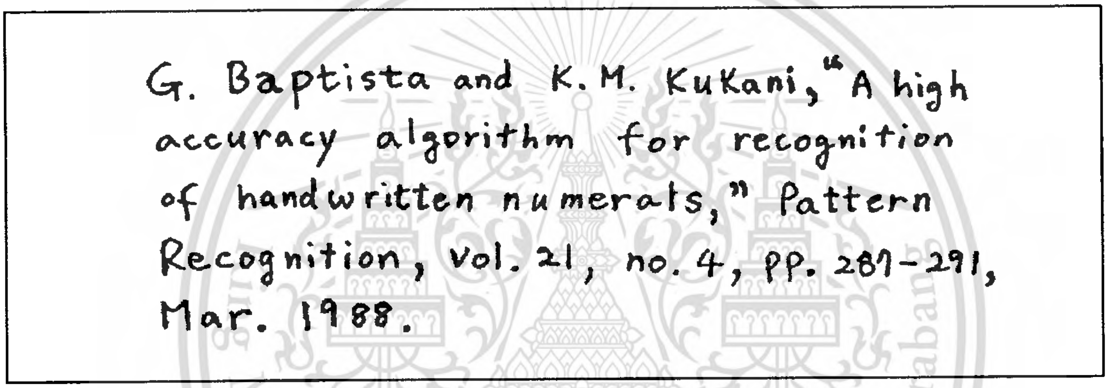


Fig 2.1 Classification of character recognitions

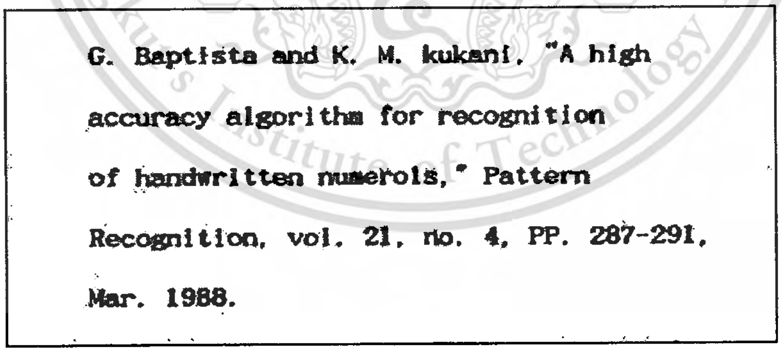
2.1 Off-line character recognition

Off-line character recognition recognizes character that has originally been written on paper. The characters are digitized into two dimensional images. Features for recognition are enhanced and extracted from the bitmap images by means of digital image processing. Character is not recognized at the same time as it is produced but after the writing task is completed. Example shown in the Fig 2.2 is Off-line character recognition which has been proposed by research [12]. This research is also proposed Off-line recognition of constrained Handwritten Hangul (Korean character) and alphanumeric characters using discrete hidden Markov models (HMMs).



G. Baptista and K. M. Kukani, "A high accuracy algorithm for recognition of handwritten numerals," Pattern Recognition, Vol. 21, no. 4, pp. 287-291, Mar. 1988.

(A)



G. Baptista and K. M. kukani, "A high accuracy algorithm for recognition of handwritten numerols," Pattern Recognition, vol. 21, no. 4, PP. 287-291, Mar. 1988.

(B)

Fig. 2.2 Off-line handwritten recognition, (A) input document, (B) its recognition result [12].

Off-line methods are suitable for automatic conversion of paper documents to electronic documents. Typical applications for optical character recognition are systems which need to automatically handle a huge amount of information in paper form. Such recognition systems have been developed for numerous application areas but are mainly used for machine printed character. The examples for handwritten characters are: reading postal address [13], handling of financial documents such as cheque [14], writer recognition, and signature verification [15] as shown in Fig 2.3 to Fig 2.5. Further details about Off-line recognition can be founded in [16].

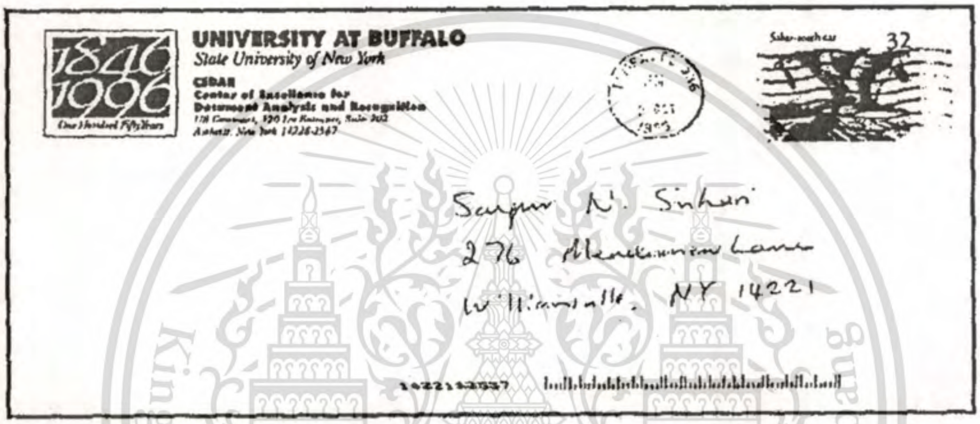


Fig 2.3 Handwritten address interpretation [13].

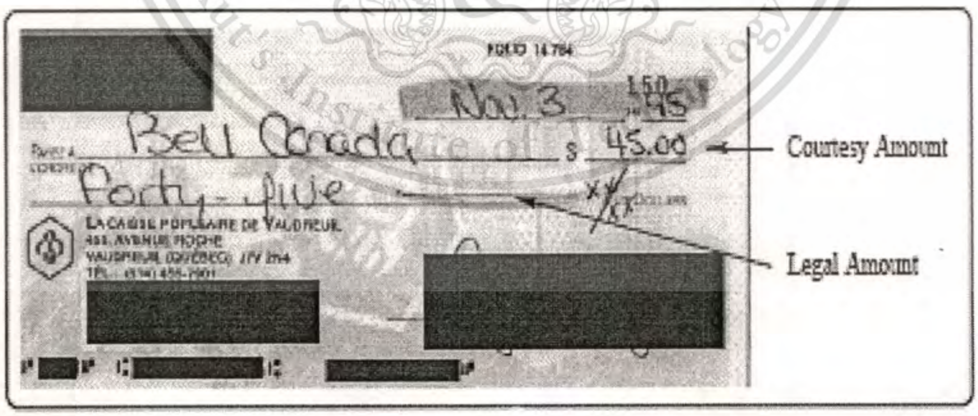


Fig. 2.4 Word recognition for bank cheque [14]

Fig. 2.5 Example of signature verification, (A) and (B) are cursive type signature, (C) and (D) are graphical type signature and on the right side (A) and (C) are original signature, (B) and (D) are forged signature [15].

2.2 On-line handwritten recognition

In On-line character recognition, character is input with special equipment while the character is written. If this input is processing fast enough, the recognition can be done in real time. The input is usually a tablet that can capture location information and motion of a pen or some other pointing device. The information data collected from input device are sent to the computer as input of recognition task. On-line recognition is considerably different from Off-line recognition because of the dynamic information on writing such as: stroke order, writing speed, angle of the pen and its pressure on writing surface. Additional information that is absent in Off-line handwritten recognition, can be derived from On-line data such as drawing order and direction of the strokes. The dynamic information can improve the recognition accuracy.

There are many papers in On-line handwritten recognition with high accuracy of recognition rate. They have been developed in many languages such as: English [5], Japanese [6], Thai [7], Chinese [17], Tamil [9], Arabic [10], Korea [11], formulas [19], etc.. More detail reviews about On-line recognitions can be founded in research [16].

The On-line handwritten recognition system performance depends on the selection of handwritten feature and the selected of recognizer to classify that feature. Feature of character can be extracted by many techniques such as: chain code, direction of stroke, stroke order, etc.. The system with appropriate feature and recognizer can give a high recognition rate as some systems described below.

Chan and Yeung [5] have proposed On-line handwritten alphanumeric characters recognition through flexible structural matching. A handwritten character consists of curve or line. Each curve or line consists of five types of primitives. They are line, up (curve with counter-clockwise), down (curve with clockwise), loop (curve joining itself at some point), and dot. Character is represented as structure of line and direction or curve and direction. Direction is calculated from the start of a curve or a line to the end of a curve or a line as shown in Fig 2.6. In recognition process, structures of test characters are compared with the models by using flexible structure matching. If there is no exact match, the primitive type or direction may be deforms. As an example, {line, 6} maybe line {line, 5} or {line, 7}. The direction deformation is shown in the Fig. 2.7. The recognition rate of this proposed method is about 98.6% for digits, 98.49% for uppercase letters and 97.44% for lowercase letters.

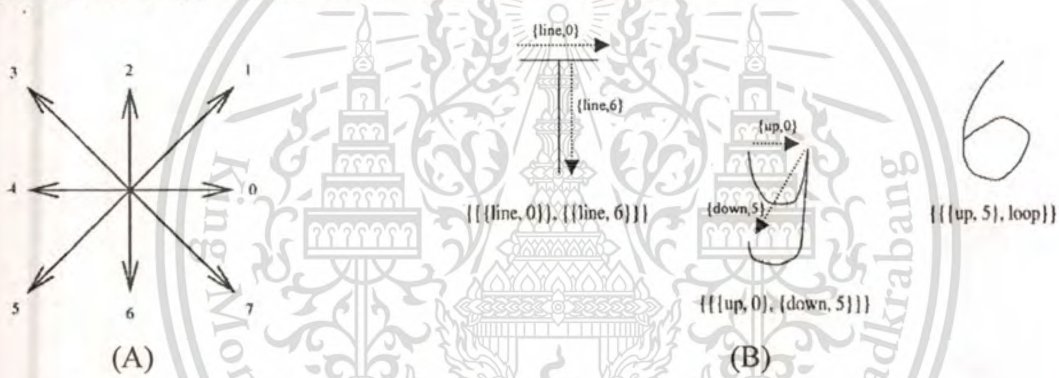


Fig 2.6 (A) Direction value. (B) Example of representation of character [5].

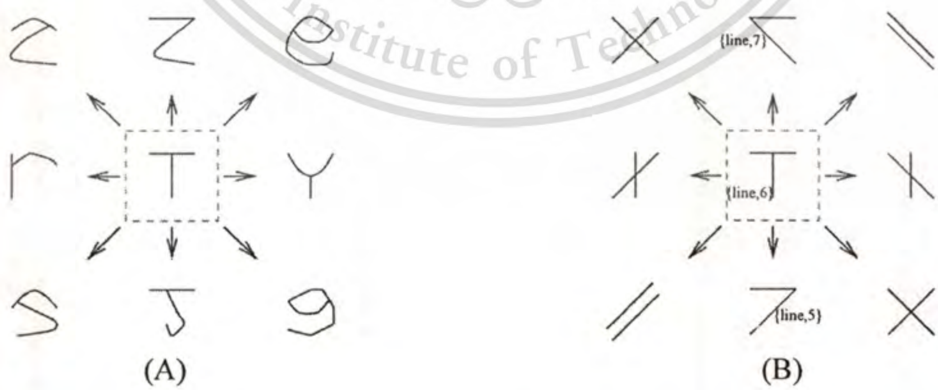


Fig 2.7 Example deformation of T. (A) Relaxed version of T as result of applying type deformations. (B) Relaxed version of T as result of applying direction deformations [5].

Okamoto and Yamamoto [6] have proposed On-line handwritten character recognition using directional feature and clockwise/counterclockwise direction change feature. The feature extraction is as follows: First on-line character data are transformed to bitmap data. Next, bitmap data and on-line data are normalized by Line Density Equalization. Directional features are extracted from bitmap data while written area, direction change and circle features are extracted from on-line data. Directional feature consists of vertical, right up slant, horizontal and left up slant as shown in the Fig. 2.8. Written area features express location and size of characters. In Japanese characters, there are sets of similar characters where their sizes and/or positions are difference while their shapes are the same. Direction change features have two types. First, after direction changes are extracted each direction is mapping to eight directions meshes as shown in the Fig 2.9. Second, after direction changes are extracted each direction is mapping to one of clockwise or counterclockwise meshes base on whether the direction change is clockwise or counter clockwise as shown in the Fig. 2.10. This feature replaces eight direction change features. The recognition is performed by pattern matching. The recognition rate of this method is about 91.33% for Kanji and 87.15% for All (Kanji and non Kanji).

Fig 2.8 Shown direction feature [6]

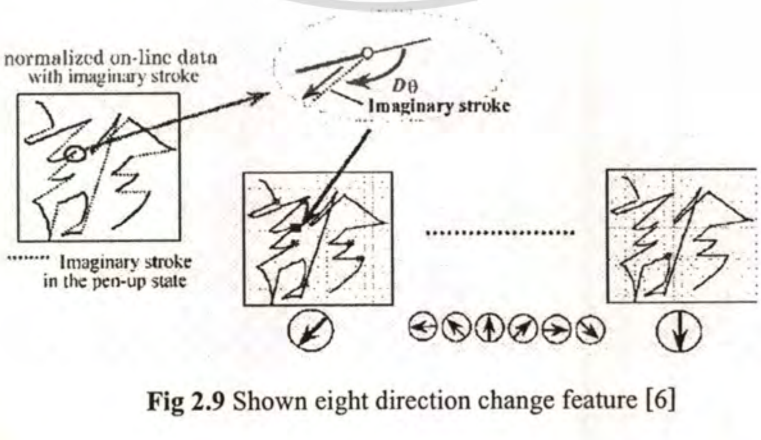


Fig 2.9 Shown eight direction change feature [6]

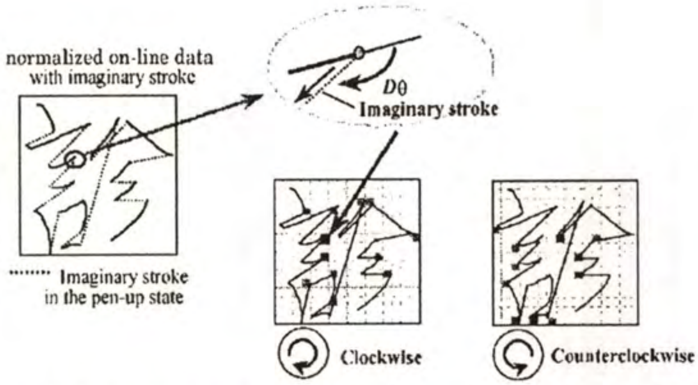


Fig 2.10 Clockwise counter clockwise direction change feature [6]

Methasate and Sea-Tang [7] have proposed On-line Thai handwriting character recognition system using the segmented stroke information and Hidden Markov Model. Firstly stroke sequences of character are smoothed and removed hooking stroke. After that character is divided into sub-stroke. It is divided to sub-stroke at point that has size of angle is bigger than its threshold as shown in Fig 2.11 (A). In case some sub-strokes are too short the merging process is applied by merging the short stroke to either of the neighbor sub-stroke, if they have same curvature direction as Fig 2.11. These sub-strokes are used to extract the information to be a feature vector sequence for HMM recognition engine as shown in the Fig 2.12. The features used to describe the sub-strokes consist of starting stroke coordinate (P_s), last stroke coordinate (P_l), center of gravity coordinate ($P_{C.G.}$), angle at the start of stroke (θ_s), angle at the end of stroke (θ_e), angle from start to end of stroke (θ_v), straight-ness (S), relative length (L), stroke shape and stroke location. From their experiments the average recognition rate of this method is about 92.17 %.

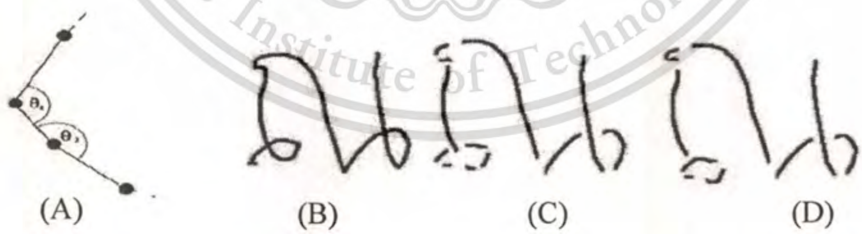


Fig 2.11 Character is divided into sub-strokes. (A) The angle of each data point. (B) Original character, (C) Divided character and (D) Merge character [7].

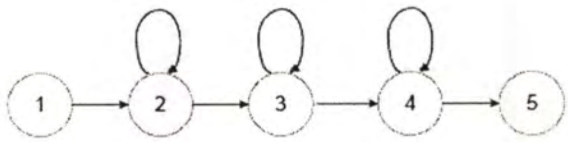


Fig 2.12 Five states HMM used as the model of recognition [7].

Lin and Jeng [17] have proposed an On-line recognition of handwritten Chinese character based on stroke-sequence feature extraction and using string matching method called "Dynamic Neighboring Matching" (DNM). Segment of an input character from "pen down" until "pen up" is called a stroke. In feature extraction strokes of character are defined by 27 primitive strokes with code as shown in the Fig 2.13. Feature of a character consists of three features. They are stroke order, primitive stroke, and number of strokes. In recognition process unknown pattern and reference pattern have the same number of strokes are compared together by using DNM. In Fig 2.14 shows the matching path of 6 strokes by DNM. The recognition rate of this method is about 93.5%.

Stroke Type	Primitive stroke codes	Stroke Type	Primitive stroke codes
	1		18
	2		19
	3		20
	4		21
	5		22
	7		23
	8		24
	9		25
	10		28
	11		29
	13		30
	15		31
	16		32
	17		

Some Examples

持 (Hold) : 1, 11, 5,1,2,1,1,11,4.

拖 (Pull) : 1,11,5,3,1,10,2,16.
拆 (Break) : 1,11,5,3,3,1,2,4.
投 (Throw): 1,11,5,3,17,10,4.

Fig 2.13 Example of primitive stroke [17].

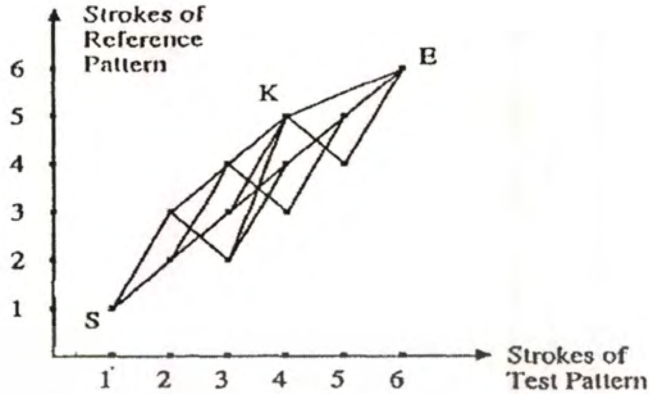


Fig 2.14 Example of matching path [17].

Joshi, Sita, Ramakrishnan and Madhvanath [9] have proposed a comparison of elastic matching schemes for writer dependent on-line handwriting recognition of isolated Tamil characters. All characters are preprocessed, smoothed and normalized to 60 sampling points. The three different features are considered namely, preprocessed x y coordinates, quantized slope values shown in the Fig 2.15, and dominant point coordinates of character shown in Fig. 2.16.

Fig 2.15 Quantization of slope values [9]

Fig 2.16 Dominant points of character. (A) Curvature threshold=0. (B) Curvature threshold=1. (C) Curvature threshold=2 [9]

There are seven schemes based on these three features using dynamic time warping (DTW) for recognition. They are four schemes employing single stage procedures for recognition and four of them are basic schemes. First scheme uses preprocessed x y coordinates as features. It used Euclidean distance as cost measure and used DTW as classifier. Second scheme uses quantized slope values as features. A fixed cost matrix is used to find out cost measure of dissimilarity between two quantized slopes and used DTW as classifier. Third scheme uses dominant point x y coordinates as features with unity curvature threshold. The rest of the procedure is the same as first scheme. Fourth scheme uses preprocessed x y coordinates as features and Euclidean distance as cost measure. The difference between this scheme and first scheme is warping path forced to follow diagonal path of the warping matrix. The other three schemes are combinations of the above four basic schemes. Each of the following three hybrid schemes accomplishes the recognition task in two stages. The first stage is a pre-classification stage with low computational complexity and selects top 5 choices as its output. Second stage selects the output from these 5 choices and provides post-classification. Fifth scheme uses quantized slope based classifier described in scheme 2 at the pre-classification stage and preprocessed x y coordinate based classifier described in first scheme at the post-classification stage. Sixth scheme uses dominant point co-ordinates based method described in third scheme at both of its stages. In the pre-classification stage, threshold is set to 2 and in the post-classification stage, it is reduced to 1. Seventh scheme uses rigid matching scheme based on preprocessed x y coordinates (Fourth scheme) at its pre-classification stage. Post-classification uses elastic matching scheme base on preprocessed x y coordinates (First scheme). The recognition rate is 94.8% from scheme 6 and 95.5% from scheme 7, which is better than the rest of the schemes.

Jeo and Lee [11] have proposed On-line and Off-line recognition Korean in one system. In Off-line system features of character are extracted using the following steps. First the thinning process is performed for the character image. Then character is separated into segment at the point where pixels are connected together. After the character has been separated if there are some segments, those segments are connected together to make complete stroke. Segments are connected under three assumptions. First the strokes in a character are very smooth everywhere except at inflection points. Second a 3 fork point is composed of a stroke going through it and another stroke terminating at it. Third a 4 fork point is composed of two stroke crossing at it. After segment connection process stroke is arranged from left top to right bottom as shown in the Fig 2.17. Then Off-line character is represented as a sequence of strokes. The recognition is done using match stroke with On-line model stored in the data base. For On-line recognizer, a set of stokes is extracted from input character after noise filtering. It is further divided in to segments, and a segment makes the rectangle wraps it. The strokes are divided in to segments according to the following three cases. Case 1 is alternative stroke. Case 2 is the accumulating angle to become 360 degree. And case 3 is direction of stroke change from clockwise to counter clockwise, and vice versa. A set of stroke is decomposed in to segments which maybe a line or curve as shown in the Fig 2.18. Each segment has feature consisting of the size of segment, the start and end point of segment, and the direction of segment. In recognition dissimilarity is calculated between feature vector of input and the reference vector in database. The character with minimum dissimilarity between the input and reference is selected as recognized character.

(a) Scaned image (b) Segments (c) Strokes (d) Arranged strokes

Fig 2.17 Example of reconstructed strokes from a scanned image [11]

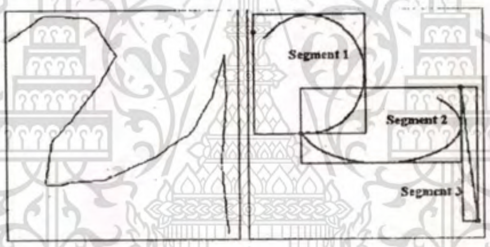


Fig 2.18 Example segment separation for On-line [11]

The main applications of On-line character recognition is limited for interaction as they are typically small devices such as PDA, palm, digital notebooks, and advance mobile phone, etc.. In more advanced applications handwriting recognition system can replace the mouse and keyboard of computer. Example of applications of on-line handwriting recognition is shown in the Fig 2.19.

(A) (B)

Fig 2.19 Example of applications of on-line handwriting recognition. (A) PDA, (B) computer.

2.3 Character sets

Character recognition system may yield higher accuracy when the number of characters is small and characters are different. On the other hand, recognition systems are difficult to recognizing when number of characters is large such as Chinese, Japanese character. It is also difficult to recognize character with complicated structure with many strokes. The Chinese characters have an average of 8-10 strokes. Strokes of Chinese characters mainly consist of straight lines. In contrast, Lao character has small number of characters. Almost all of them have just one stroke. Only few characters have two or more strokes. Lao character consists of sequence of curve with small and big curve. The difficulty in recognition of Lao characters is the resemblance of small curve and un-intentional curve.

2.4 Writing style variations

Normally, handwritten characters are various in shapes and writing style when compared to machine printed characters. Most of handwritten characters are different between writers. Also character, from the same writer but write at different time, may not be the same. Handwritten characters may be distorted from ideal character models and there are many factors influencing this distortion. Examples of these factors are personal background factors such as writing style, left hand or right hand, the writing from head to tail of or from tail to head. Situational factors such as writing speed for example, important letter address are very carefully written while an ordinary shopping list is unreadable even to the writer himself. Other factors are material factors such as friction of the surface, side of box writing, line of writing, etc.. These factors can affect both the generation and recognition of the characters.

Handwritten characters can vary in their static and dynamic properties [18]. Static properties are ideal model of the character and their geometrical properties such as relative position, size, corners and retraces, aspect ratio, and slant. Dynamic properties are collected from the generation of characters such as writing speed, writing pressure. Characters may look similar but have different dynamic properties such as different number of strokes, stroke order, and direction. Because of these variations, the number of writing styles can be very large as shown in the Fig 2.20.

Character always has some very short retraces at the beginning and ending of character. This kind of stroke is called "hook" as shown in the Fig 2.21. Hook can decrease recognition rate if not handle properly.

In order to achieve higher recognition rate, some handwritten recognition system put some constraints in handwritten character. Usually users are expected to write specific character style and characters are written in boxes to simplify separation. Some system provides guide lines to help the users write more consistently such as shown in the Fig 2.22.

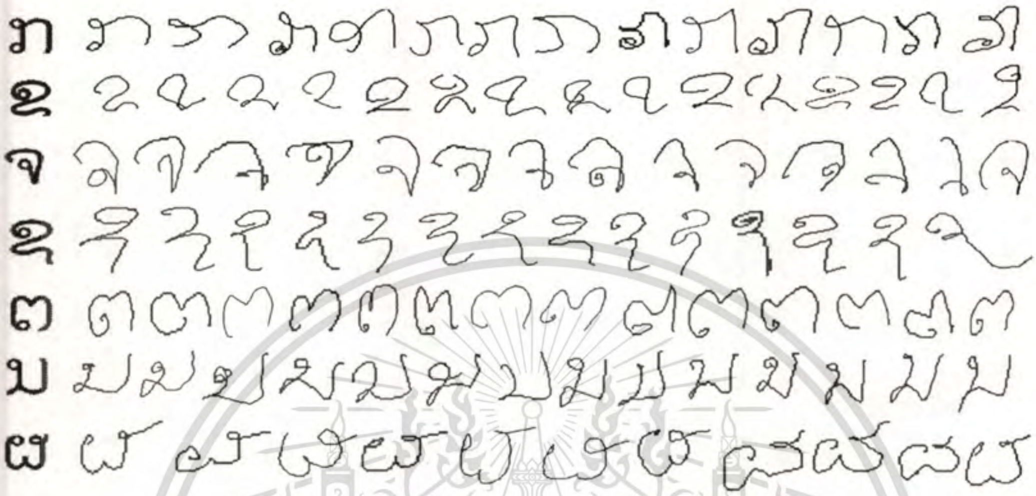


Fig 2.20 Example of handwritten style Lao character.

(A) (B)

Fig 2.21 Example of character with hook, (A) Lao character, (B) English character [22].

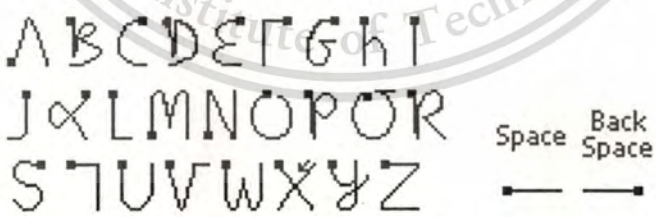


Fig 2.22 Example of constrained writing style [22].

2.5 Lao character recognition

Very few methods have been proposed for Lao handwritten character recognition. Intawongkam and Chamnongthai [20] have proposed Off-line recognition of handwritten Lao alphabet by using back-propagation neural networks. The input is the image of Lao alphabet. The feature is the set of line, curve and circle encode by chain code, which are used as input of back-propagation neural networks. The recognition rate is about 80%.

Moungsinh [21] has proposed printed Lao characters recognition by using fuzzy and back-propagation neural techniques. This research is a work on Lao printed character with 8 types of front. They are Alice0, Alice1, Alice2, Alice3, Alice4, Alice5, ChanthabouliLao, and Sysetthalao. Each font is used 4 with different sizes, 14, 16, 18 and 20 points. Character is scanned in to a bitmap image, and then translate to binary image as show in Fig.2.23.

	0																														m
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
	0	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	0	0	0
	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
	0	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
p	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig 2.23 Binary image of character "ກ" [21].

From the character with n rows and m columns, shown in Fig. 2.23 a set of feature is calculated below:

1. TOT is the summation of all points (pixel) with value 1 (Total of point with value 1).

$$TOT = \sum_{i=1}^{n} \sum_{j=1}^{m} X_{i,j} \tag{2.1}$$

2. UP is the summation from 1 to p, where p is center of character (Up part of character with value 1)

$$UP = \sum_{i=1}^{p} \sum_{j=1}^{m} X_{i,j} \tag{2.2}$$

3. DOWN is the summation from P to n (Down part of character with value 1)

$$DOWN = \sum_{i=p+1}^{n} \sum_{j=1}^{m} X_{i,j} \tag{2.3}$$

4. DO_UP is ratio of DOWN and UP

$$DO\_UP = \frac{DOWN}{UP} \tag{2.4}$$

5. LEFT

$$LEFT = \sum_{i=1}^{n} \sum_{j=1}^{p} X_{i,j} \tag{2.5}$$

6. RIGHT

$$RIGHT = \sum_{i=1}^{n} \sum_{j=p+1}^{m} X_{i,j} \tag{2.6}$$

7. RG_LF is ratio of RIGHT and LEFT

$$RG\_LF = \frac{RIGHT}{LEFT} \tag{2.7}$$

8. MaxH is maximum point of projection in horizontal direction

$$\begin{aligned} H^{t}_{PROJ} &= [h_1, \quad h_2, \quad h_3, \quad \ldots\ldots, \quad h_n]^T \\ h_i &= \sum_{j=1}^{m} X_{i,j} \\ MaxH &= MAX\left(H^{t}_{PROJ}\right) \end{aligned} \tag{2.8}$$

9. MaxV is maximum point of projection in vertical direction

$$
\begin{aligned}
V'_{PROJ} &= [V_1, \quad V_2, \quad V_3, \quad \ldots\ldots, \quad V_n]^T \\
V_i &= \sum_{i=1}^{n} X_{i,j} \\
MaxV &= MAX\left(V'_{PROJ}\right)
\end{aligned}
\tag{2.9}
$$

Then features of character can be represented as:

$$
F_t = \{\text{TOT, UP, DOWN, LEFT, RIGHT, MaxH, MaxV, DW\_UP, RG\_LF}\}^T \tag{2.10}
$$

This feature value is translated to fuzzy set domain as shown in the Fig 2.24 and used as input for back-propagation neural network. The network has three layers with 9 nodes input layer, 120 nodes hidden layer and 46 nodes output layer. The average recognition rate is about 93.50%.

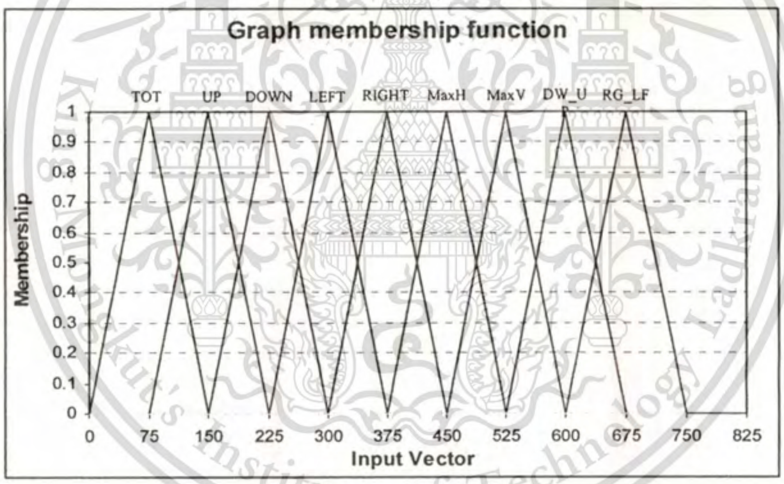


Fig 2.24 Graph of fuzzy membership function in each term [21].

Chapter 3

Feature extraction

Performance of handwritten recognition system depends on main two attributes. The first one is the selected feature gathering from a handwritten character. The second one is the recognizers training to remember the feature of each character in order to provide correct recognition with good generalization. This section describes feature extraction of Lao handwritten character recognition system.

3.1 Input data.

In this dissertation, the Lao handwritten character system receives input as a sequence of X, Y coordinates from tablet with the rate of 100 samples per second in surface of 100 lines per inch. The information from a tablet is frequently sent to the computer which performs the recognition task as shown in the Fig 3.1. A tablet has constant sampling rate in time. So length between two sampling points in a sequence of X, Y coordinate is not equal. It depends on a writing speed. If the writing speed is faster the length between two sampling point is longer. But if speed of writing is slower the length between two sampling points is shorter as shown in Fig 3.2.

Fig 3.1 Handwritten input as a sequence of X, Y coordinates of pen tracing.

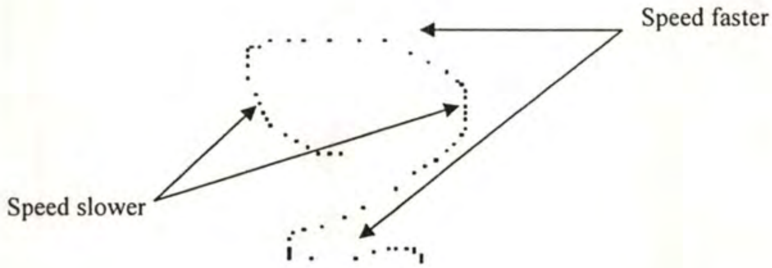


Fig 3.2 Handwritten speed effect on sampling point of pen tracing.

The handwritten character always has noises and variation due to writing environment such as writing speed, writing person. A "hook" noises appear at the starting and the ending of character as shown in the Fig 3.3. These noises come out because of hand shaking when putting pen down or taking pen up when finish.

Fig 3.3 Hooks at head and tail of character.

3.2 Lao handwritten character feature extraction.

In this dissertation, a handwritten character is represented as a sequence of curvature where the current curvature and the next one in the sequence have opposite direction (clockwise and counter-clockwise). In other words, a handwritten is separated into segments at the point changing a drawing direction (from counter-clockwise to clockwise and vice versa). The representation of a character as a curve sequence is shown in Fig 3.4. The rest of this section describes how to obtain the curvature sequence feature from the raw sampling of handwritten in X,Y coordinates.

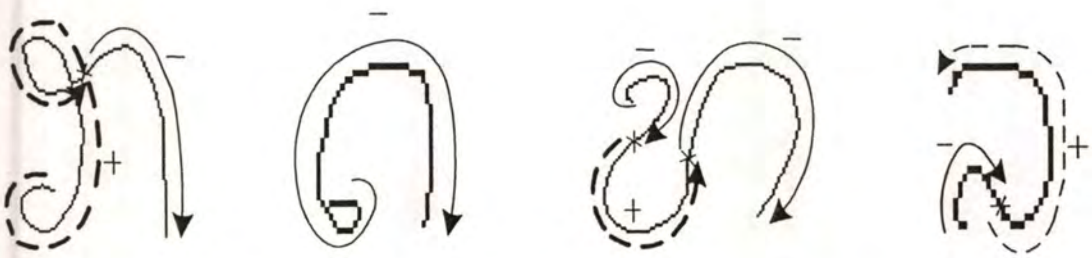


Fig 3.4 Lao handwritten character perceive as a sequence curvature.

The sequence of X, Y coordinates from tablet is translated to the sequence of directional code (*DC*). The DC is a floating point version of Chain Code [31]. Obtaining *DC* is fast and easy. It needs no normalization of input X,Y sampling, such as normalization of equi-distance between sampling points. It is translate from floating point X,Y coordinates of sampling point pen trajectory. The directional code is also in real number not integer like normal chain code as shown in Fig 3.5. The 45-degree means one unit (360-degree is 8 units). For the convenience of explanation, an example of the directional code calculation is shown in Fig 3.5. The directional code of vectors A (from point 0,0 to 2,3) is given as 1 + 2/3.

A directional codes starting from a point in the X,Y coordinates to the next point can be calculated using the equation (3.1) below. The equation is based on the X,Y axis shown in Fig 3.6, where X is positive on the right of original point and Y is positive downward below original point. The calculation examples are shown in Fig 3.6.

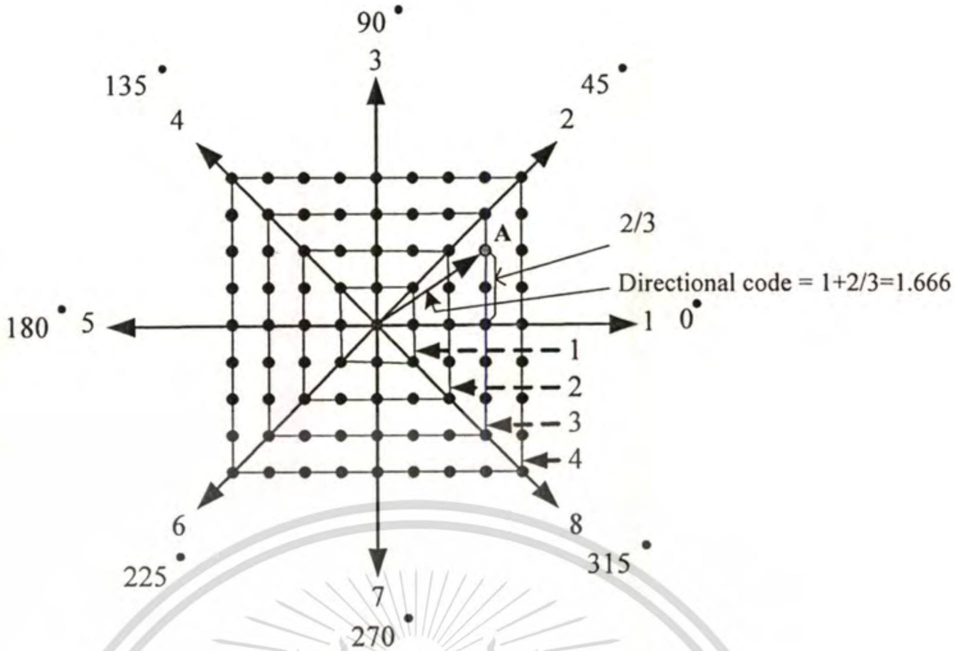


Fig 3.5 The directional code calculation form GCC.

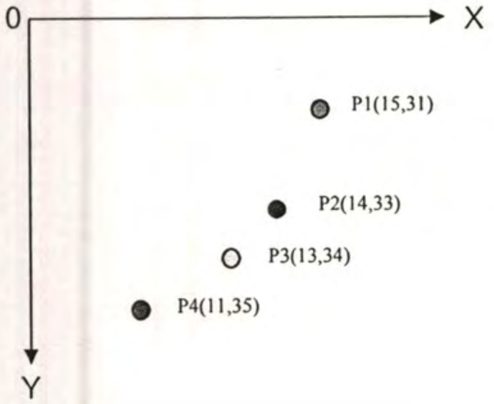
$$DC_i = \begin{cases} 1 + \dfrac{|dy_i|}{dx_i}; & dx_i > 0; dy_i \le 0; dx_i \ge |dy_i| \\ 3 - \dfrac{dx_i}{|dy_i|}; & dx_i > 0; dy_i < 0; dx_i < |dy_i| \\ 3 + \dfrac{dx_i}{dy_i}; & dx_i \le 0; dy_i < 0; |dx_i| \le |dy_i| \\ 5 - \dfrac{dy_i}{dx_i}; & dx_i < 0; dy_i < 0; |dx_i| > |dy_i| \\ 5 + \dfrac{dy_i}{|dx_i|}; & dx_i < 0; dy_i \ge 0; |dx_i| \ge dy_i \\ 7 - \dfrac{|dx_i|}{dy_i}; & dx_i < 0; dy_i > 0; |dx_i| < dy_i \\ 7 + \dfrac{dx_i}{dy_i}; & dx_i \ge 0; dy_i > 0; dx_i \le dy_i \\ 9 - \dfrac{dy_i}{dx_i}; & dx_i > 0; dy_i > 0; dx_i > dy_i \end{cases} \tag{3.1}$$

Where: $x = \{x_1, x_2, x_3 \ldots\ldots x_N\}$

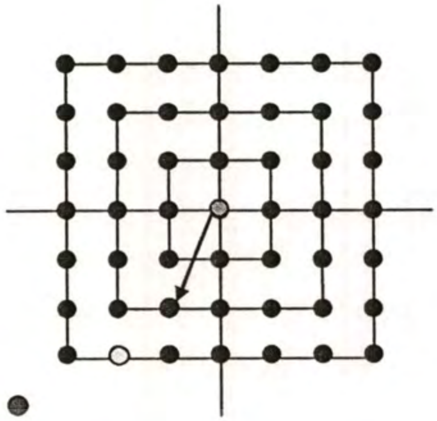
$y = \{y_1, y_2, y_3 \ldots\ldots y_N\}$

$dx_i = x_{i+1} - x_i$; dx is difference in sequence of x axis.

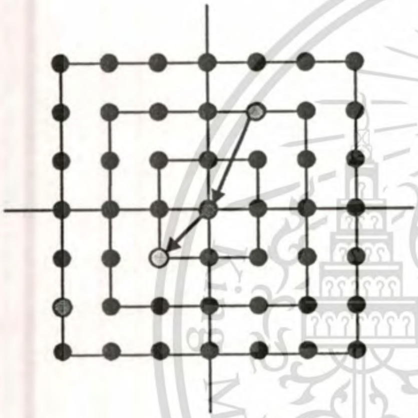
$dy_i = y_{i+1} - y_i$; dy is difference in sequence of y axis.



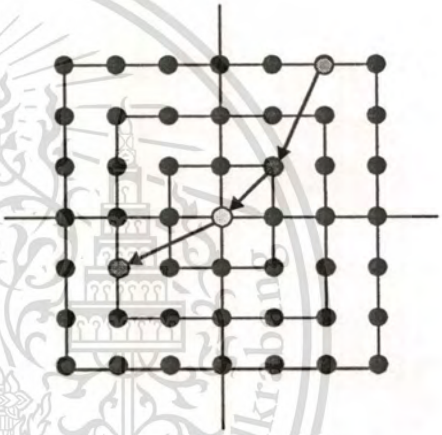
(A). Sequence of X, Y coordinates of samples



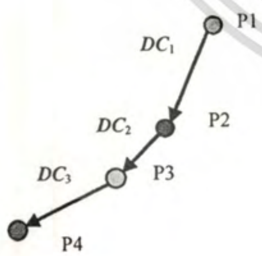
(B). Directional code, $DC_1 = 7\text{-}1/2 = 6.5$



(C). Directional code, $DC_2 = 5+1/1 = 6$



(D). Directional code, $DC_3 = 5+1/2 = 5.5$



Sequence of Directional code

$DC_1 = 7\text{-}1/2 = 6.5000$

$DC_2 = 5+1/1 = 6.0000$

$DC_3 = 5+1/2 = 5.5000$

(E). Sequence of directional codes

Fig 3.6 Examples input data translation to *DC*s.

All the points in sequence X, Y coordinates are translated in to sequence of directional codes and lengths. The length is the length between two consecutive of points in the sequence of X, Y coordinates. The length is defined here using "L", which can be calculated as in equation (3.2). Fig 3.7 shows example of L calculation.

$$L_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad , \quad i = 0; \quad i < N \tag{3.2}$$

Where: L_i is a data in sequence of lengths of Lao handwritten character.

N is the number of sampling point

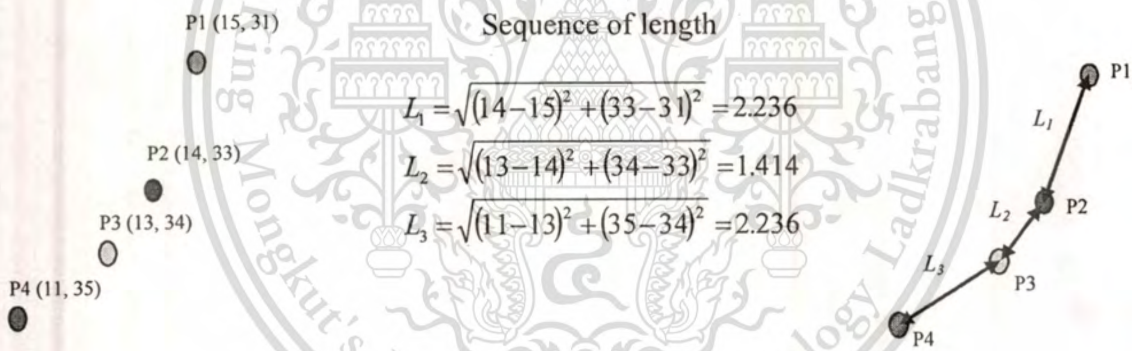
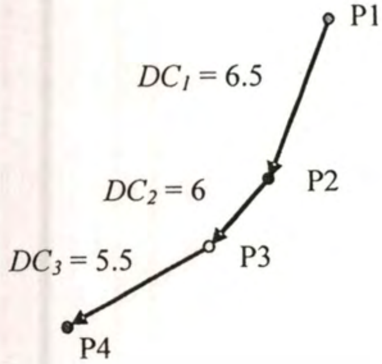


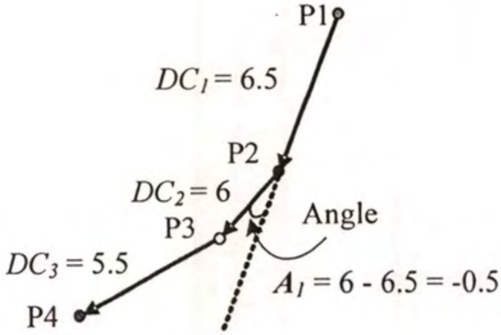
Fig 3.7 Length calculation between two sampling points.

The next step is to obtain sequences of angle from DC sequence. An angle is defined as the change (difference) in direction between two DCs in a sequence. The angle is defined here using "A", which can be calculated as in equation (3.3). Fig 3.8 shows example of angle, A, calculation.

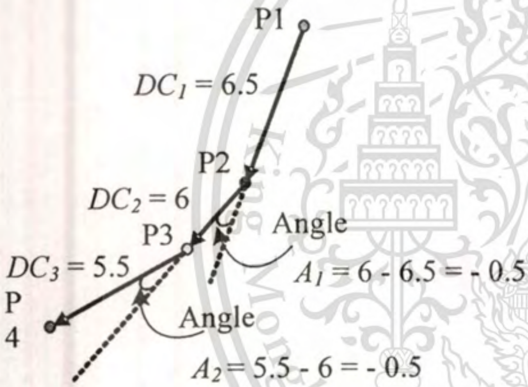
$$A_i = Dc_{i+1} - Dc_i \quad , \quad i = 0; \quad i < N - 1 \tag{3.3}$$



(A). Sequence of directional code



(B). Different of DC_1 and DC_2 is angle number 1 (A_1)



Sequence of angle

$A_1 = 6 - 6.5 = - 0.5$

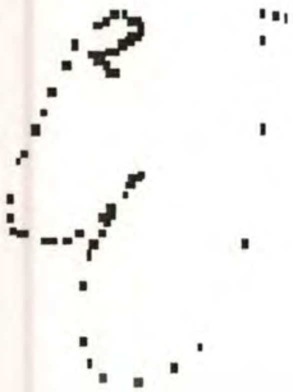
$A_2 = 5.5 - 6 = - 0.5$

(C). Different of DC_2 and DC_3 is angle number 2 (A_2)

(D). Sequence of angle

Fig 3.8 Angles calculation

Fig 3.9 shows Lao handwritten character with its X, Y coordinates sequence, directional codes sequence, lengths sequence and angles sequence.



Index [i]	X	Y	Directional code	Length (L)	Angle (A)
0	200	234	5.000	1.000	-1.000
1	199	234	4.000	1.414	0.000
2	198	233	4.000	1.414	1.000
3	197	232	5.000	1.000	-1.000
4	196	232	4.000	1.414	-3.000
5	195	231	1.000	1.000	0.000
6	196	231	1.000	1.000	0.590
7	107	231	1.590	2.236	-0.590
8	199	230	1.000	1.000	1.000
9	200	230	2.000	1.414	-1.000
10	201	229	1.000	1.000	1.000
11	202	229	2.000	1.414	-1.000
12	203	228	1.000	1.000	2.000
13	204	228	3.000	1.000	-2.000
14	204	227	1.000	1.000	1.000
15	205	227	2.000	1.414	1.000
16	206	226	3.000	1.000	1.000
17	206	225	4.000	1.414	1.000
18	205	224	5.000	1.000	-0.590
19	204	224	4.410	2.236	0.590
20	202	223	5.000	2.000	0.590
21	200	223	5.590	2.236	0.000
22	198	224	5.590	2.236	0.269
23	196	225	5.859	6.403	0.551
24	191	229	6.410	4.472	-0.098
25	189	233	6.312	. 5.831	0.098
26	186	238	6.410	4.472	-0.158
27	184	242	6.251	3.606	0.061
28	182	245	6.312	5.831	0.098
29	179	250	6.410	2.236	0.236
30	178	252	6.646	7.280	0.354
31	176	259	7.000	4.000	0.590
32	176	263	7.590	2.236	0.410
33	177	265	8.000	1.414	1.000
34	178	266	1.000	1.000	-0.251
35	179	266	8.749	5.099	0.251
36	184	267	1.000	2.000	0.000
37	186	267	1.000	3.000	0.410
38	189	267	1.410	3.162	0.278
39	192	266	1.688	5.831	0.312
40	197	263	2.000	2.828	0.000
41	199	261	2.000	7.071	0.000
42	204	256	2.000	1.414	0.000
43	205	255	2.000	1.414	3.590
44	206	254	5.590	2.236	0.410
45	204	255	6.000	1.414	0.410
46	203	256	6.410	2.236	-0.229
47	202	258	6.181	5.000	0.229
48	199	262	6.410	2.236	0.000
49	198	264	6.410	2.236	-0.410
50	197	266	6.000	2.828	0.410
51	195	268	6.410	2.236	0.380
52	194	270	6.790	6.083	0.210
53	193	276	7.000	11.00	0.312
54	193	287	7.312	4.123	0.688
55	194	291	8.000	4.243	0.842
56	197	294	8.842	8.062	0.615
57	205	295	1.457	8.544	0.292
58	213	292	1.749	7.211	0.661
59	219	288	2.410	22.361	0.371
60	229	268	2.781	23.345	0.219
61	233	245	3.000	17.000	0.000
62	233	228	3.000	5.000	-2.500
63	233	223	8.500	2.236	0.000
64	235	224	0.000	0.000	0.000

Fig 3.9 Example of Lao handwritten character and its X,Y coordinates, directional codes , lengths and angles.

3.3 Removing of hook at head and tail of a character

The hooks at head and tail of character are appeared because of high acceleration of pen movement at the starting and stopping of a character writing. These hooks need to be removed from the character before a character feature extraction and recognition.

It is easy to find these hooks because of its high amount of angle value. We can find these hooks at the head and tail of character by monitoring the angle at the head and tail of character. The angle is a hook if the angle has bigger value than some threshold and its position is near the starting and ending of a character.

The process of removing hooks from head and tail of character is described here. First define the angle threshold (*AT*) and boundary threshold (*BT*) which is some boundary from starting and ending of character that perform hook monitoring. In this dissertation, *AT* is 2.5. The angle amount of 2.5 is shown in Fig 3.10 below.

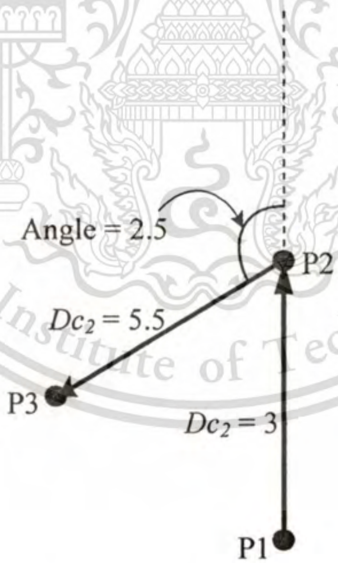


Fig 3.10 Angle threshold of 2.5

The boundary length threshold (*BLT*) is set to 15% of the total length of a character or fix number is 15 as shown in equation (3.4).

$$BLT = \min\begin{cases} 0.15 * TotalLength \\ 15 \end{cases} \tag{3.4}$$

Where: *Total Length* is Sum of all the Euclidean distance between all the sampling points.

Sequence of N points sampling from tablet are indexed from 0 to N-1 as shown in Fig 3.11. Another boundary threshold is boundary point threshold (*BPT*). *BPT* is set to 30% of total sampling points or fix the number of sampling points from starting and ending instead of using the normalized distance. If *BPT*=3, hook has to be checked from starting point to point number 3 or from ending point to 3 points back. In this dissertation, *BPT* is set to the maximum of 30% of all sampling points or 6 points, as in equation (3.5).

$$BPT = \min\begin{cases} 0.3 * N \\ 6 \end{cases} \tag{3.5}$$

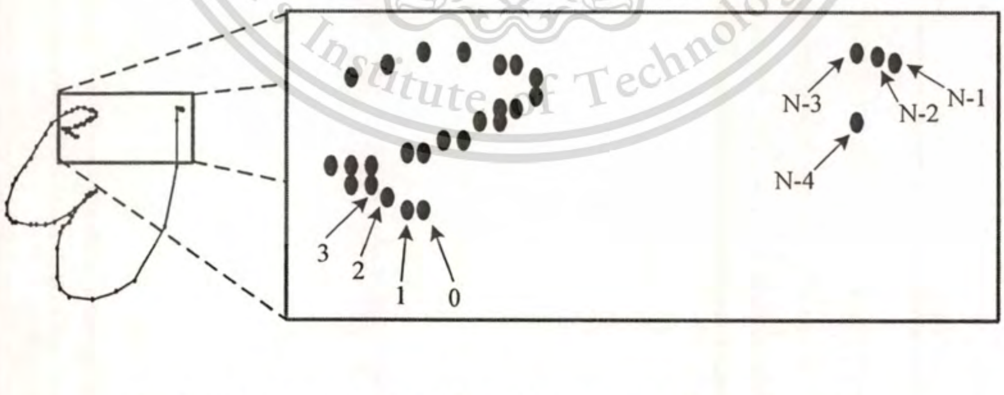
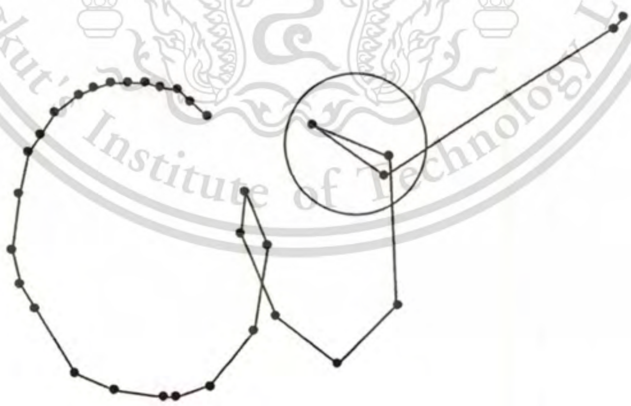


Fig 3.11 Show points index from starting and ending of character

The reason to use two boundaries in length and in points is due to the variation in size and speed of handwritten. In length, boundary is set to minimum of 15% or 15 of total length. If writer writes with very low speed of handwritten this boundary might be too many number of points then they have to be checked as shown in Fig 3.12 (A). Normally hook is always found in boundary 6 points from start or end of character. On the other hand, if boundary was just the point, the boundary is set to maximum of 6 or 30% of all the points. If a writer writes with fast speed, this boundary might be too large (cover the real part of a character not just the beginning), as shown in Fig 3.12 (B).

(A)



(B)

Fig 3.12 (A) Boundary length for check hook. (B) Main dominance of character like as hook

The examples of removed hook from character are shown in Fig 3.13, Fig 3.14 below.

Index [i]	X	Y	Directional code	Length (L)	Angle (A)
0	**200**	**234**	**5.000**	**1.000**	**-1.000**
1	**199**	**234**	**4.000**	**1.414**	**0.000**
2	**198**	**233**	**4.000**	**1.414**	**1.000**
3	**197**	**232**	**5.000**	**1.000**	**-1.000**
4	**196**	**232**	**4.000**	**1.414**	**-3.000**
5	195	231	1.000	1.000	0.000
6	196	231	1.000	1.000	0.590
7	107	231	1.590	2.236	-0.590
8	199	230	1.000	1.000	1.000
9	200	230	2.000	1.414	-1.000
10	201	229	1.000	1.000	1.000
11	202	229	2.000	1.414	-1.000
12	203	228	1.000	1.000	2.000
13	204	228	3.000	1.000	-2.000
14	204	227	1.000	1.000	1.000
15	205	227	2.000	1.414	1.000
16	206	226	3.000	1.000	1.000
17	206	225	4.000	1.414	1.000
18	205	224	5.000	1.000	-0.590
19	204	224	4.410	2.236	0.590
20	202	223	5.000	2.000	0.590
21	200	223	5.590	2.236	0.000
22	198	224	5.590	2.236	0.269
23	196	225	5.859	6.403	0.551
24	191	229	6.410	4.472	-0.098
25	189	233	6.312	5.831	0.098
26	186	238	6.410	4.472	-0.158
27	184	242	6.251	3.606	0.061
28	182	245	6.312	5.831	0.098
29	179	250	6.410	2.236	0.236
30	178	252	6.646	7.280	0.354
31	176	259	7.000	4.000	0.590
32	176	263	7.590	2.236	0.410
33	177	265	8.000	1.414	1.000
34	178	266	1.000	1.000	-0.251
35	179	266	8.749	5.099	0.251
36	184	267	1.000	2.000	0.000
37	186	267	1.000	3.000	0.410
38	189	267	1.410	3.162	0.278
39	192	266	1.688	5.831	0.312
40	197	263	2.000	2.828	0.000
41	199	261	2.000	7.071	0.000
42	204	256	2.000	1.414	0.000
43	205	255	2.000	1.414	3.590
44	206	254	5.590	2.236	0.410
45	204	255	6.000	1.414	0.410
46	203	256	6.410	2.236	-0.229
47	202	258	6.181	5.000	0.229
48	199	262	6.410	2.236	0.000
49	198	264	6.410	2.236	-0.410
50	197	266	6.000	2.828	0.410
51	195	268	6.410	2.236	0.380
52	194	270	6.790	6.083	0.210
53	193	276	7.000	11.00	0.312
54	193	287	7.312	4.123	0.688
55	194	291	8.000	4.243	0.842
56	197	294	8.842	8.062	0.615
57	205	295	1.457	8.544	0.292
58	213	292	1.749	7.211	0.661
59	219	288	2.410	22.361	0.371
60	229	268	2.781	23.345	0.219
61	233	245	3.000	17.000	0.000
62	**233**	**228**	**3.000**	**5.000**	**-2.500**
63	**233**	**223**	**8.500**	**2.236**	**0.000**
64	**235**	**224**	**0.000**	**0.000**	**0.000**

Fig 3.13 Example of hooks locating at head and tail of character.

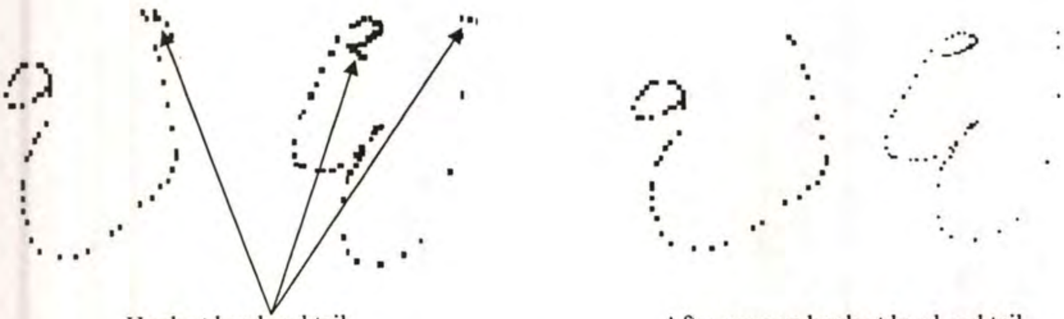


Fig 3.14 Hook removing example.

3.4 Segmentation.

The final part toward obtaining curvature sequence of a handwritten character is to divide a character into segments and add all the angle of all the points in a segment. There are many variations in this feature depending on a segmentation choice.

3.4.1 Segmentation using dominant point.

In the previous research Hung Yuen [31], divided handwritten character in to segments by using dominant point or point where the pen trajectory turns the direction quickly as a segmentation point, shown in the Fig 3.15. From our experiment, segmentation by using retrace point has a lot of variations in number of segments. The same character written in different style get different numbers of segmentation points. The position of segmentation points also varies as shown in Fig 3.16. The segmentation point is very sensitive to the threshold angle selection of retrace point. If the threshold value is smaller, the number of segmentation points will be more as shown in Fig 3.16. Conversely, if it is too large, there may be no segmentation points as shown in the Fig 3.17. From the experiment, this segmentation method is not suitable for Lao character. It is more suitable for English character as shown in Fig 3.18..

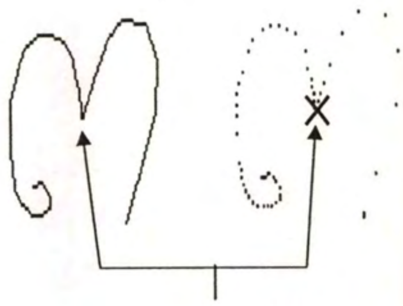


Fig 3.15 Segmentation points of character using retrace point.

Fig 3.16 The same character with difference number and position of segmentation points.

Fig 3.17 The threshold value is too big to find the segmentation points.

Fig 3.18 Segmentation points using retracing point [31].

3.4.2 Segmentation using direction change of handwritten.

As previously mentioned Lao handwritten character is a continuous drawing of curves with clockwise and counter clockwise directions. It is very natural to represent Lao handwritten character as a sequence of curve segments with clockwise and counter clockwise directions as the main feature of character. In order to restrict curvature sequence to start in one direction (clockwise) followed by the opposite direction (counter-clockwise), the segmentation point should be at the point where a curvature starts to change direction. The segmentation point is the point where a curvature changes direction in clockwise and counter clockwise directions is shown in Fig 3.19.

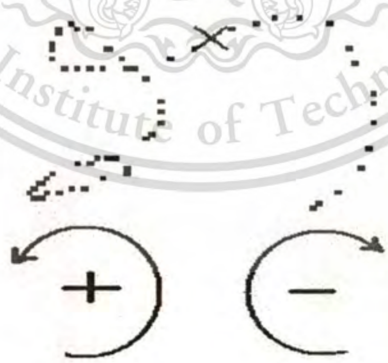


Fig 3.19 Segmentation using curvature rotation in clockwise and counter clockwise directions.

Since segmentation point occurs the change in direction with clockwise and counter clockwise in a pen trajectory. It can be located using angle sequence at the point changing angle sign. The angle between two consecutive directional codes will have positive value if character written in counter clockwise direction and negative value in clockwise direction. This is because, when a character is written in counter clockwise direction, the current directional code is kept increasing from the previous one. Hence the angle, obtained from subtracting the current directional code from the previous one, will have positive value. This is shown in the Fig 3.20.

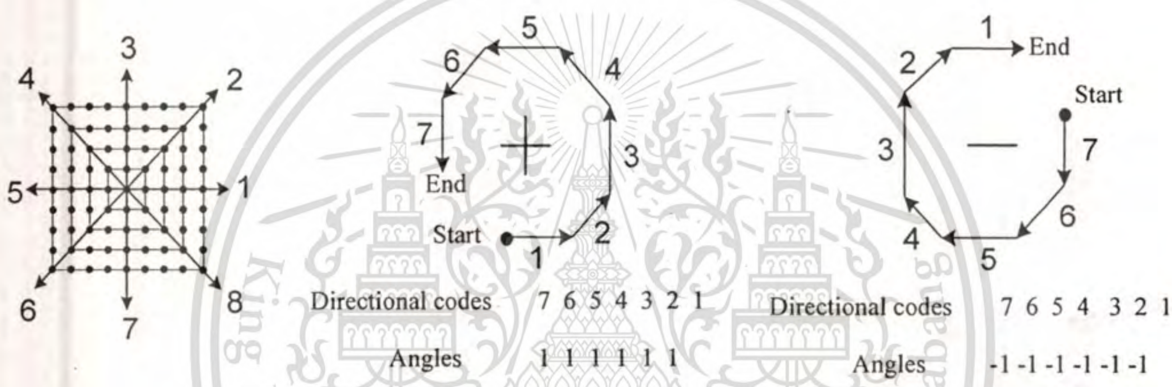


Fig 3.20 The relation of angle sign and writing direction in clockwise and counter clockwise direction.

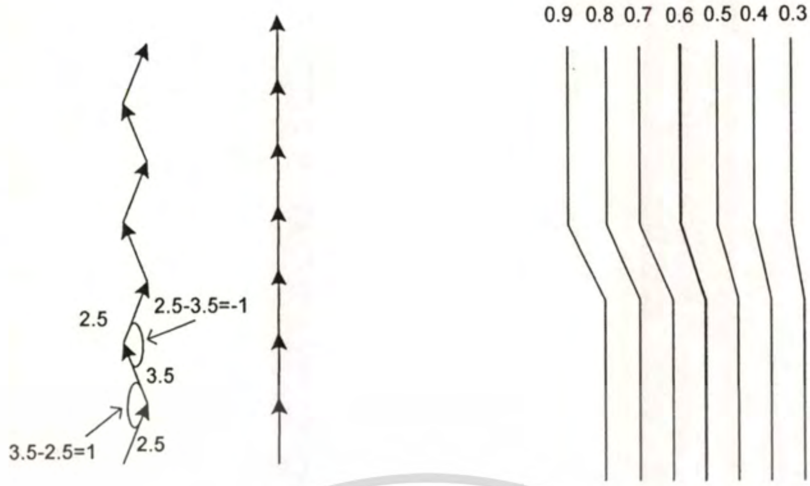
The segmentation points can be located at the sampling point where the angle sign is changed. This change from positive to negative and negative to positive occurs. In other words, a segmentation point is the point changing the direction of pen trajectory from the clockwise to the counter clockwise direction and vice versa. Since there may be some noises in handwritten, some points having a sign change are not treated as segmentation points. The details of segmentation point identification are shown in table 3.1.

Table 3.1 Segmentation points

Sign of angle					Comment
i-2	*i*-1	*i*	*i*+1	*i*+2	
+	+	-	-	-	The segmentation point is at *i*.
+	+	-	+	+	• The segmentation point is at *i* if $\|A_i\|$ >= Threshold. • In case of $A_i = A_{i-1}$ or $A_i = A_{i+1}$ point *i* is not a segmentation point.

Where: *i* is the index and Threshold is 0.6.

In order to represent Lao handwritten characters with segmentation points effectively, the noise such as small variation or fluctuation in the pen trajectory should be removed. In this thesis the noise cancellation is made as shown in Fig 3.21(A) and Fig 3.21(B). When the direction of pen trajectory is not alignment but in zigzags (clockwise and counter clockwise curve segments with small equal angle), the zigzags are considered as a straight line. In other words, the pen trajectory with the zigzags segments is considered as a straight line if sum of the angles of curve segments between the consecutive segmentation points is equal to zero. Another kind of noise is a small and quick change of curvature as show in Fig 3.21(B). In this case, pen trace is considered as a straight line according to the amount of the angle of the curve segments. The point is not considered as the segmentation point if the angle is less than a selected threshold angle. In Fig 3.21(B), pen trajectory is considered as straight line if the amount of the angle of the curve segments is less than threshold value (0.6). Otherwise it is treated as a zigzag line.



(A) (B)

Fig 3.21 (A) Quantized noise cancellation, (B) The threshold 0.6 curvature

Fig 3.22 shows some examples of segmentation points of Lao handwritten characters characters. Fig 3.23 shows the details calculation of the segmentation points of hand writing Lao character "ຕ". At index 2, the angle is changing from -2, 0, 1 but the next angle at index 3 is -1 which is exactly opposite direction (summation of both angles 1 and -1 equal to zero), hence, the point at index 2 is not treated as segmentation point.

Lao characters

Fig 3.22 Examples of segmentation points of handwritten characters.

Segmentation points number	Index [i]
1	0
2	20
3	22
4	31
5	37
6	52
7	53
8	55
9	58

Index [i]	X	Y	Directional code	Length (L)	Angle (A)
0	**349**	**361**	**3.000**	**1.000**	**-2.000**
1	349	360	1.000	1.000	0.000
2	350	360	1.000	1.000	1.000
3	351	360	2.000	1.414	-1.000
4	352	359	1.000	1.000	0.000
5	353	359	1.000	1.000	-1.000
6	354	359	8.000	1.414	-2.000
7	355	360	6.000	1.414	-1.000
8	354	361	5.000	1.000	1.000
9	353	361	6.000	1.414	-1.000
10	352	362	5.000	1.000	0.000
11	351	362	5.000	1.000	0.000
12	350	362	5.000	1.000	0.000
13	349	362	5.000	1.000	-1.000
14	348	362	4.000	1.414	-1.000
15	347	361	3.000	1.000	0.000
16	347	360	3.000	1.000	0.000
17	347	359	3.000	1.000	0.000
18	347	358	3.000	1.000	0.000
19	347	357	3.000	2.000	-0.590
20	**347**	**355**	**2.410**	**2.236**	**0.181**
21	348	353	2.590	3.162	1.000
22	**349**	**350**	**3.590**	**2.236**	**-2.000**
23	348	348	1.590	2.236	0.819
24	350	347	2.410	2.236	-0.819
25	351	345	1.590	2.236	0.410
26	353	344	2.000	1.414	-1.000
27	354	343	1.000	1.000	-2.000
28	355	343	7.000	1.000	2.000
29	355	344	1.000	1.000	-1.000
30	356	344	8.000	1.414	0.000
31	**357**	**345**	**8.000**	**1.414**	**1.000**
32	358	346	1.000	1.000	0.000
33	359	346	1.000	1.000	1.000
34	360	346	2.000	1.414	0.000
35	361	345	2.000	1.414	-1.000
36	362	344	1.000	1.000	1.000
37	**363**	**344**	**2.000**	**1.414**	**-0.410**
38	364	343	1.590	2.236	-0.590
39	366	342	1.000	1.000	1.000
40	367	342	2.000	1.414	-1.000
41	368	341	1.000	3.000	0.000
42	371	341	1.000	1.000	0.000
43	372	341	1.000	1.000	0.000
44	373	341	1.000	2.000	0.000
45	375	341	1.000	1.000	-0.410
46	376	341	8.590	3.162	0.410
47	379	342	1.000	1.000	-1.000
48	380	342	8.000	1.414	0.000
49	381	343	8.000	1.414	-1.000
50	382	344	7.000	1.000	0.000
51	382	345	7.000	2.000	-1.000
52	**382**	**347**	**6.000**	**1.414**	**0.688**
53	**381**	**348**	**6.688**	**4.123**	**-0.437**
54	380	352	6.251	3.606	-0.251
55	**378**	**355**	**6.000**	**1.414**	**1.000**
56	377	356	7.000	1.000	0.000
57	377	357	7.000	1.000	
58	**377**	**358**			

Fig 3.23 Locate segmentation points by monitoring angle signs in the sequence.

After all the segmentation points are located, the angle of each curve segment consists of all the angle of all the points in a segment is calculated from the summation of smaller angle, it is abbreviated as *SA*. The length of a curve segment is also calculated from the summation of the distance between all points in a segments, it is abbreviated as *SL*. At this step, the Lao handwritten character can be represented as a sequence of curve segment with clockwise direction and counter clockwise direction and vice versa as shown in Fig 3.24.

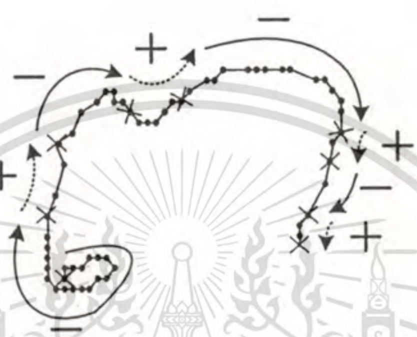


Fig 3.24 Sequence of curvature segments of character

Curve segment may still have a lot of small angle curves. These small curve segments are considered as noise if it is smaller than some threshold. After all noises have been removed, a character can be represented by a curve segment sequence. The recognition accuracy of this representation depends on the correctness of these small noise curvatures identification, which directly depends on the threshold selection. If the threshold is small, there can be more segments due to inclusion of noisy segments. On the other hand, if the threshold is large, some small real curvature segments will be missing

3.5 Alternative curvature tree representation of character.

This section explains the construction of alternate curvature tree for Lao handwritten character. From section 3.4.2, a character is represented as a single sequence of curve segments with clockwise (-) or counter clockwise direction (+) alternately. As previous mentioned, the problem of this representation is the decision of threshold whether some small curvature segment could be a noise or a real intension curve in a character. In sequence of curve segments, a segment with big amount of curvature is easily identified as real curve segments of handwritten character. Also, a segment with very small amount of curvature is easily identified as noise. But, there are some segments with relatively small value of curvature which can not be certainly identified as either noise or real curvature. These segments sometimes are real segments but sometimes they are noise segments. Usually, the noise curve segment can be distinguished from the real curve by the high amount of curvature of the real signal. But when the written speed is fast the curvature is reducing close to the noisy curve as shown in Fig 3.25. These noisy curves are hard to distinguish with normal noise reduction technique such as Fourier expansion or Fourier descriptor due to their similarity in frequency and amplitude.

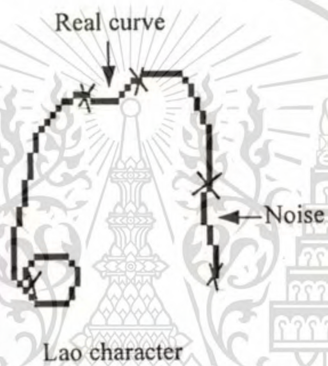


Fig 3.25 Uncertain segments with small curvature.

In order to successfully use the curvature sequence in the recognition one has to identify real curvature from the noisy one. As shown in Fig 3.26, the character "ຕ" consist of three curvatures in sequence. If all small curvature segments are treated as noise and omitted, the character will match to another character "ດ" that has one clockwise curvature. On the other hand, if all curvatures are treated as real curve the character will have 4 curvatures and it can not match to the correct character with three curvature segments.

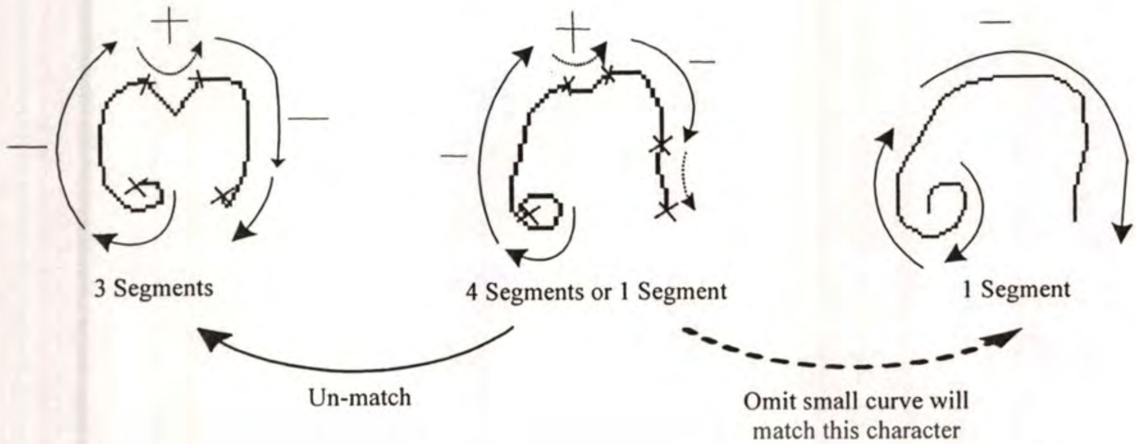


Fig 3.26 Uncertain segment and its effect in recognition.

So in this dissertation, we classify curve segment into three groups, noise segment, real segment and uncertain segments. The noise group consists of the noise curve that can be identified and removed. The uncertain group consists of noise and real signal that is hard to distinguish. The real group has high curvature enough to be certain as a real signal. The handwritten character can then be represented as a binary tree of curve segments, instead of just a single curvature sequence. The branch in the tree represents uncertain curve segment since an uncertain segment can both be a real segment or a noise segment. Hence, the feature of Lao handwritten character is not just a single sequence of curve but a tree of curve segments.

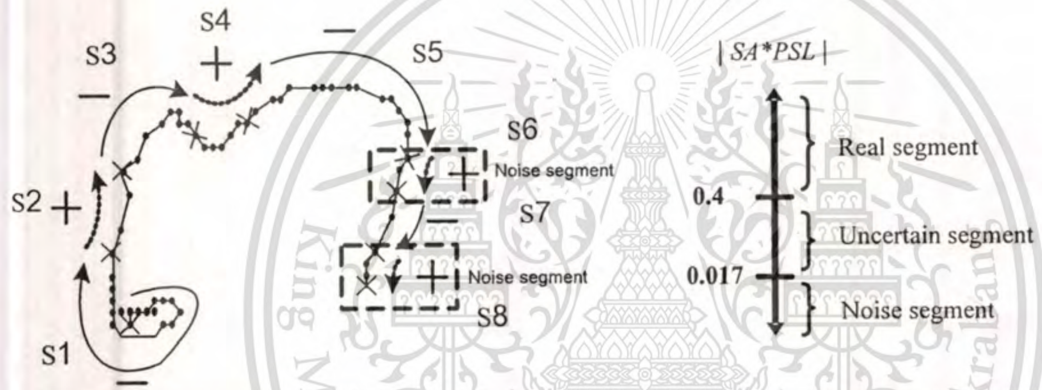
The sequence consisting of clockwise and counter clockwise curve segments is used to construct alternative curvature tree as followed:

1: Classify each curve segment as one of the following segments: noise segment, uncertain segment and real segment by using length and angle of the segment. Discard the noise segment by merging noise segment with the previous segment and subsequent segment.

2: Re-classify each merged segment as either uncertain or real. If any adjacent uncertain segments has the same amount of angles but different signs (- clockwise, + counter clockwise), then both uncertain segments are treated as a noises with zero angle and merge both adjacent segments as mentioned above.

3: Construct binary tree of curve segments consisting of the real segments and uncertain segments.

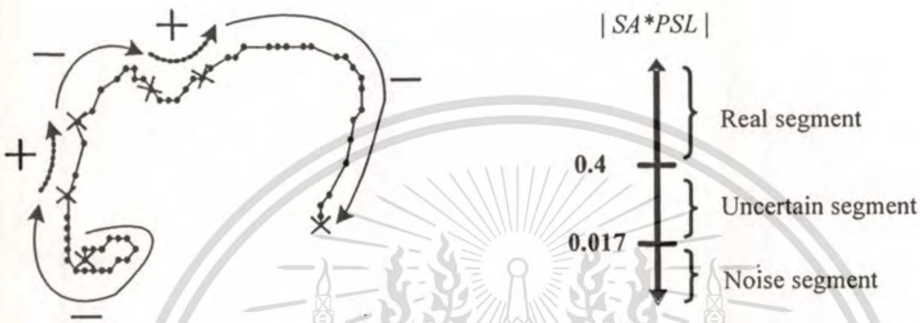
The real curve segment should have high curvature and significant length. Some noise segments may have high curvature with short length or small curvature with long length but not both. Hence, the noise segment can be classified here using angle-length feature, the angle multiplied by the length (the ratio of the length over the summation of the entire segments length). From observation from Lao handwritten database, if the angle-length feature is less than a threshold value (0.017) or angle <=1 and length <=5 pixels then the segment is considered as a noise segment. If it is between 0.017 and 0.4, then it is considered an uncertain segment and furthermore if it is greater than 0.4, then it is considered as a real segment as shown in Fig 3.27.



R = Real segment, U = Uncertain segment, N = Noise segment								
Segment number	S1	S2	S3	S4	S5	S6	S7	S8
Sum Angle (*SA*)	-8.59033	1.18066	-3.59033	2	-4	0.068808	-0.068808	1
Sum Length (*SL*)	23.07107	5.39835	14.77207	7.24264	24.0552	1.41421	7.72866	3.41421
% Sum Length (*PSL*)	0.26489	0.06198	0.16961	0.08316	0.27619	0.01624	0.08874	0.03920
| *SA*PSL* |	2.27549	0.07318	0.60896	0.16631	1.10478	0.01117	0.06106	0.03920
Segment status	**R**	**U**	**R**	**U**	**R**	**N**	**U**	**N**

Fig 3.27 Examples of noise segment, uncertain segment, and real segment calculation.

The noise segment is removed from the handwritten by adding its angle and its length both adjacent segments (previous and subsequent segments) to form a substitute segment. This segment replaces the 3 original segments. In other word, the noise segments are merged with the adjacent segments into a single segment with angle and length equal to the summation of its original 3 segments. The purpose of the addition is to preserve the length feature since some segment may have small turn but its length is still significant in feature representation of the character. The noise classification of the merged segments is repeated until this classification results in only uncertain and real segments without noise as shown in Fig 3.28.



R = Real segment,	U = Uncertain segment,		N = Noise segment		
Segment number	S1	S2	S3	S4	S5
Sum Angle (*SA*)	-8.59033	1.18066	-3.59033	2	-3
Sum Length (*SL*)	23.07107	5.39835	14.77207	7.24264	36.61228
% Sum Length (*PSL*)	0.26489	0.06198	0.16961	0.08316	0.42036
| *SA*PSL* |	2.27549	0.07318	0.60896	0.16631	1.26109
Segment status	**R**	**U**	**R**	**U**	**R**

Fig 3.28 After removing of noises segments, only uncertain and real curve segment remain.

Now, the curve sequence is a sequence of clockwise and counter clockwise curve segments with some sequent label as an uncertain segment. In the next step, a binary tree of all possible representation is formed, where each uncertain segment can be treated as a noise segment or real segment and forms the binary branch of the tree. When uncertain segment is treated as a noise segment, the adjacent segments (three segments) are merged together, as shown in Fig 3.29 – Fig 3.38. The figure also shows the tree representation of character.

As mentioned previously, the purpose of feature tree is to keep all the possible representations of a character. At this moment, we still do not know is which the real curve sequence due to uncertainty. But, the real representation can be identified by comparing with the other characters.

R = Real segment, U = Uncertain segment, N = Noise segment					
Segment number	**S1**	**S2**	**S3**	**S4**	**S5**
Sum Angle (*SA*)	-8.59033	1.18066	-3.59033	2	-3
Sum Length (*SL*)	23.07107	5.39835	14.77207	7.24264	36.61228
% Sum Length % (*PSL*)	0.26489	0.06198	0.16961	0.08316	0.42036
| *SA*PSL* |	2.27549	0.07318	0.60896	0.16631	1.26109
Segment status	R	U	R	U	R

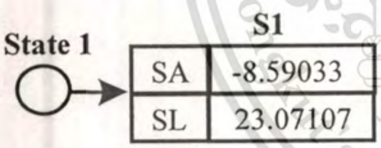


Fig 3.29 State 1 of tree generations at segment number 1 (real segment).

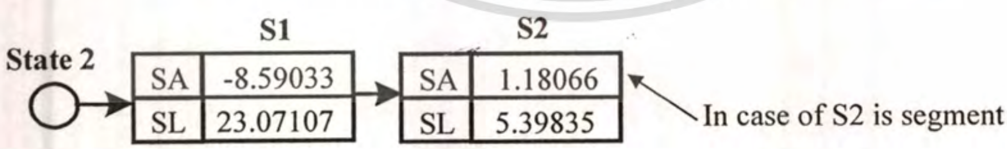


Fig 3.30 State 2 at segment number 2 (uncertain segment and assumes it is real segment).

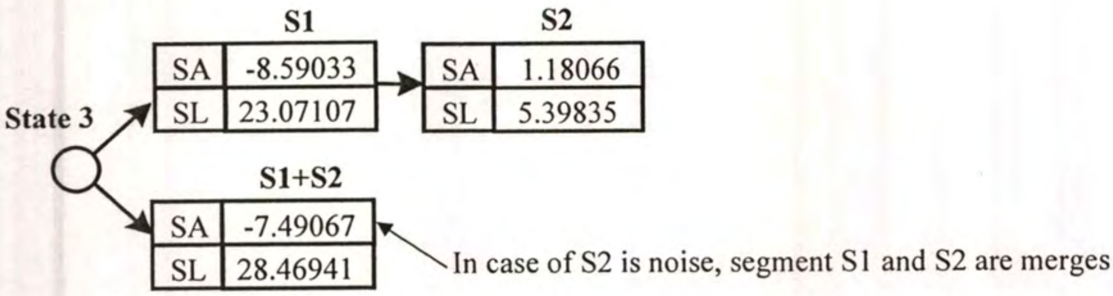


Fig 3.31 State 3 at segment number 2 (uncertain segment with noise segment assumption).

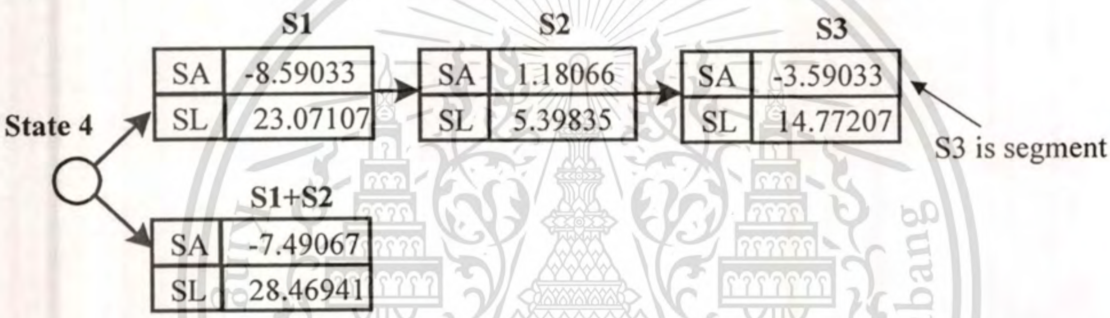


Fig 3.32 State 4 at segment number 3 (real segment).

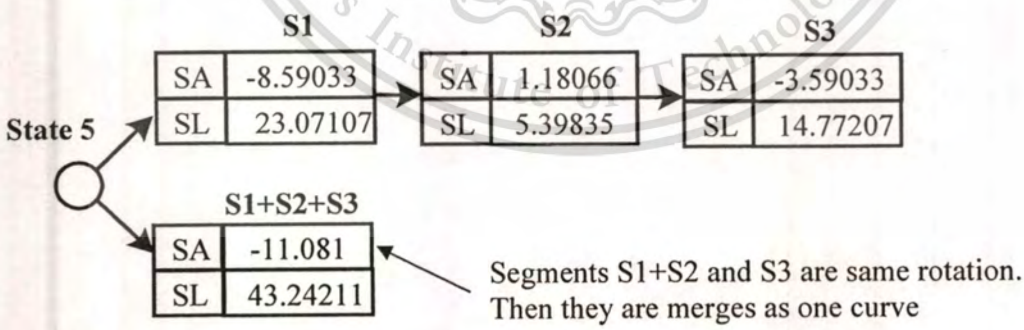


Fig 3.33 State 5 of tree generations at segment number 3 is real segment and merges with segment S1+S2+S3 that have the same rotation.

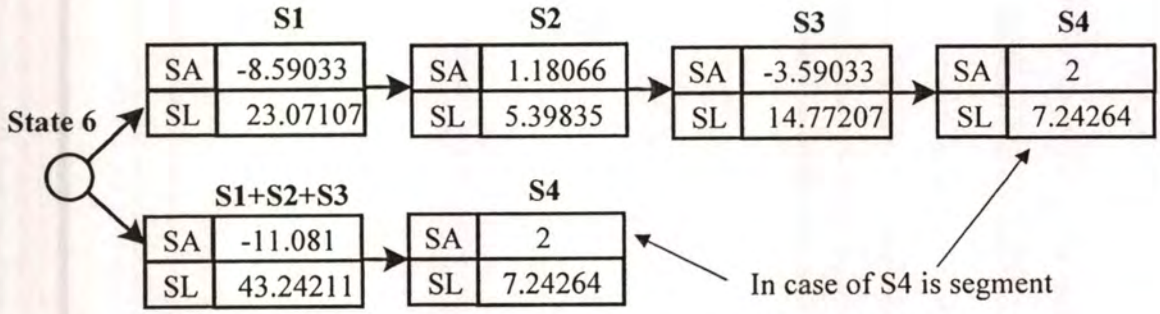


Fig 3.34 State 6 at segment number 4 (uncertain segment with real segment assumption).

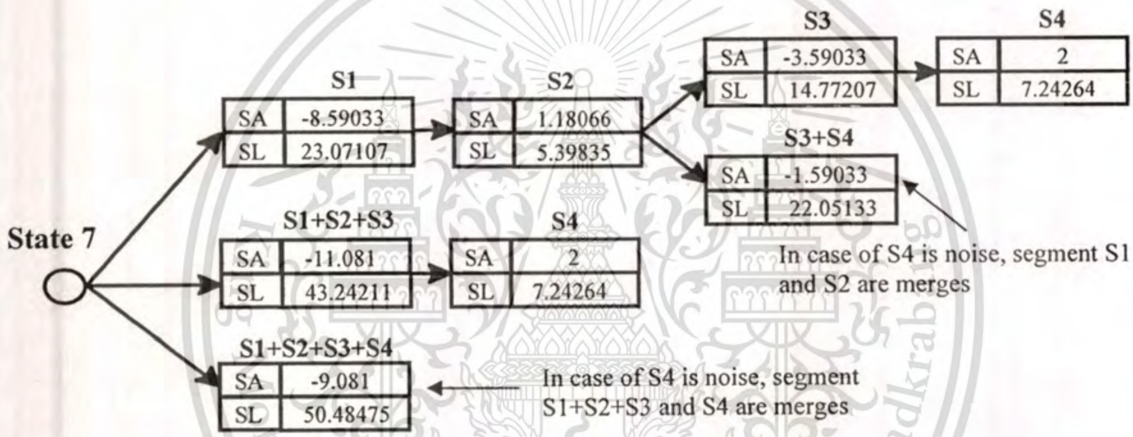


Fig 3.35 State7 at segment number 4 (uncertain segment with noise segment assumption).

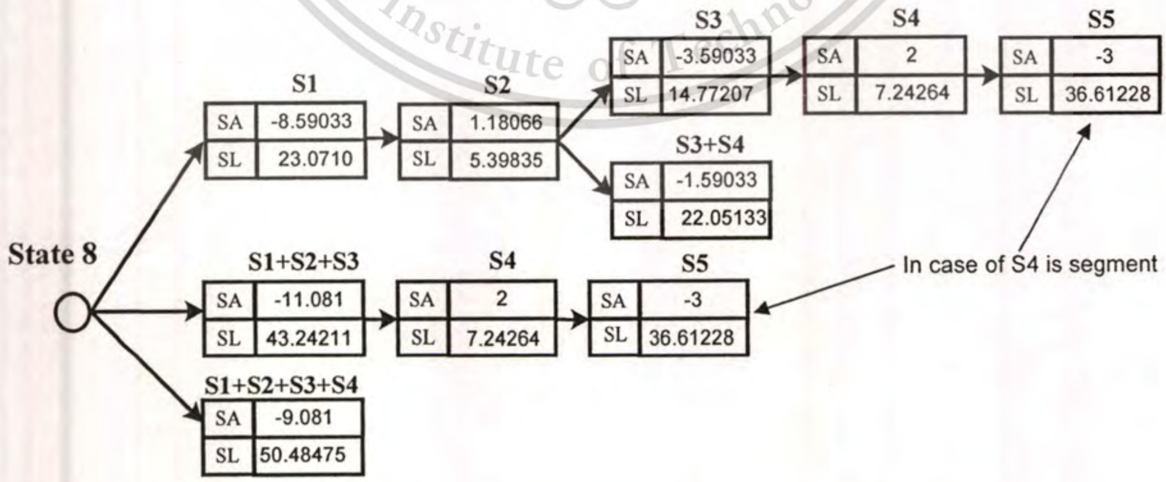


Fig 3.36 State 8 of tree generations at segment number 5 is real segment.

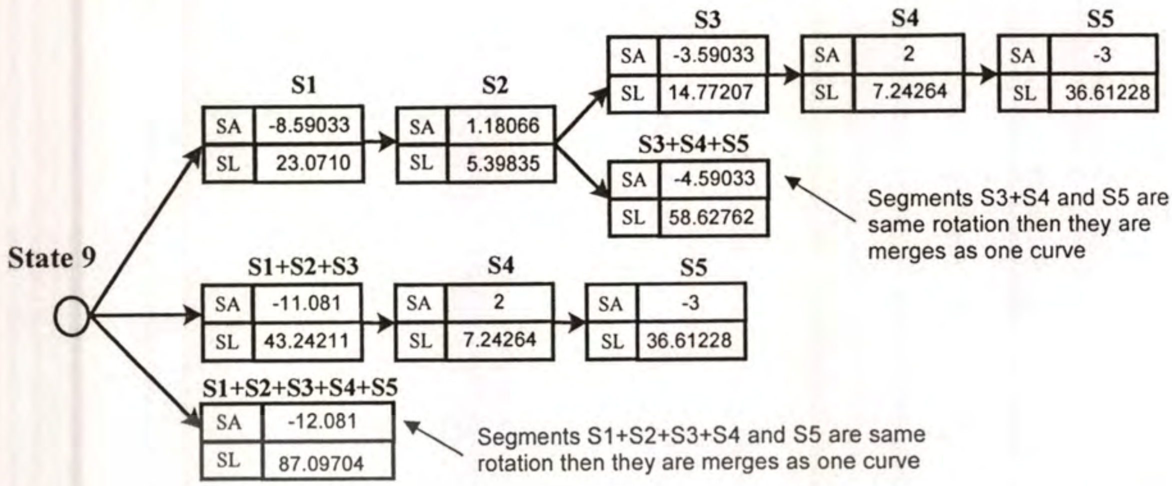


Fig 3.37 State 9 of tree generations at segment number 5 is real segment merges with segment S3+S4, and also merge with segment S1+S2+S3+S4, they have the same rotation.

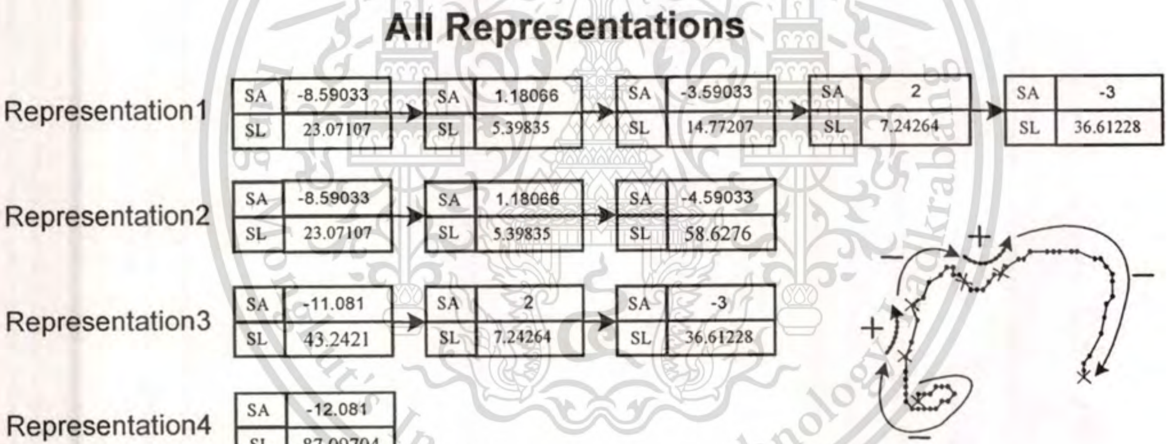


Fig 3.48 Shown all representation of Lao handwritten character "ᏣϽ"

From this it can be seen that the angle of a curve segment is rotation invariant and shape invariant. As shown in Fig 3.39, the directional code for the rotation of any counter clockwise 90 degree turn is 2.0 and the directional code for whole circle 270 turn is 6 independent of exact shape.

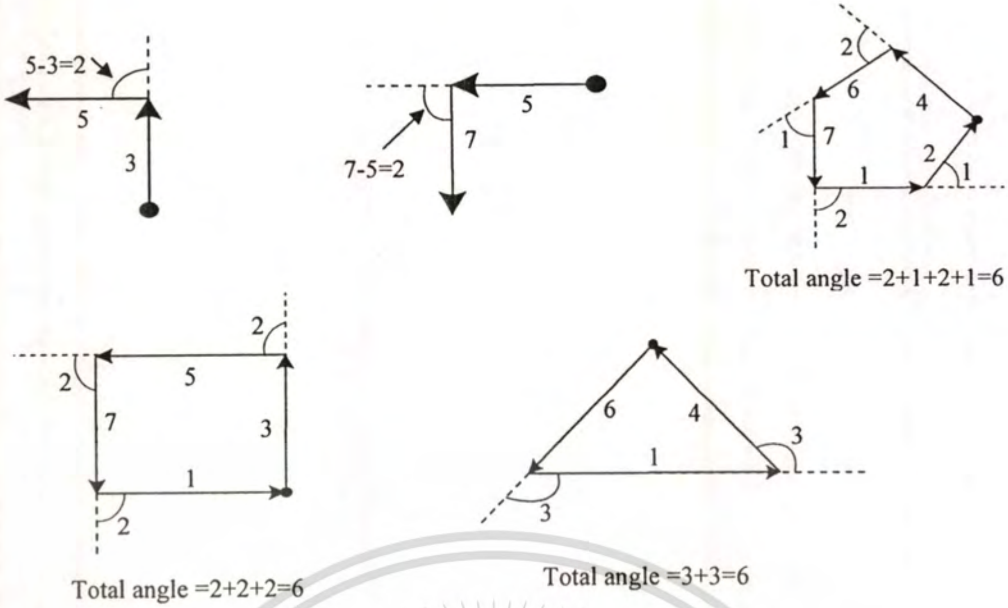


Fig 3.39 Shape invariant.

In the Fig 3.40, the two different curvatures with the same direction in clockwise or counter clockwise direction can not be distinguished, because the two curvatures have the same amount angle and length. Consequently, other features are added to help distinguishing them at the cost of rotation invariant reduction.

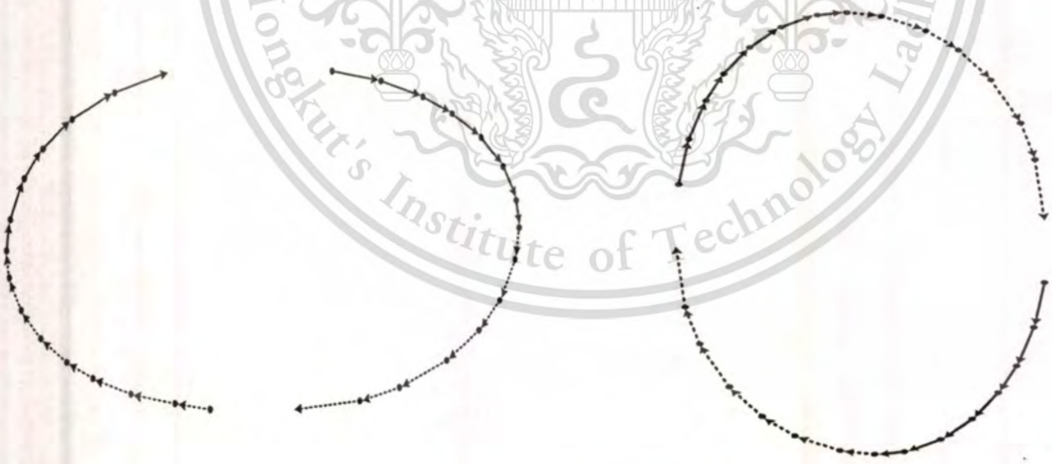


Fig 3.40 Rotation invariant.

The change in X of a segment is the summation of all the change in X of all the point in the segment. Fig 3.41 shows two Lao handwritten characters that can not be distinguished due to the above reason. Thus, the change in X and Y coordinates of pen trajectory are added to the angle and length features of a segment. The calculation of change in X and Y coordinates is similar to the calculation of angle. The change in X at a point i is defined as $X_{i+1} - X_i$. If a character written from left to right a sequence of x coordinate is increasing and results in positive value of the change in X. If character written from right to left a sequence of X coordinate is decreased and results in negative value of the change in X. The calculation of the change in Y is similar to X coordinate. After the calculation for all the points in a character pen tracing, all the adjacent values (of change in X) with the same sign in the sequence are added together until the sign change from + to - or - to + (sum value in same direction of writing from left to right, from right to left). This is analogous to segmentation using change in angle sign. But the segmentation points here are the point changing in X sign. From the segmentation points, all the changes in X the same segments are added together and form another sequence of segment called "segment in X", as shown in Fig 3.42 and Fig 3.43.

Fig 3.41 Show difference character but have the same single clockwise curve segment.

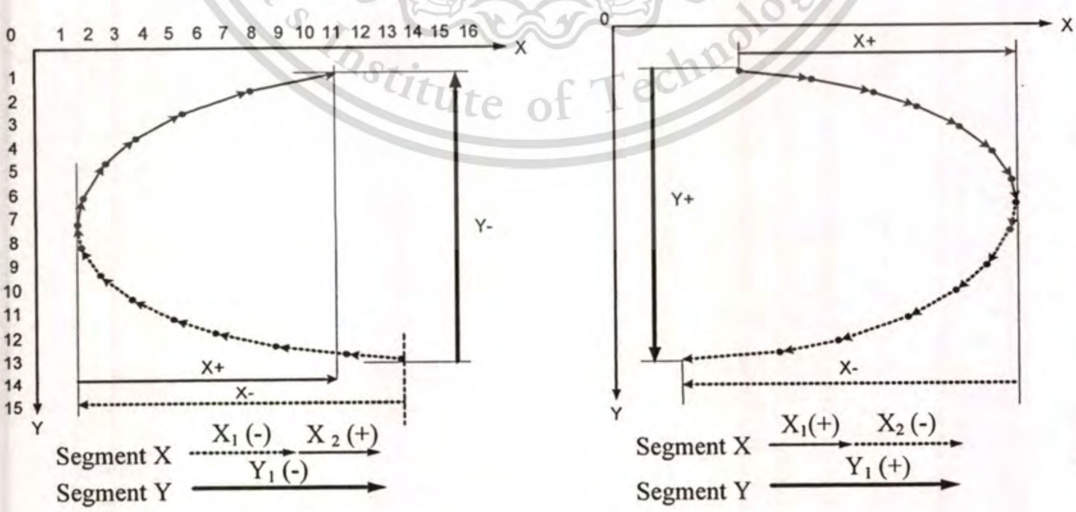


Fig 3.42 Segments in X and Y coordinate to assist easier separation of different curve with the same rotation.

Index [i]	X	Y	Difference X (*DX*)	Sign X	Segment X (*SX*)	SX (%)	Difference Y (*DY*)	Sign Y	Segment Y (*SY*)	SY (%)
0	349	361	0	+			-1	-		
1	349	360	1	+			0	-		
2	350	360	1	+			0	-	-2	-0.0408
3	351	360	1	+	6	0.1071	-1	-		
4	352	359	1	+			0	-		
5	353	359	1	+			0	-		
6	354	359	1	+			1	+		
7	355	360	-1	-			1	+		
8	354	361	-1	-			0	+		
9	353	361	-1	-			1	+		
10	352	362	-1	-			0	+	3	0.0612
11	351	362	-1	-			0	+		
12	350	362	-1	-	-8	-0.1428	0	+		
13	349	362	-1	-			0	+		
14	348	362	-1	-			-1	-		
15	347	361	0	-			-1	-		
16	347	360	0	-			-1	-		
17	347	359	0	-			-1	-		
18	347	358	0	-			-1	-		
19	347	357	0	-			-2	-		
20	347	355	1	+			-2	-		
21	348	353	1	+			-3	-	-19	-0.3877
22	349	350	-1	+			-2	-		
23	348	348	2	+			-1	-		
24	350	347	1	+			-2	-		
25	351	345	2	+			-1	-		
26	353	344	1	+			-1	-		
27	354	343	1	+			0	-		
28	355	343	0	+			1	+		
29	355	344	1	+			0	+		
30	356	344	1	+			1	+		
31	357	345	1	+			1	+	3	0.0612
32	358	346	1	+			0	+		
33	359	346	1	+			0	+		
34	360	346	1	+			-1	-		
35	361	345	1	+	35	0.6250	-1	-		
36	362	344	1	+			0	-		
37	363	344	1	+			-1	-		
38	364	343	2	+			-1	-		
39	366	342	1	+			0	-	-5	-0.1020
40	367	342	1	+			-1	-		
41	368	341	3	+			0	-		
42	371	341	1	+			0	-		
43	372	341	1	+			0	-		
44	373	341	2	+			0	-		
45	375	341	1	+			0	-		
46	376	341	3	+			1	+		
47	379	342	1	+			0	+		
48	380	342	1	+			1	+		
49	381	343	1	+			1	+		
50	382	344	0	+			1	+		
51	382	345	0	+			2	+	17	0.3469
52	382	347	-1	-			1	+		
53	381	348	-1	-			4	+		
54	380	352	-2	-			3	+		
55	378	355	-1	-	-5	-0.0892	1	+		
56	377	356	0	-			1	+		
57	377	357	0	-			1	+		
58	377	358	0	-			1	+		

Fig 3.43 Example calculation of segments in X and segments in Y.

In conclusion, the handwritten character is represented by alternative curvature tree and sequence of X and Y as follows and shown in Fig 3.44.

1. A tree of curvature in clockwise or counter-clockwise direction. A branch in a tree indicating the possibilities where uncertain curvature segment can either be a noise or a real curve. This curvature is calculated from summation of the drawing angle of all the points in a segment. A segment having angle in the same clockwise or counter-clockwise direction is formed from all points.
2. Sequence of segments in X coordinate is the total change in X coordinate. The change can be positive or negative depending on the drawing. If the drawing is from left to right the change is positive. The segmentation points is located at point where drawing change direction from left to right to from right to left.
3. Sequence of segments in Y coordinate is analogous that of segments in X.

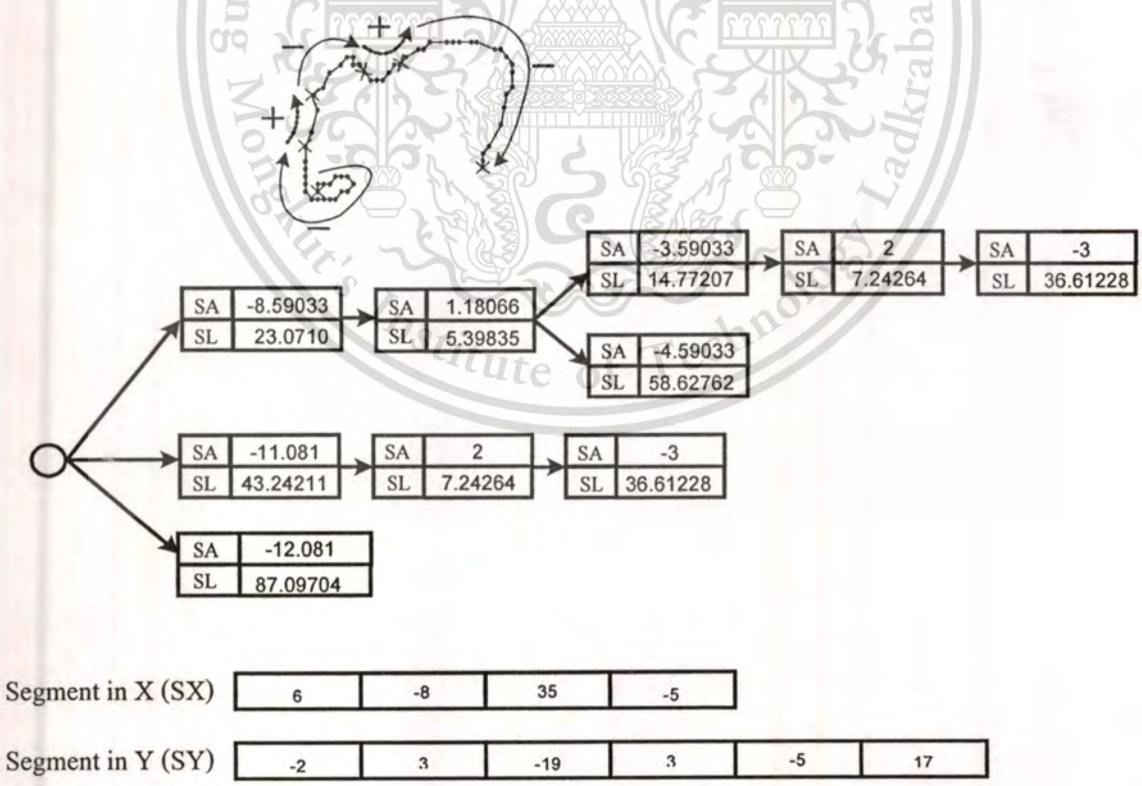


Fig 3.44 Example feature-vector of Lao handwritten character "ຕ".

Chapter 4

Handwritten Character Classification

4.1 Recognition process

The handwritten feature used in this dissertation is a tree of curvature segments. They are considering as featureless [41] since they can not exist on the same feature-vector space. Therefore, classification techniques that require feature space such as neural network and clustering can not be used. The choice of classifier is limited to prototypes matching only. Lao handwritten character recognition system is shown in the Fig 4.1. The main recognition method is 1-nn using prototype matching where each prototype has more than one feature-vector and with various numbers of dimensions.

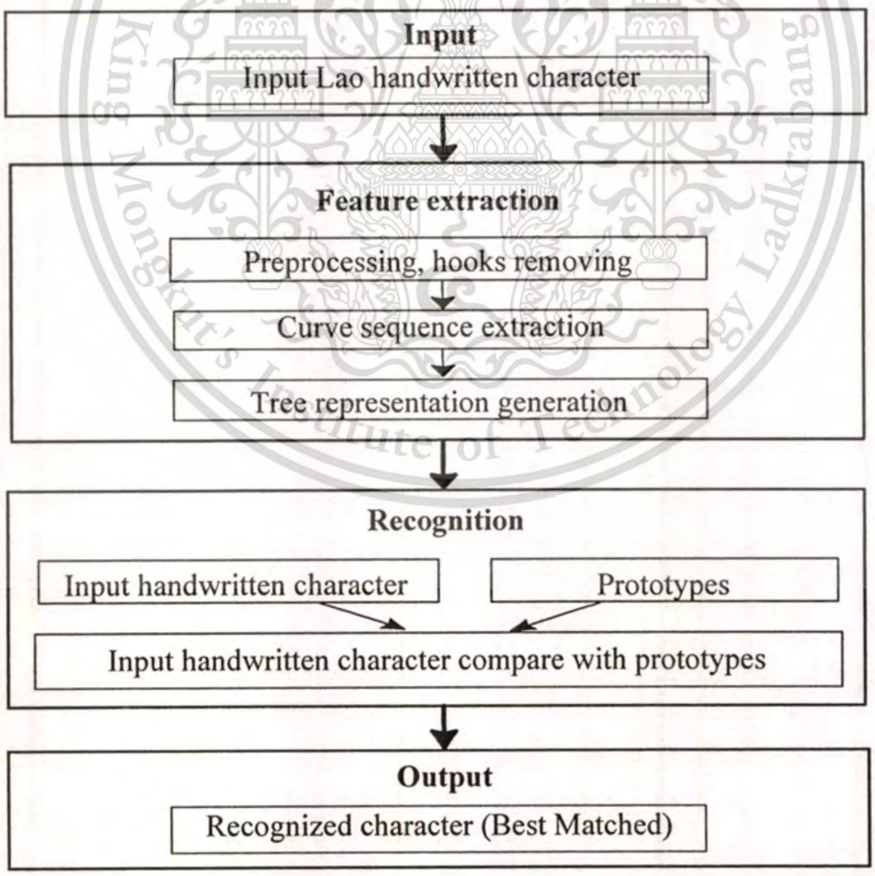


Fig 4.1 Recognition system

4.2 Data normalization

As mentioned in chapter 3, Lao handwritten character representation consists of:

1. Sequence of curvature segments is represented in tree format. In other words, multiple feature-vectors have multiple dimensions, where each segment is a single dimension and each sequence is a feature vector. Since there are multiple sequences in a tree, a character is represented as a multiple feature-vector. To be precise, segment is even not a dimension, each segment still consist of segment angle, segment length.
2. Two sequences of segment in X and Y coordinates are partitioned. The segment in X coordinates is partitioned at a point where the value in X coordinates changes the direction from left-right to right-left. Each segment is a summation of all the X coordinate displacements between adjacent points for all the points in a segment. The Y segment is similar except the direction change from up-down to down-up.

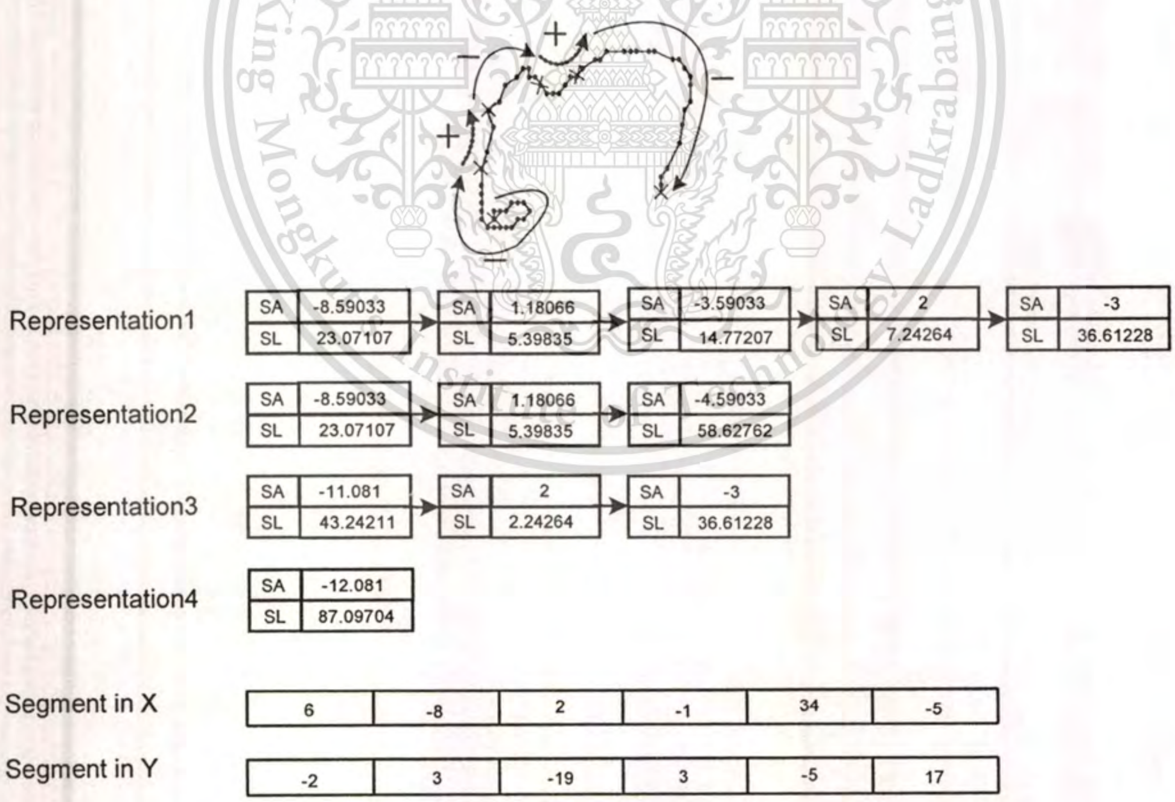


Fig 4.2 Feature of Lao handwritten character "ບ"

Each dimension of a Character feature-vector has to be normalized before further used in classifications. The normalization is performed in angle and length of a segment.

The angle of each segment has a lot of variations from very small angle less than 1 unit to a largest one of 48 units (multiple loops in a head of character). As mentioned in chapter 3, angle unit is obtained from difference of DC at the sampling point, where 1 unit equal 45 degree. The angle also has sign, plus for counter-clockwise and minus for clockwise. The angle is further applied log scale to emphasis difference in small angle less than π more than distance in large angle (more than 2π). The angle is normalized by the largest angle encounter, 48 units. The segment angle is normalized as in equation 4.1.

$$NSA_i = \frac{Log\ 10\left(\left|SA_i\right| + 1\right)}{Log\ 10\,(48)}; \quad i = 0; \quad i \le K \tag{4.1}$$

where K is segment number and NSA_i is the normalization segment angle.

After this normalization, there is not much different for angle in the "head" segment between the two characters in Fig 4.3 since the angle is in log scale.

Fig 4.3 The two "head" angle are about the same size after normalization even with multiple loops.

The segment length is normalized to summation of all the segment length of a character. This is applied to segment length of the clockwise and counter-clockwise segment as shown in equation (4.2). The normalization length of segment in X and Y coordinates is similar to equation (4.2) except that the segment is in X and Y coordinates. Fig 4.4 shows the feature-vector of the same character of Fig 4.2 after normalization.

$$NSL_i = \frac{|SL_i|}{TSL}; \qquad where \quad TSL = \sum_{j=0}^{N-1} |L_j|; \quad i=0; \quad i \le K; \quad j=0 \tag{4.2}$$

NSL_i is normalized segment length of segment i^{th}.

TSL is Total Segment Length (Summation of all segment lengths of a character).

N is total number of sampling point of a character.

K is Number of segments.

Representation1

SA	-0.58399	SA	0.20139	SA	-0.39366	SA	0.28379	SA	-0.35810
SL	0.26489	SL	0.06198	SL	0.16961	SL	0.08316	SL	0.42036

Representation2

SA	-0.58399	SA	0.20139	SA	-0.44458
SL	0.26489	SL	0.06198	SL	0.67313

Representation3

SA	-0.64190	SA	0.28379	SA	-0.35810
SL	0.49648	SL	0.08316	SL	0.42036

Representation4

SA	-0.66257
SL	- 1

Segment in X

0.1111	-0.1481	0.6481	-0.0925

Segment in Y

-0.0408	0.0612	-0.3878	0.0612	-0.102	0.3469

Fig 4.4 Data of character after normalization.

4.3 Dissimilarity of characters

In the proposed handwritten representation, a character may be represented as one or more sequences of curvatures. The exact number of representations depends on the number of uncertainty segments. In order to find dissimilarity between two handwritten characters, all representations of both prototype and test character are comparable; they will compare and select the closest match as the distance between the two characters. Sequences of curvatures are comparable if the numbers of segments in both sequences are the same and the segments have the same rotations (clockwise or counter-clockwise). The example of dissimilarity calculation is shown in Fig 4.5, where the closest match is selected from all comparable segments.

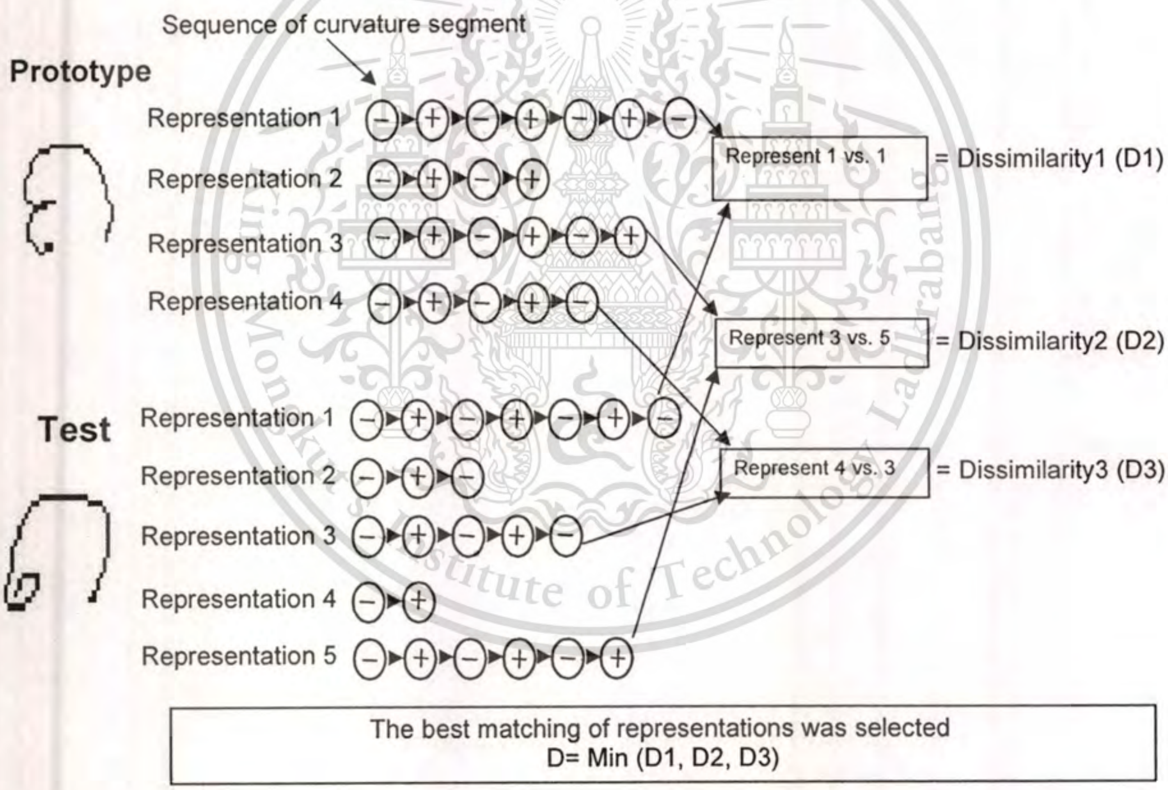


Fig 4.5 Example of similarity calculation between two characters.

Dissimilarity of representation is calculated below:

1. Dissimilarity between two sequences of curvature segments is calculated as the average of dissimilarity of each segment in the sequence. The segments in the two sequences are compared

in order from the first to the last one. Each segment consists of two attributes, segment angle and segment length. The details of dissimilarity calculation are as follows:

1: if the number of segments of the two sequences is different, those segments are incomparable (distance is 1).

2: if first segment of the two sequences is opposite rotation, they are incomparable.

3: if any corresponding segments in the two sequences are not overlap, they are incomparable. Segments are not overlaped if their relative positions with in a character are shifted over 12% of the length summation of all the segment of a character. The examples involve segment overlap are shown in the Fig. 4.6.

All segments in Fig. 4.6 (A) are overlapped. In (B) S1 and S3 are overlapped but S2 is not overlapped at all. If the part not overlapped is less than 12% of the total length, the dissimilarity of that segment (S2) is 1. But if the unoverlapped part is larger than 12% the two sequences are incomparable, as shown in Fig4.6(C).

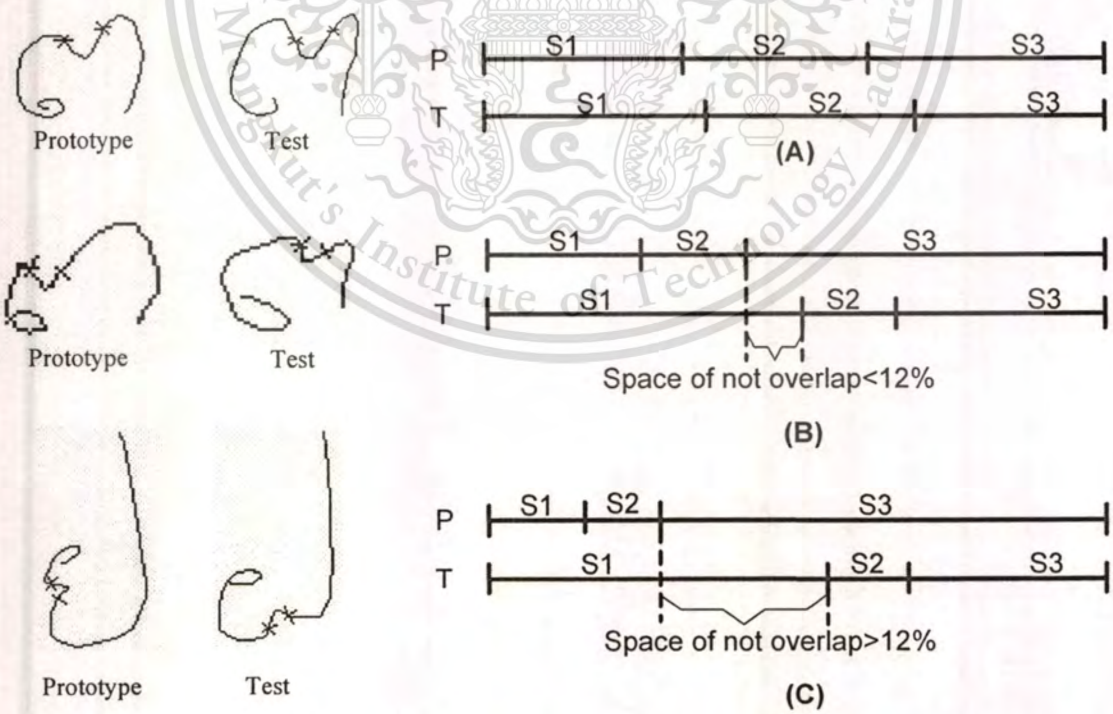


Fig 4.6 Example of segment overlap.

Dissimilarity of segment angle is calculated as in equation (4.3).

$$DA_i = \left(NSA_{P_i} - NSA_{T_i}\right)^2 \tag{4.3}$$

Where: DA_i is Dissimilarity of Segment Angle of curve segment i^{th} in the sequence.

NSA_{Pi} is Normalize of Segment Angle of prototype character at segment i^{th}

NSA_{Ti} is Normalize of Segment Angle of unknown character at segment i^{th}

Dissimilarity of segment length can be calculated as equation (4.4) below.

$$DL_i = \begin{cases} \left(NSL_{P_i} - NSL_{T_i}\right)^2; & if \quad Length \quad is \quad overlap \\ 1; \quad if \quad length \quad isn't \quad overlap & < \quad 12\% \end{cases} \tag{4.4}$$

Where: DL_i is Dissimilarity of Segment Length of curve segment i^{th} in the sequence.

NSL_{Pi} is Normalized Segment Length of prototype character at segment i^{th}

NSL_{Ti} is Normalized Segment Length of unknown character at segment i^{th}

Dissimilarity of a sequence of curve segment is calculated as average of dissimilarity of each segment in the sequence as equation (4.5) below.

$$DSCS = \frac{\sum_{i=0}^{i<K} \sqrt{\left(DA_i + DL_i\right)}}{K}; \quad i = 0; \quad i < K \tag{4.5}$$

Where: $DSCS$ is Dissimilarity Sequence of Curve Segment.

K is number of segment.

2. Dissimilarity of sequent of segment in X and Y is calculated differently from sequence of segment in angle. Since the sequence in angle has multiple representations to reflect uncertainty, the sequence is incomparable if segment has different number of segment. On the other hand

sequence of segment in X and Y has single representation so we allow more flexibility in comparison those sequence as follows:

If both sequence have the same number of segments and segment has the same direction (left to right or right to left) as shown in Fig 4.7, the dissimilarity of both sequences is the average dissimilarity of each segment in the two sequences as in equation (4.6).

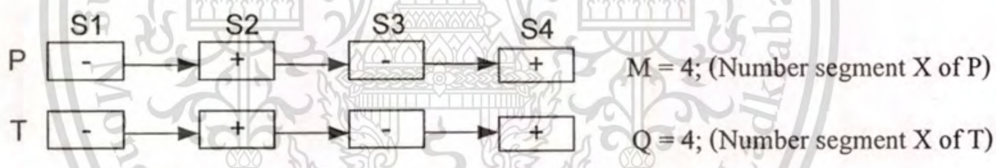
$$DSX = \frac{\sum_{i=j=0}^{i,j<K} \sqrt{\left(NSDX_{P_i} - NSDX_{T_j}\right)^2}}{K}; \quad i=j=0; \quad i,j<K \tag{4.6}$$

Where: *DSX* is Dissimilarity of sequence X in case direction is the same and number segment is the same.

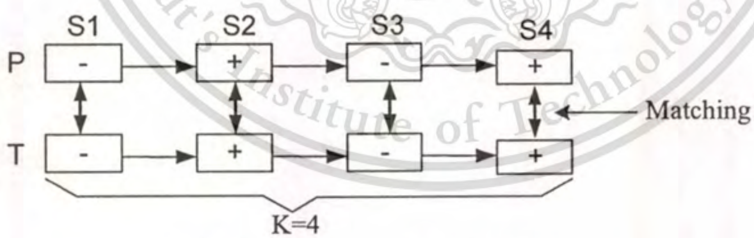
$NSDX_{Pi}$ is Normalize of Sum Difference X of prototype at segment i^{th}.

$NSDX_{i}$ is Normalize of Sum Difference X of unknown at segment j^{th}.

K is number of segment.



(A) Representation of prototype and unknown



(B) Matching of prototype and unknown

Fig 4.7 Calculation of dissimilarity of sequence of segment in X, with identical number segments and direction.

If both sequences have different number of segments as shown in Fig 4.8, to 4.10, the dissimilarity of both sequence is the average dissimilarity of each segment as in equation 4.6 except that all the shifted segments have value = 0. A sequence is shifted to align with the other sequence in both directions and maximum number of segment. If there are more than one possible shifted alignment between the two compared sequences, the best matched sequent is chosen.

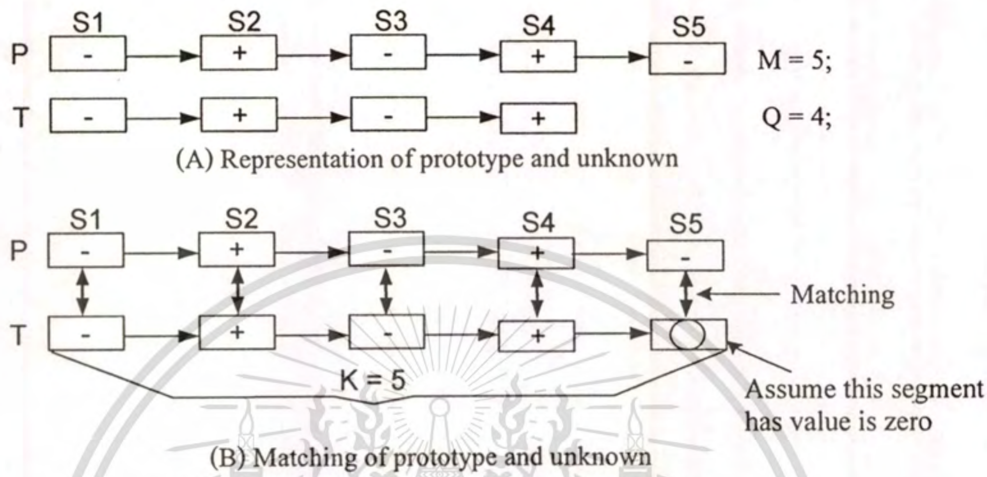


Fig 4.8 Calculation of dissimilarity of sequence of segment in X, with different number of segments but identical start direction.

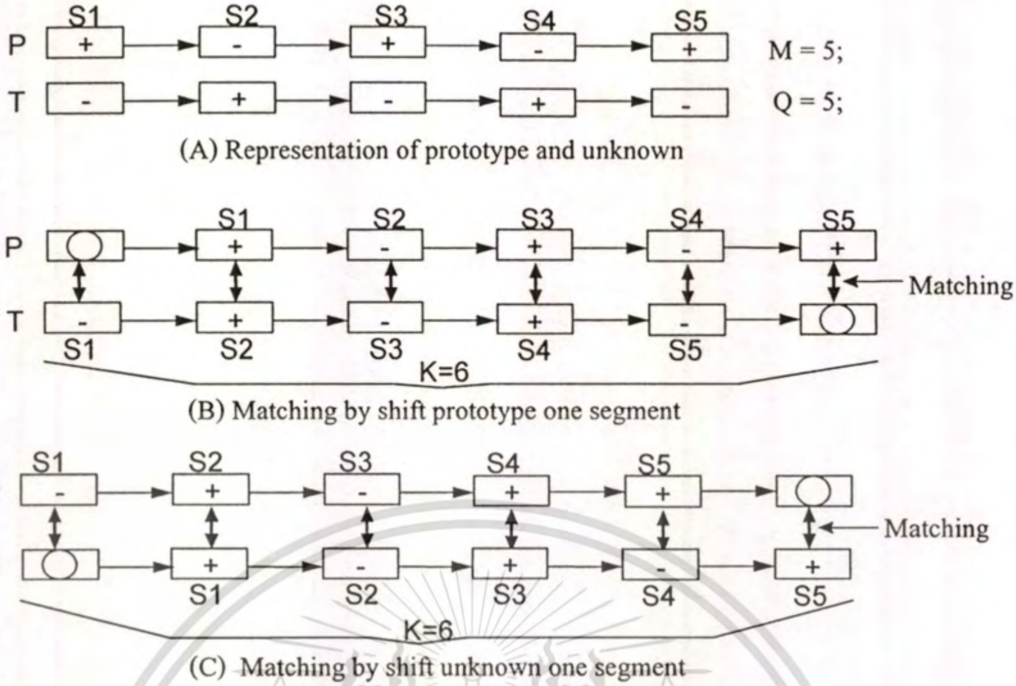


Fig 4.9 Calculation of dissimilarity sequence of segment in X, with different direction.

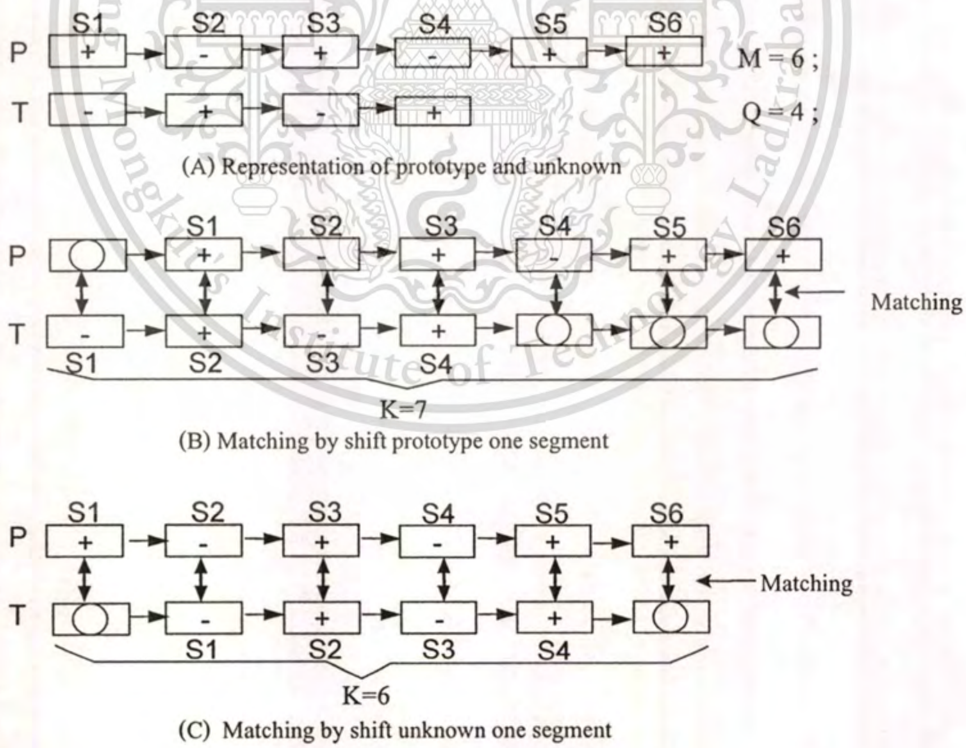


Fig 4.10 Calculation dissimilarity sequence of segment in X, with numbers of segment and direction are difference.

The dissimilarity between two representations is calculated as in equation (4.7) below:

$$DRP = \frac{(DSCS + DSX + DSY)}{3} \tag{4.7}$$

Where: DRP is Dissimilarity of representation. If there are more than one representations of a compared character, the minimum DRP (best match) is selected for the dissimilarity of the two characters.

4.4 Prototype classification

As previously mentioned, the classifier used in this work is the prototype matching. The unknown character is compared with all prototypes and the classification result is the character of the best matched prototype as shown in Fig 4.11.

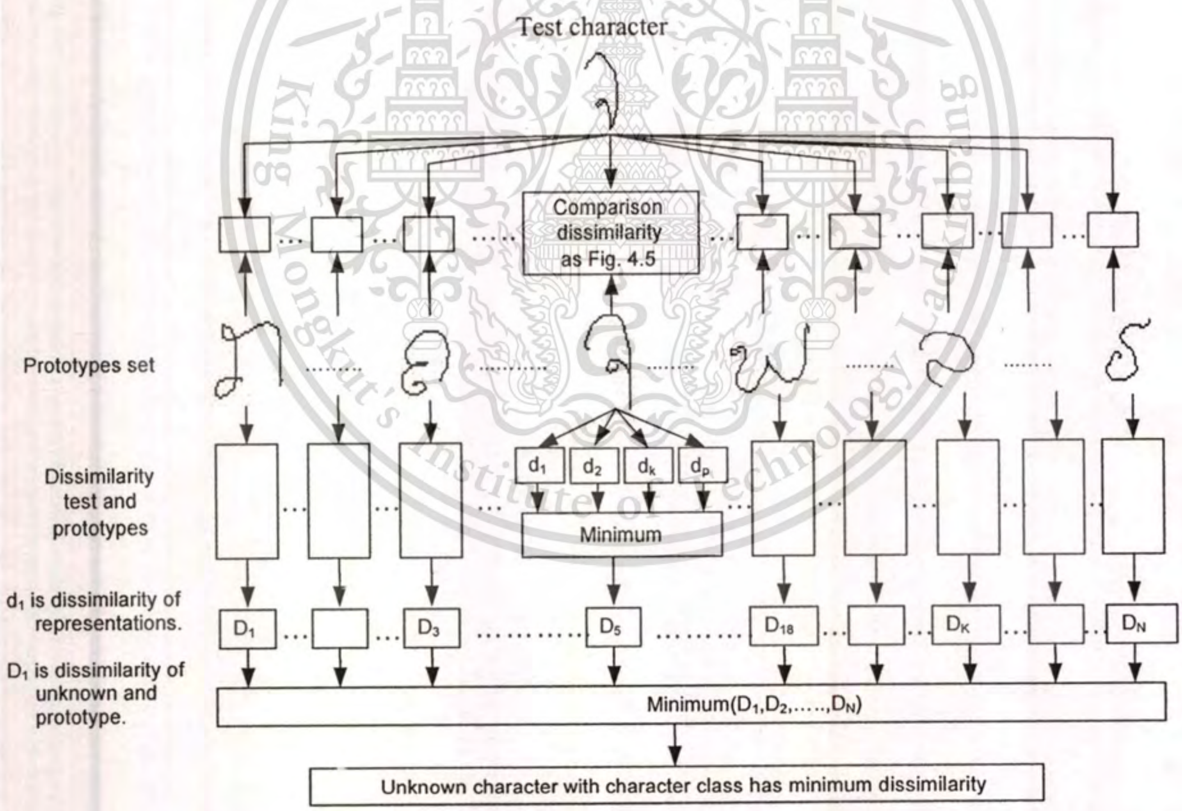


Fig 4.11 Prototype matching of Lao handwritten character recognition.

Chapter 5

Selected Related Methods

In this chapter the selected methods of handwritten recognition used in comparison to the proposed multiple representation method are described. They are elastic matching, single representation, and dominant point. The single representation is similar to the proposed method without uncertain curvature segment. The dominant point is similar to single representation with variation in segmentation point, where dominant point is used in segmentation. Finally the prototype selection method maximum cover first is presented. This prototype selection is used for the all the presented feature.

5.1 Elastic matching

Elastic matching allows comparing two sequences with difference of lengths. Input of Lao handwritten character represented as sequence of coordinate X, Y. Since a character may be written in different speed and in different size, further normalization is performed as follows:

5.1.1 Data normalization.

A handwritten character size is normalized as followed:

$$X_i = \frac{(X_i - Min(X_i))}{SX} \qquad (5.1)$$
$$SX = (Max(X_i) - Min(X_i))$$

Where $Max(X_i)$ is maximum value of X.

$Min(X_i)$ is minimum value of X.

$$Y_i = \frac{(Y_i - Min(Y_i))}{SY} \qquad (5.2)$$
$$SY = (Max(Y_i) - Min(Y_i))$$

Where $Max(Y_i)$ is maximum value of Y.

$Min(Y_i)$ is minimum value of Y.

After size normalization a handwritten character position is further normalized as follows:

$$AX = \frac{\sum_{i=0}^{N} X_i}{N} \tag{5.3}$$
$$X_i = X_i - AX$$

Where AX is average X value of character in X coordinate.

N is number of sampling points.

$$AY = \frac{\sum_{i=0}^{N} Y_i}{N} \tag{5.4}$$
$$Y_i = Y_i - AY$$

Where AY is average Y value of character in Y coordinate.

N is number of sampling points.

5.1.2 Dissimilarity by elastic matching

In order to find dissimilarity of characters by elastic matching, it is assumed here that characters are already normalized. And dissimilarity measure is given as Euclidean distance.

To compare two characters A and B, Dynamic Time Warping (DTW) is used. They are represented as sequences of point ($A=a_1,a_2,\ldots,a_m$) and ($B=b_1,b_2,\ldots,b_n$) with lengths m and n respectively. First, construct an m×n matrix, whose element $(i,j)^{th}$ contains dissimilarity cost measure between the two points a_i and b_j. Each matrix element (i,j) corresponds to the alignment between the points a_i and b_j. Once this matrix is created, an optimal warping path W is selected which is contiguous set of matrix an element that defines a mapping between A and B. The k^{th} element of W is defined as $w_k=(i,j)_k$, where $W=w_1,w_2,\ldots,w_k,\ldots,w_K$ and K is the length of warping path. The DTW distance measure between two sequences A and B is given by equation (5.5).

$$D(A,B) = \min_{w} \left[\frac{\sum_{k=1}^{K} w_k}{K} \right] \tag{5.5}$$

Where, W is a set of all possible warping paths under given constraints. The minimization in equation 4.26 is carried out using dynamic programming techniques. The following recurrence relation is used to find the DTW dissimilarity between two sequences as equation (5.6).

$$r(i,j) = d(i,j) + \min\begin{cases} r(i-1,j) \\ r(i,j-1) \\ r(i-1,j-1) \end{cases} \tag{5.6}$$

where $r(i,j)$ is cumulative distance up to the current element. $d(i,j)$ is cost measure of distance between i^{th} and j^{th} point of the two sequences calculate as equation (5.7).

$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{5.7}$$

The warping path is subjected to several constraints such as, boundary conditions, continuity as below and other in [9], [22].

$$\begin{aligned} w_1 &= (a_1, b_1) \\ w_K &= (a_m, b_n) \end{aligned} \tag{5.8}$$

Boundary condition of DTW is the first point and last point of sequence A and B which are match first. Then, all data points are matched at least once and several data points can matched to one point [9]. Fig 5.1 and Fig 5.2 below show the warping path and matching points of sequence A and B.

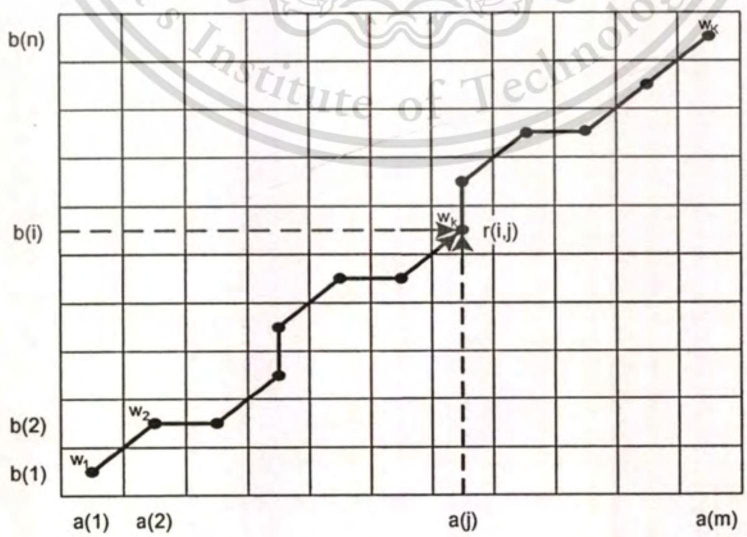


Fig 5.1 Illustration of warping path of sequence A and B

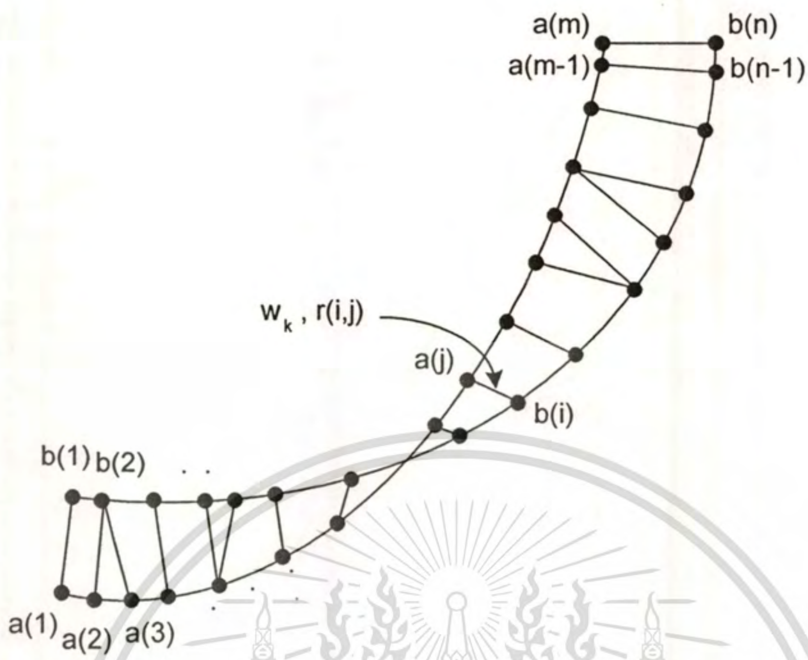


Fig 5.2 Point to point matching of sequence A and B

Lao handwritten character recognition uses elastic matching as shown in the Fig 5.3.

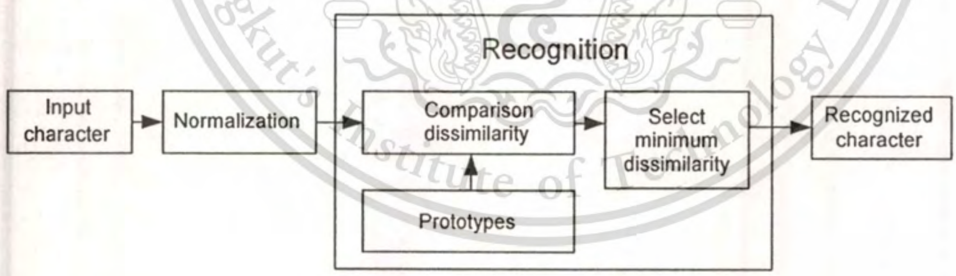
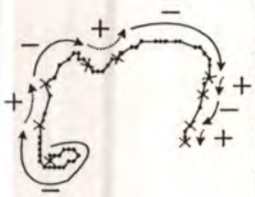


Fig 5.3 Recognition system of elastic matching.

5.2 Single representation

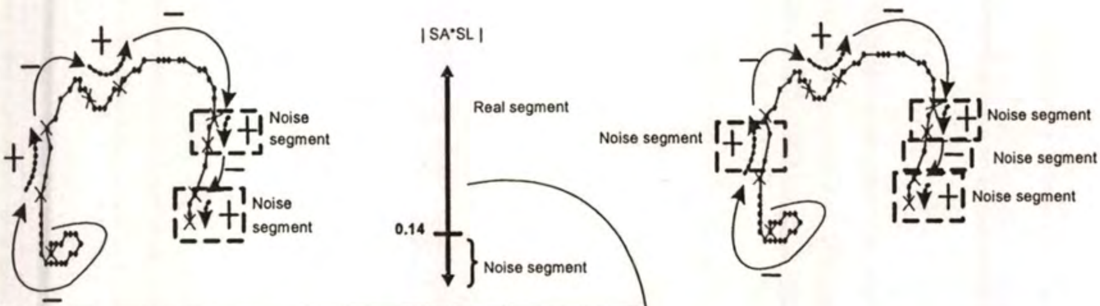
In comparing to the proposed multiple representations, single representation is the conventional version where there is no uncertainty. In this method, the curvature segment is classified into only two classes, noise and real curvature. Hence, there is only single representation of sequent of curvature as shown in Fig 5.4. This method also use segment X and segment Y feature of the multiple representation version as shown in the Fig 3.43 in chapter 3.

The curvature segment of this method may still have a lot of short small angle curves. These very small curve segments are not irrelevant to the writing of a character and may cause error in the recognition. It is like a noise curve segment not a real curve segment of a handwritten character. In our previous research [38], small curve segments are considered as noise if it is smaller than some threshold as shown in Fig 5.5. After all noises have been removed, a handwritten character can be represented by just a single representation of curve segment sequence. The performance of this representation depends on the correctness of this small curvatures noise identification, which directly depends on the threshold selection. A variation in threshold causes variation in number of segment of the same character. If the threshold is small, there can be more segments due to inclusion of noisy segments. On the other hand, if the threshold is large, some small real curvature segments will be missing as shown in the Fig 5.6. To obtain the optimum threshold, the recognitions of test handwritten data of various thresholds are performed and the threshold with the highest recognition rate is selected. From the experiment by using prototype matching, the recognition of this feature is about 96.19 %. The main error of this method is due to the variation of the curve segments, which cause recognition error.



Index [i]	X	Y	Directional code	Length (*L*)	Angle (*A*)	Sign	Sum Angle (*SA*)	Sum Length (*SL*)	Segment number
0	**349**	**361**	**3.000**	**1.000**	**-2.000**	-			
1	349	360	1.000	1.000	0.000	-			
2	350	360	1.000	1.000	1.000	-			
3	351	360	2.000	1.414	-1.000	-			
4	352	359	1.000	1.000	0.000	-			
5	353	359	1.000	1.000	-1.000	-			
6	354	359	8.000	1.414	-2.000	-			
7	355	360	6.000	1.414	-1.000	-			
8	354	361	5.000	1.000	1.000	-			
9	353	361	6.000	1.414	-1.000	-			
10	352	362	5.000	1.000	0.000	-	-8.59033	23.07107	S1
11	351	362	5.000	1.000	0.000	-			
12	350	362	5.000	1.000	0.000	-			
13	349	362	5.000	1.000	-1.000	-			
14	348	362	4.000	1.414	-1.000	-			
15	347	361	3.000	1.000	0.000	-			
16	347	360	3.000	1.000	0.000	-			
17	347	359	3.000	1.000	0.000	-			
18	347	358	3.000	1.000	0.000	-			
19	347	357	3.000	2.000	-0.590	-			
20	**347**	**355**	**2.410**	**2.236**	**0.181**	+	1.18066	5.39835	S2
21	348	353	2.590	3.162	1.000	+			
22	**349**	**350**	**3.590**	**2.236**	**-2.000**	-			
23	348	348	1.590	2.236	0.819	-			
24	350	347	2.410	2.236	-0.819	-			
25	351	345	1.590	2.236	0.410	-			
26	353	344	2.000	1.414	-1.000	-	-3.59033	14.7727	S3
27	354	343	1.000	1.000	-2.000	-			
28	355	343	7.000	1.000	2.000	-			
29	355	344	1.000	1.000	-1.000	-			
30	356	344	8.000	1.414	0.000	-			
31	**357**	**345**	**8.000**	**1.414**	**1.000**	+			
32	358	346	1.000	1.000	0.000	+			
33	359	346	1.000	1.000	1.000	+	2	7.24264	S4
34	360	346	2.000	1.414	0.000	+			
35	361	345	2.000	1.414	-1.000	+			
36	362	344	1.000	1.000	1.000	+			
37	**363**	**344**	**2.000**	**1.414**	**-0.410**	-			
38	364	343	1.590	2.236	-0.590	-			
39	366	342	1.000	1.000	1.000	-			
40	367	342	2.000	1.414	-1.000	-			
41	368	341	1.000	3.000	0.000	-			
42	371	341	1.000	1.000	0.000	-			
43	372	341	1.000	1.000	0.000	-			
44	373	341	1.000	2.000	0.000	-	-4	24.0552	S5
45	375	341	1.000	1.000	-0.410	-			
46	376	341	8.590	3.162	0.410	-			
47	379	342	1.000	1.000	-1.000	-			
48	380	342	8.000	1.414	0.000	-			
49	381	343	8.000	1.414	-1.000	-			
50	382	344	7.000	1.000	0.000	-			
51	382	345	7.000	2.000	-1.000	-			
52	**382**	**347**	**6.000**	**1.414**	**0.688**	+	0.06880	1.41421	S6
53	**381**	**348**	**6.688**	**4.123**	**-0.437**	-	-0.68808	7.72866	S7
54	380	352	6.251	3.606	-0.251	-			
55	**378**	**355**	**6.000**	**1.414**	**1.000**	+			
56	377	356	7.000	1.000	0.000	+	1	3.41421	S8
57	377	357	7.000	1.000		+			
58	**377**	**358**				+			

Fig 5. 4 Sequence of curvature segments of Lao handwritten character "ໆ".



R = Real segment, N = Noise segment								
Segment number	S1	S2	S3	S4	S5	S6	S7	S8
Sum Angle (SA)	8.59033	1.18066	3.59033	2	4	0.06880	0.06880	1
Sum Length % (SL)	0.26489	0.06198	0.16961	0.0831	0.2761	0.01624	0.08874	0.03920
| SA*SL |	2.2754	0.07318	0.6089	0.1663	1.1047	0.01117	0.06106	0.03920
Segment status	R	N	R	R	R	N	N	N

R = Real segment, N = Noise segment			
Segment number	S1	S2	S3
Sum Angle (SA)	11	2	3
Sum Length % (SL)	0.49648	0.08316	0.42036
| SA*SL |	5.46128	0.16631	1.26108
Segment status	R	R	R

Fig 5.5 Example of noise segments is removed.

Fig 5.6 The effect when threshold is too big, small curves are miss interpret as noises.

In some character from some writer, the angle is too small and should be considered as a noise. But the same angle on the other writer or character should be considered as small intentional curve.

The data normalization and the dissimilarity calculation are similar to the multiple representation method as equations (4.3) to (4.7) in the chapter 4.

5.3 Dominant point

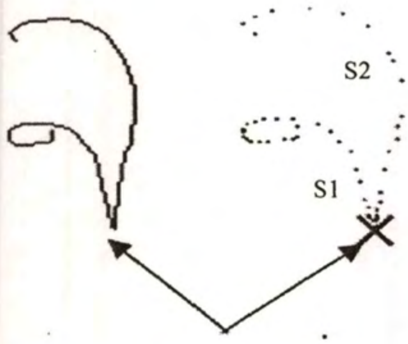
This method divided handwritten character in to segments by using dominant point or point where the pen trajectory turns the direction quickly as a segmentation point [31], as shown in the Fig 5.7. These points are always have big turn, then their angle value is big. These points can be found by monitoring of the angle value. If found the angle with the value bigger than or equal to the threshold value, so that point is segmentation as shown in the Fig 5.8. After get the segmentation point all the angle of all the points in a segment are summation together, also lengths in a segment are summation together as shown in the Fig 5.9. From the experiment, segmentation by using retrace point has a lot of variations in number of segments. The same character written in different style gets different number of segmentation points. The position of segmentation points also varies. The segmentation point is very sensitive to the threshold angle selection of retrace point. If the threshold value is smaller, the number of segmentation points will be more as shown in Fig 5.10 (A). Conversely, if it is too large, there may be no segmentation points as shown in the Fig 5.10(B). The recognition rate of this method for Lao handwritten character by using prototype matching is about 89.44%. From the experiment, this segmentation method is not suitable for Lao character.

The dissimilarity of this method is calculated as in equations (4.3) to (4.7) in the chapter 4.

Fig 5.7 Segmentation points of character using retrace point.

Index [i]	Direction	Angle	Sum Angle	Segment
0	7.000000	-1.000000	-8	S1
1	6.000000	-1.000000		
2	5.000000	0.311916		
3	5.311916	-0.311916		
4	5.000000	0.000000		
5	5.000000	-1.000000		
6	4.000000	0.000000		
7	4.000000	-1.000000		
8	3.000000	0.000000		
9	3.000000	-0.590335		
10	2.409665	-0.819331		
11	1.590334	0.000000		
12	1.590334	-0.590334		
13	1.000000	0.000000		
14	1.000000	-0.409665		
15	8.590335	-0.590335		
16	8.000000	-0.409665		
17	7.590334	0.228997		
18	7.819331	-0.638662		
19	7.180669	0.070662		
20	7.251331	-0.070662		
21	7.180669	0.228997		
22	7.409666	-0.409666		
23	7.000000	0.000000		
24	7.000000	1.000000		
25	8.000000	-1.000000		
26	7.000000	1.000000		
27	**8.000000**	**3.000000**	6.59033	S2
28	3.000000	0.000000		
29	3.000000	0.000000		
30	3.000000	0.000000		
31	3.000000	0.000000		
32	3.000000	0.000000		
33	3.000000	-0.409666		
34	2.590334	0.409666		
35	3.000000	-0.140894		
36	2.859106	-0.039775		
37	2.819331	-0.303807		
38	2.515524	0.484476		
39	3.000000	-0.311916		
40	2.688084	0.311916		
41	3.000000	0.311916		
42	3.311916	0.097749		
43	3.409665	0.133534		
44	3.543199	0.572233		
45	4.115432	0.572652		
46	4.688084	0.45281		
47	5.140893	0.449441		

Fig 5.8 Locate segmentation points by monitoring angle value in the sequence.



Segment	1	2
Sum Angle	-8	6.59033
Sum Length	80.816	115.228

Fig 5.9 Example segments of character.

Fig 5.10 (A) The threshold value is too small. (B) The threshold value is too big.

5.4 Prototype selection

The prototype selection method is maximum cover first [40]. It is described in the following and shown in the Fig 5.11.

1. The distance list for each data to all the data point is calculated and sorted from closest to furthest. This list is used to identify the candidate prototypes. For example, from distance list A3: A3 B1 B2 A2 A1, A3 is a reference point. The closest to A3 is A3 followed by B1, B2, A2 and A1. This suggests that only A3 itself must be used as a prototype in order to correctly recognize A3, since A3 is the only A closer to A3 (more similar) than B1. Otherwise, A3 may be misclassified as B1 if B1 is selected to be one of a prototype.
2. The cover list for each data point is constructed from the distance lists (cover list 1). If A3 is selected, it can also be used as a prototype to correctly classify A1 and A2. Since, A1 is closer to A3 than A1 to B1 (the closest other class of A1). A2 is also closer to A3 than A2 to B1. Hence, A1, A2, and A3 are all cover by A3. Another word, if A3 is selected as a prototype, A1, A2, and A3 will be the closest prototype than any other non A prototypes.
3. Select prototype with the maximum number of data points in the cover list (list A3 cover 3 data points A1, A2, and A3).
4. Remove data points (A1, A2, and A3) in the selected prototype cover list (list A3) from the entire cover list. The cover list after removing prototype A1, A2 and A3 is shown in cover list 2. Since A3 is already selected to be one of a prototype, data points A1, A2, A3 can be classified correctly to prototype A3.
5. Go to step 3 again until the entire cover list is empty. (Select B1 and remove B1 and B2)

The selection algorithm can not generate the minimum number of prototype nor the best prototype with the highest recognition rate of the unseen data (test data). But it will generate the prototype that will 100% correctly classify the data used for prototype selection. Hence, our data can be separated into two group training and testing data. The prototypes are selected from the training data and used in recognition of the testing data.

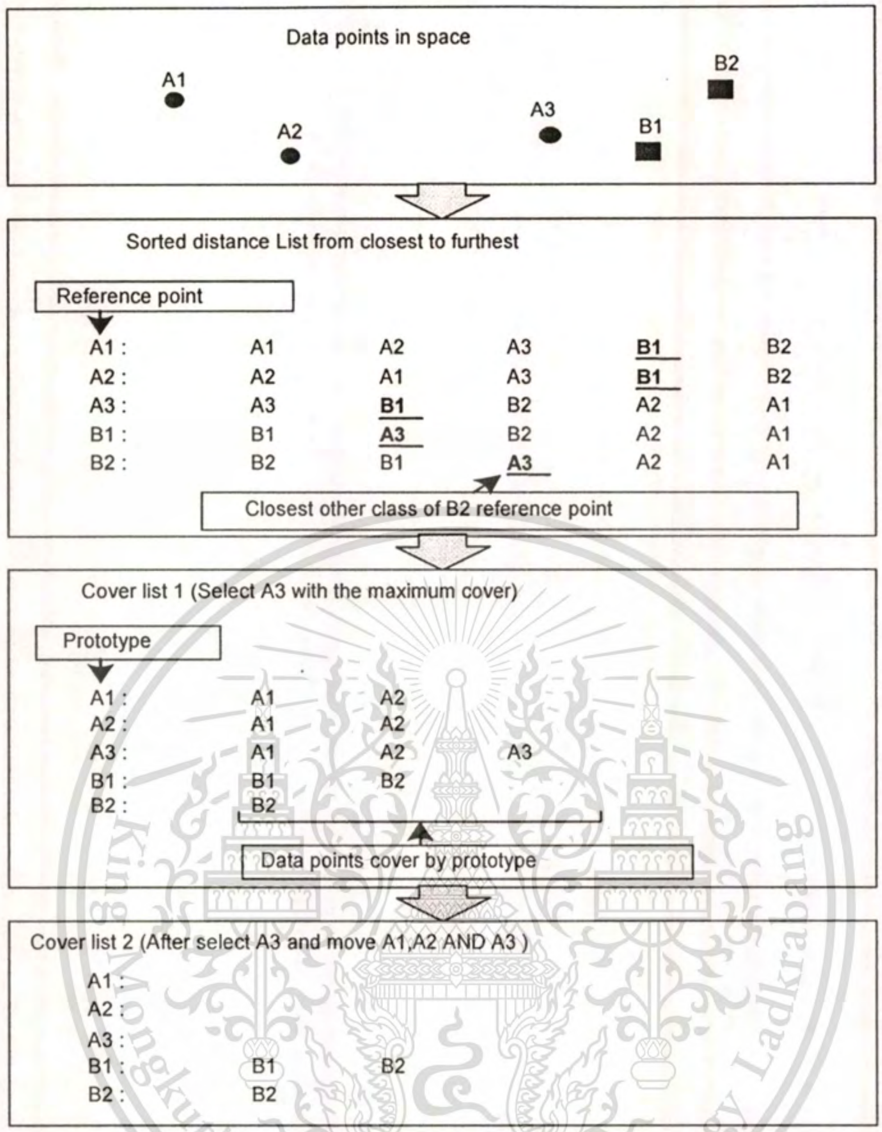


Fig 5.11 Prototypes selection.

Chapter 6

Recognition Experiments and Results

6.1 Recognition of Lao handwritten character

This chapter describes Lao handwritten character recognition experiment. The proposed alternative curvature tree feature described in the chapter 3 and the elastic matching method [9], single representation method [39], and segmentation by dominant point method [31] are compared in recognition performance. The experiment in handwritten is performed using our database consisting of 19,481 Lao handwritten characters from 18 writers, belonging to 27 classes as shown in the Fig 6.1. The prototypes of characters are selected using prototype selection algorithm [40] as described in section 5.4. The handwritten data was partitioned into 5 sets of training and testing data as shown in Table 6.1.

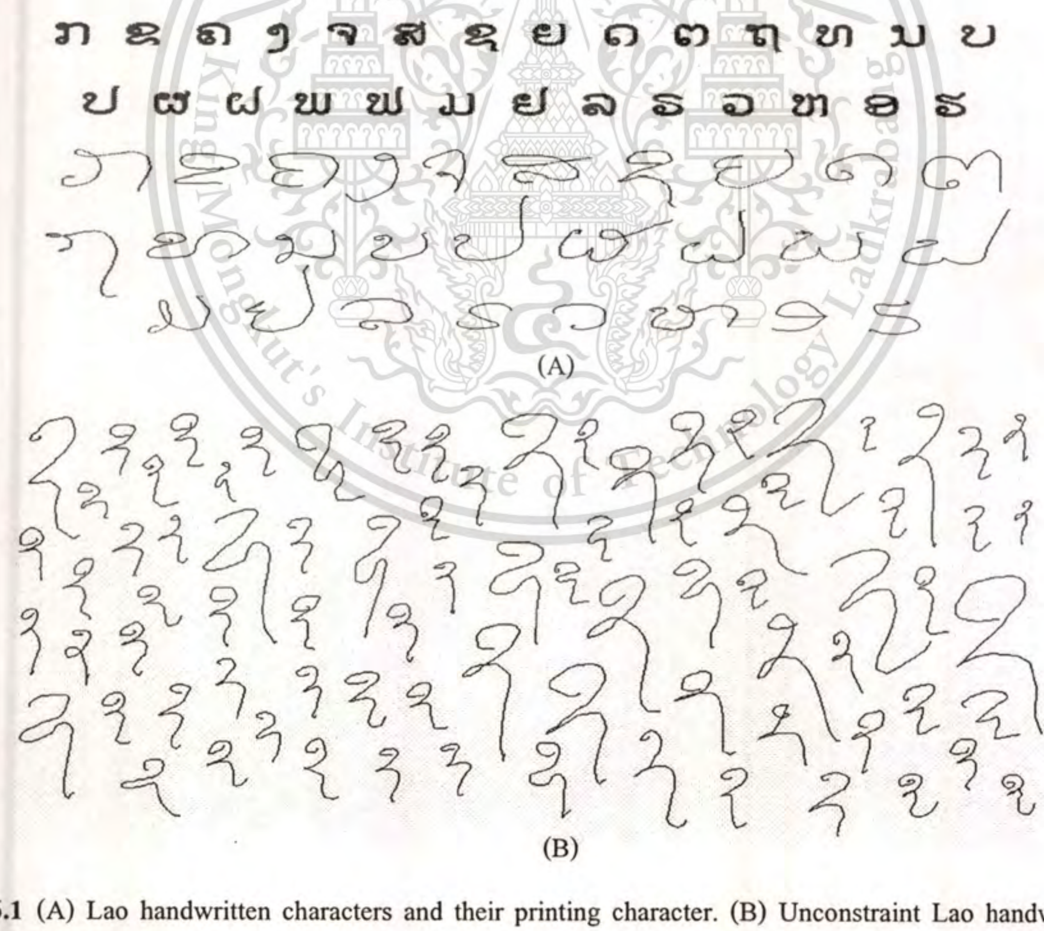


Fig 6.1 (A) Lao handwritten characters and their printing character. (B) Unconstraint Lao handwritten character “ຊ”

Table 6.1 Experiment handwritten training and testing data sets randomly partition from 19481 handwritten data set.

Data	Number Test set	Number Train set
Data set 1	15356	4125
Data set 2	15357	4124
Data set 3	15357	4124
Data set 4	15357	4124
Data set 5	15357	4124

6.2 Recognition results of Lao handwritten character

The training data set is used in prototypes selection and they are used for classification of the train data set. The numbers of prototypes selected for each data set are shown in Table 6.2 along with their recognition accuracy and recognition time. The average recognition rates of the proposed method is 98.97%%, elastic matching method is 96.476%, single representation method is 96.196% and dominant point method is 89.443%. The average recognition times per character using the PC with 1.4 GHz Pentium 4 is 0.00110 second for the proposed method, 0.05416 second for elastic matching method, 0.00078 second for single representation method and 0.00222 second for dominant point method. Table 6.3 shows the number of Lao handwritten character error of each method in each data set.

Table 6.2 Recognition result

Methods	Data set	Set 1	Set 2	Set 3	Set 4	Set 5	Average
	Test	15356	15357	15357	15357	15357	15357
	Train Data	4125	4124	4124	4124	4124	4124
Proposed method	Prototype	325	326	329	327	319	325
	Recognition (%)	99.08	98.87	98.94	99.10	98.88	98.974
	Average Time (s)	0.00119	0.00071	0.00107	0.00098	0.00152	0.00110
Elastic matching	Prototype	434	403	379	406	401	405
	Recognition (%)	96.31	96.62	96.48	96.60	96.37	96.476
	Average Time (s)	0.05780	0.05372	0.0512	0.05398	0.05404	0.05416
Single representation	Prototype	459	423	436	453	448	444
	Recognition (%)	96.23	96.35	96.02	96.18	96.20	96.196
	Average Time (s)	0.00076	0.00080	0.00075	0.00080	0.00079	0.00078
Dominant point	Prototype	325	326	329	327	319	325
	Recognition (%)	89.62	90.06	89.01	90.54	87.99	89.443
	Average Time (s)	0.00226	0.00210	0.00221	0.00210	0.00241	0.00222

Table 6.3 Comparison of handwritten error.

Method	Data set				
	Set 1	Set 2	Set 3	Set 4	Set 5
Tests data	15356	15357	15357	15357	15357
Proposed method	140	173	162	137	171
Elastic matching	559	519	541	522	558
Single representation	569	560	611	586	583
Dominant point	1562	1517	1678	1445	1783

In the Fig 6.2 shows comparison average recognition rate of Lao handwritten character from 5 sets in each character of each method. Experiment with test data set 1-5 as show below.

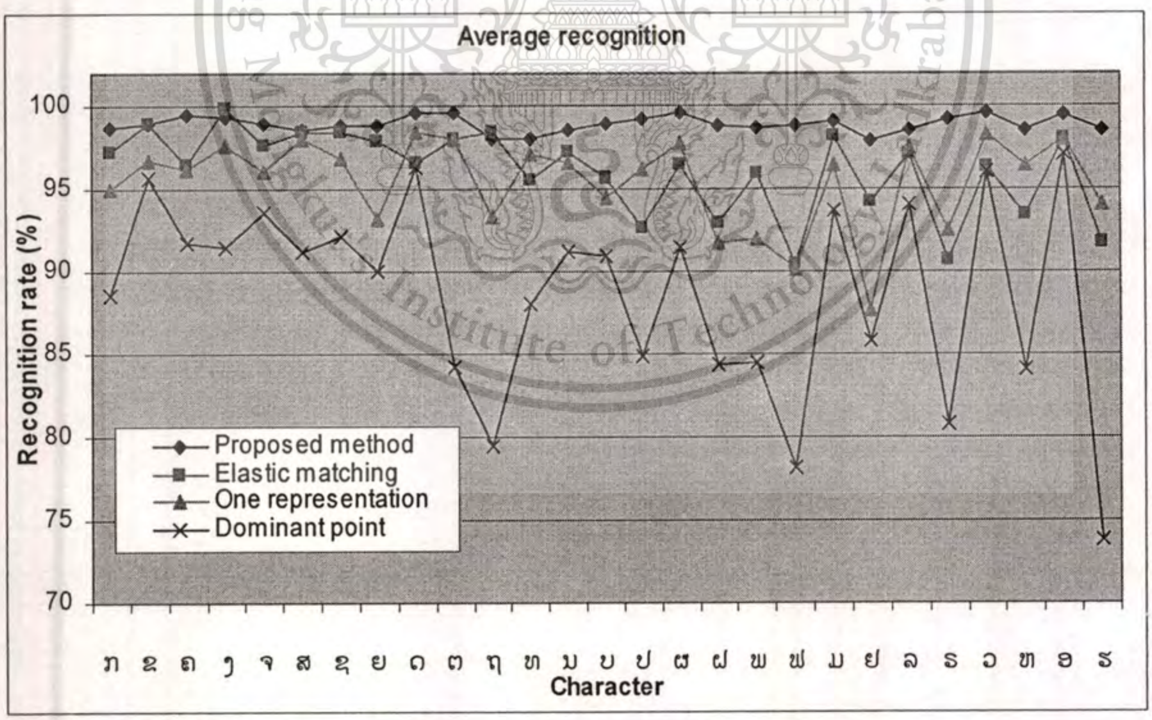


Fig 6.2 Average recognition rate of Lao handwritten character from 5 data sets.

From the Table 6.4 to Table 6.7 are shows summation the right recognition and wrong recognition character of each method from 5 sets. The highlight is wrong recognition character.

Table 6.4 Confusing matrix of Lao handwritten character recognition of the proposed method (5 sets)

Tests	Number	Character	Prototypes																										
			ກ	ຂ	ຄ	ງ	ຈ	ສ	ຊ	ຍ	ດ	ຕ	ຖ	ທ	ນ	ບ	ປ	ຜ	ຝ	ພ	ຟ	ມ	ຢ	ຣ	ລ	ວ	ຫ	ອ	ຮ
	2562	ກ	2530	0	0	0	0	0	0	1	0	0	26	0	0	0	0	0	1	0	0	1	0	0	0	0	3	0	0
	3133	ຂ	0	3101	0	0	0	0	18	0	2	0	0	0	4	7	0	0	0	1	0	0	0	0	0	0	0	0	0
	3312	ຄ	0	0	3294	5	0	0	3	0	6	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3964	ງ	0	4	1	3941	0	0	6	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2964	ຈ	0	5	0	0	2935	0	1	0	1	0	0	0	1	3	1	0	0	0	0	0	0	17	0	0	0	0	0
	3988	ສ	0	0	0	0	17	3930	0	5	0	0	0	0	0	0	0	1	0	9	13	1	1	2	2	0	4	3	0
	2716	ຊ	0	10	5	0	0	0	2683	0	0	1	1	7	4	3	0	0	0	1	0	0	0	1	0	0	0	0	0
	3128	ຍ	0	0	0	0	0	0	0	3091	0	0	0	0	0	0	0	2	1	0	0	1	30	0	3	0	0	0	0
	3928	ດ	0	1	10	2	0	0	0	0	3913	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4343	ຕ	0	0	1	0	0	0	4	0	3	4327	0	7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2737	ຖ	32	1	1	0	0	0	5	0	0	1	2682	2	0	0	0	0	0	6	0	0	0	0	0	0	7	0	0
	3452	ທ	0	7	13	0	0	0	25	0	0	2	0	3387	0	0	11	0	0	4	1	0	1	0	1	0	0	0	0
	3416	ນ	0	17	8	0	0	0	6	0	3	9	0	1	3366	6	0	0	0	0	0	0	0	0	0	0	0	0	0
	2621	ບ	0	1	0	0	0	0	0	0	0	0	0	0	0	2594	19	0	0	5	2	0	0	0	0	0	0	0	0
	2427	ປ	0	0	0	0	0	0	0	0	0	0	0	3	0	2	2409	0	0	0	8	0	1	2	1	0	1	0	0
	3000	ຜ	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	2987	1	0	0	4	1	0	0	0	0	0	0
	2556	ຝ	6	0	0	0	0	1	0	15	0	0	0	0	0	0	0	2	2526	0	0	4	0	0	2	0	0	0	0
	2260	ພ	0	0	0	0	1	4	6	0	0	0	0	0	0	2	0	0	0	2232	15	0	0	0	0	0	0	0	0
	1477	ຟ	0	0	0	0	0	2	0	0	0	0	0	3	1	0	1	0	2	9	1459	0	0	0	0	0	0	0	0
	2063	ມ	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	1	4	0	0	2045	0	0	0	0	0	0	0
	1372	ຢ	1	0	0	0	0	0	0	20	0	0	0	0	0	0	1	0	5	0	0	0	1342	0	2	0	0	0	1
	2284	ຣ	0	0	0	0	31	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2252	0	0	0	0	0
	2456	ລ	2	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	2437	0	0	0	15
	3076	ວ	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	3064	0	5	0
	2445	ຫ	4	0	0	0	0	6	0	2	0	0	1	1	0	1	0	0	0	2	8	0	1	2	3	0	2413	0	1
	2931	ອ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	6	0	0	3	0	2918	0
	2173	ຮ	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	2	0	26	0	0	0	2138

Table 6.5 Confusing matrix of Lao handwritten character recognition of the elastic matching (5 sets)

Tests	Number	Character	Prototypes																										
			ກ	ຂ	ຄ	ງ	ຈ	ສ	ຊ	ຍ	ດ	ຕ	ຖ	ທ	ນ	ບ	ປ	ຜ	ຝ	ພ	ຟ	ມ	ຢ	ຣ	ລ	ວ	ຫ	ອ	ຮ
	2562	ກ	2493	0	2	0	0	0	0	0	11	2	32	6	0	0	3	0	2	0	1	0	0	0	1	0	9	0	0
	3133	ຂ	0	3102	2	0	0	0	15	0	1	0	3	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
	3312	ຄ	1	0	3195	0	0	0	0	0	106	8	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3964	ງ	0	1	0	3961	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2964	ຈ	0	0	0	0	2894	0	0	0	0	0	0	0	2	0	1	0	0	0	1	0	0	37	0	29	0	0	0
	3988	ສ	0	0	0	0	3	3928	0	0	0	0	0	0	0	0	2	0	14	0	0	0	0	18	2	0	0	4	17
	2716	ຊ	0	37	0	1	0	0	2674	0	0	0	2	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
	3128	ຍ	0	0	0	0	0	0	0	3061	0	0	0	0	8	23	0	16	0	1	0	0	19	0	0	0	0	0	0
	3928	ດ	3	0	56	0	0	0	0	0	3795	71	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4343	ຕ	1	0	3	0	0	0	0	0	81	4257	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2737	ຖ	33	6	0	0	0	0	2	0	2	0	2694	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3452	ທ	0	0	1	0	0	0	0	0	0	1	2	3298	0	0	1	0	0	0	1	0	0	0	0	0	148	0	0
	3416	ນ	0	0	0	0	0	0	0	1	0	0	0	0	3322	31	6	0	0	44	11	0	1	0	0	0	0	0	0
	2621	ບ	0	1	0	0	0	0	1	9	0	0	0	4	59	2509	5	2	0	30	0	1	0	0	0	0	0	0	0
	2427	ປ	0	0	0	0	0	0	0	0	0	0	0	0	4	1	2249	1	25	0	83	0	58	0	6	0	0	0	0
	3000	ຜ	0	0	0	0	0	0	0	4	0	0	0	0	2	11	2	2896	52	16	16	0	1	0	0	0	0	0	0
	2556	ຝ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	34	2375	2	125	0	6	0	0	0	0	0	0
	2260	ພ	0	1	0	0	0	0	0	1	0	0	0	4	20	34	2	9	2	2167	20	0	0	0	0	0	0	0	0
	1477	ຟ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	1	104	1	1336	0	3	0	0	0	0	0	0
	2063	ມ	0	0	0	0	0	0	0	3	0	0	0	0	0	3	3	6	11	7	3	2026	1	0	0	0	0	0	0
	1372	ຢ	0	0	0	0	0	0	0	10	0	0	0	0	0	0	50	0	12	0	7	0	1293	0	0	0	0	0	0
	2284	ຣ	0	0	0	0	36	8	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	2219	0	18	0	0	0
	2456	ລ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	3	0	0	0	0	0	2227	0	0	0	222
	3076	ວ	0	0	0	0	43	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	2	53	0	2962	0	13	0
	2445	ຫ	0	0	0	0	0	0	0	0	0	0	0	151	0	6	0	2	0	1	0	0	0	0	0	0	2285	0	0
	2931	ອ	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	7	7	0	0	0	9	3	0	24	0	2873	3
	2173	ຮ	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	167	0	0	0	1994

Table 6.6 Confusing matrix of Lao handwritten character recognition of the single representation (5 sets)

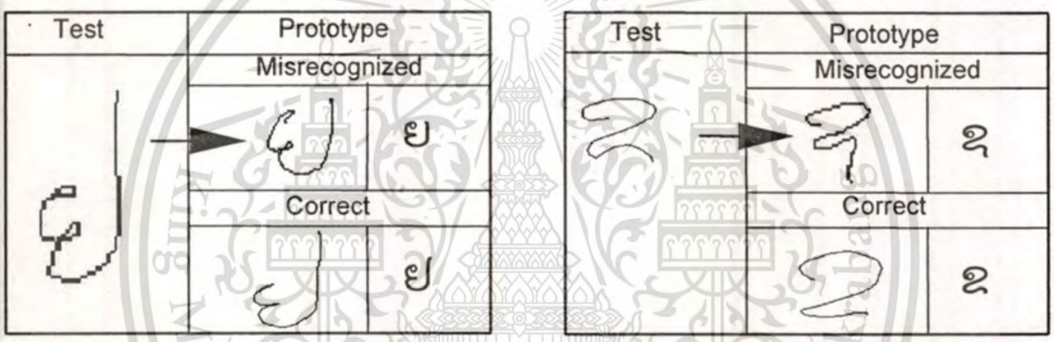
	Number	Character	Prototypes																										
			ກ	ຂ	ຄ	ງ	ຈ	ສ	ຊ	ຍ	ດ	ຕ	ຖ	ທ	ນ	ບ	ປ	ຜ	ຝ	ພ	ຟ	ມ	ຢ	ຣ	ລ	ວ	ຫ	ອ	ຮ
Tests	2562	ກ	2436	0	0	0	0	0	0	7	0	0	54	5	0	0	0	2	15	0	0	11	5	0	0	0	2	6	11
	3133	ຂ	0	3031	9	14	0	0	23	0	1	0	0	11	10	24	0	0	0	4	0	0	0	1	0	0	0	0	0
	3312	ຄ	0	7	3187	5	0	0	17	0	33	8	5	27	1	0	1	0	0	0	7	0	0	0	4	0	0	0	0
	3964	ງ	1	29	6	3869	3	0	25	0	9	0	0	0	6	0	0	0	0	0	4	0	0	0	1	0	1	0	1
	2964	ຈ	0	9	4	0	2848	7	8	2	4	0	5	2	6	12	5	0	0	1	1	0	0	28	1	4	6	0	0
	3988	ສ	0	0	0	0	4	3909	0	4	0	0	0	2	0	2	2	0	0	16	10	2	2	10	2	2	5	8	0
	2716	ຊ	0	13	8	3	2	0	2630	0	6	2	11	4	13	6	0	0	0	6	0	0	0	0	0	0	0	0	0
	3128	ຍ	3	0	0	0	4	0	0	2916	0	0	3	0	0	0	0	48	23	1	1	15	96	0	0	0	0	3	3
	3928	ດ	0	1	34	1	0	0	5	0	3865	11	1	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4343	ຕ	0	0	14	0	0	0	19	0	28	4258	0	11	3	0	0	0	0	3	1	0	0	0	0	0	0	0	0
	2737	ຖ	115	0	2	0	0	0	14	4	0	0	2555	8	0	0	0	1	1	14	0	10	0	0	0	0	0	0	4
	3452	ທ	0	3	19	0	0	0	30	0	0	8	1	3352	0	3	7	0	0	10	3	0	5	0	0	0	1	0	0
	3416	ນ	0	10	9	6	0	0	16	3	7	14	0	4	3302	10	1	0	1	20	4	0	0	0	0	0	0	0	1
	2621	ບ	0	9	0	0	1	1	4	4	0	0	0	6	1	2476	75	0	0	23	5	0	2	2	0	0	3	0	0
	2427	ປ	0	0	0	0	0	0	0	0	1	1	0	15	1	38	2335	0	0	3	18	0	0	2	1	0	0	0	2
	3000	ຜ	3	0	0	0	0	0	0	28	0	0	1	0	0	0	0	2933	14	1	0	3	2	0	0	0	1	2	2
	2556	ຝ	15	0	0	0	0	2	1	54	0	0	10	1	1	0	0	29	2344	0	10	21	31	0	8	0	4	0	16
	2260	ພ	0	1	0	0	0	17	11	2	0	0	10	10	6	18	1	2	0	2079	76	4	0	0	0	0	12	0	0
	1477	ຟ	0	0	0	0	1	8	2	1	0	0	0	12	2	5	9	0	9	88	1331	0	0	0	0	0	3	0	0
	2063	ມ	0	0	0	0	0	2	0	26	0	0	6	0	0	0	0	8	10	1	0	1991	4	0	0	0	2	4	0
	1372	ຢ	2	0	0	0	0	0	0	110	0	0	0	0	0	0	0	7	23	1	1	10	1203	0	0	2	1	0	2
	2284	ຣ	0	0	0	0	25	8	1	0	0	0	2	0	0	5	5	0	0	3	0	0	0	2227	0	1	0	0	0
	2456	ລ	18	0	1	0	0	2	0	0	0	0	1	0	0	0	19	0	1	0	0	0	1	0	2271	0	6	4	125
	3076	ວ	0	0	0	0	2	3	0	2	0	0	0	0	0	0	0	0	0	0	0	0	14	6	1	3024	0	11	7
	2445	ຫ	5	1	0	0	0	40	3	0	0	0	0	2	0	4	0	0	3	19	5	0	0	1	0	0	2358	2	0
	2931	ອ	1	0	0	0	0	18	0	0	0	0	0	0	0	0	0	1	0	0	0	8	5	0	0	10	1	2875	5
	2173	ຮ	11	0	0	0	0	0	0	2	0	0	2	0	0	0	0	0	1	0	0	0	3	0	102	0	3	2	2046

Table 6.7 Confusing matrix of Lao handwritten character recognition of the dominant point (5 sets)

	Number	Character	Prototypes																										
			ກ	ຂ	ຄ	ງ	ຈ	ສ	ຊ	ຍ	ດ	ຕ	ຖ	ທ	ນ	ບ	ປ	ຜ	ຝ	ພ	ຟ	ມ	ຢ	ຣ	ລ	ວ	ຫ	ອ	ຮ
Tests	2562	ກ	2270	0	0	0	0	3	0	28	0	0	129	0	0	0	1	27	5	0	11	38	18	0	0	0	18	6	0
	3133	ຂ	0	2993	45	2	3	2	9	0	2	1	0	18	3	18	0	13	0	10	2	0	0	0	1	0	1	0	0
	3312	ຄ	0	62	3031	2	10	1	36	0	60	1	2	34	16	0	6	10	0	3	14	0	0	0	0	0	0	0	0
	3964	ງ	0	98	98	3623	15	1	54	0	0	12	1	6	21	19	0	2	0	0	5	0	0	0	0	0	0	0	0
	2964	ຈ	0	8	4	0	2775	15	6	2	5	0	1	14	3	9	17	1	0	0	10	1	1	31	11	12	24	0	2
	3988	ສ	3	5	7	0	93	3635	4	14	0	0	1	3	3	30	7	9	4	43	35	15	0	6	17	0	7	3	35
	2716	ຊ	0	32	78	0	16	2	2498	0	0	2	3	17	21	1	0	5	0	18	2	0	0	2	2	0	3	0	0
	3128	ຍ	1	0	0	0	0	6	0	2814	0	0	0	2	0	0	1	42	28	1	6	38	156	0	2	0	9	0	8
	3928	ດ	0	7	70	0	0	0	7	0	3787	33	0	14	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0
	4343	ຕ	0	14	123	1	0	9	217	0	8	3654	12	26	53	3	4	0	0	197	10	0	0	0	0	0	0	0	0
	2737	ຖ	193	0	39	0	25	12	27	41	0	7	2164	8	1	17	19	50	16	11	9	20	6	0	1	0	47	0	0
	3452	ທ	2	27	80	0	9	0	97	0	3	26	1	3045	5	68	25	0	0	38	3	0	1	0	0	0	12	0	0
	3416	ນ	0	31	37	7	0	4	13	0	0	20	0	5	3113	7	0	1	3	119	46	0	1	0	0	0	0	0	0
	2621	ບ	0	13	7	0	2	10	11	4	0	4	0	47	17	2382	67	0	1	17	8	0	8	0	0	0	11	0	3
	2427	ປ	0	0	8	0	16	1	0	0	0	3	0	34	7	129	2060	0	0	20	19	0	27	10	55	0	8	0	18
	3000	ຜ	3	0	0	0	0	5	0	137	0	0	1	0	0	0	0	2718	58	4	3	26	6	0	0	0	0	0	0
	2556	ຝ	15	0	0	0	0	2	0	112	0	0	0	0	0	2	1	71	2151	5	88	23	55	1	4	0	6	0	0
	2260	ພ	0	2	1	0	7	20	10	21	0	0	1	23	14	55	0	29	18	1904	108	10	4	0	0	0	18	0	1
	1477	ຟ	0	0	11	0	1	7	4	32	0	1	0	10	46	9	4	3	37	130	1153	0	14	5	0	0	2	0	0
	2063	ມ	2	0	0	0	1	4	0	62	0	0	0	0	0	1	0	29	15	1	1	1934	0	2	0	0	1	0	0
	1372	ຢ	2	0	0	0	0	2	0	106	0	0	0	0	0	0	6	9	7	0	6	11	1175	0	19	0	3	0	16
	2284	ຣ	0	2	0	0	51	15	3	3	0	1	1	1	0	1	3	0	2	0	12	0	0	2154	17	6	0	0	5
	2456	ລ	2	1	6	0	1	7	0	6	0	1	0	0	0	0	78	0	0	0	1	4	12	1	1983	3	6	6	331
	3076	ວ	0	0	0	0	5	3	0	3	0	0	0	0	0	0	0	2	1	0	0	1	13	9	21	2954	0	5	53
	2445	ຫ	21	1	1	0	34	46	3	8	0	0	24	9	0	19	7	18	3	136	17	6	21	0	1	0	2057	1	8
	2931	ອ	3	0	0	0	0	13	0	0	0	0	0	0	0	0	0	5	1	0	0	16	2	0	3	4	0	2848	31
	2173	ຮ	8	0	0	0	0	63	0	8	0	0	0	0	0	0	0	11	0	0	2	6	25	0	406	19	5	12	1602

6.3 Analysis of recognition results

The examples of some of the errors in the recognition are shown in the following Figures. In Fig 6.3 the unknown Lao handwritten character "ຢ" and prototype "ຍ" have one segment with the same direction. It miss differentiate the short and long tail. The same kind of error also occurring in the case of character "ຂ" is recognized as "ຊ". Both of them have the same shape with three segments with the same direction. They are only difference in segment three where one has short tail and the other has long tail. The recognition of characters has short and long tail is shown in the table 6.8 and 6.9. Misrecognition of Lao handwritten character "ຈ" and "ລ" is shown in Fig 6.4. They have same segment and same rotation. It can be seen that is rather difficult for the proposed method to recognize some of characters having same curve. But it is easy to recognize curve segment with clockwise or counter clockwise direction.



(A) (B)

Fig 6.3 Misrecognition of handwritten characters with the same shape but difference in tail length.

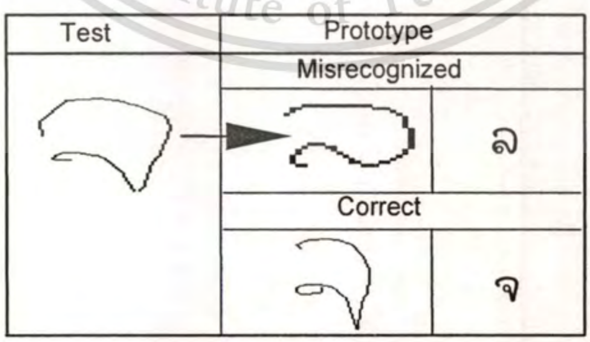


Fig 6.4 Misrecognition of handwritten character with the same shape and rotation.

Table 6.8 Recognition of character has short tail and long tail from the proposed method

Tests		prototype									
		ກ	ຂ	ຊ	ຍ	ຖ	ບ	ປ	ພ	ຟ	ຢ
2562	ກ	2530	0	0	1	26	0	0	0	0	0
3133	ຂ	0	3101	18	0	0	7	0	1	0	0
2716	ຊ	0	10	2683	0	1	3	0	1	0	0
3128	ຍ	0	0	0	3091	0	0	0	0	0	30
2737	ຖ	32	1	5	0	2682	0	0	6	0	0
2621	ບ	0	1	0	0	0	2594	13	5	2	0
2427	ປ	0	0	0	0	0	2	2409	0	8	1
2260	ພ	0	0	6	0	0	2	0	2232	15	0
1477	ຟ	0	0	0	0	0	0	1	9	1459	0
1372	ຢ	1	0	0	20	0	0	1	0	0	1342

Table 6.9 Recognition of character has short tail and long tail from elastic matching method

Tests		prototype									
		ກ	ຂ	ຊ	ຍ	ຖ	ບ	ປ	ພ	ຟ	ຢ
2562	ກ	2493	0	0	0	32	0	3	0	1	0
3133	ຂ	0	3102	15	0	3	0	0	10	0	0
2716	ຊ	0	37	2674	0	2	1	0	1	0	0
3128	ຍ	0	0	0	3061	0	23	0	1	0	19
2737	ຖ	33	6	2	0	2694	0	0	0	0	0
2621	ບ	0	1	1	9	0	2509	5	30	0	0
2427	ປ	0	0	0	0	0	1	2249	0	83	58
2260	ພ	0	1	0	1	0	34	2	2167	20	0
1477	ຟ	0	0	0	0	0	0	32	1	1336	3
1372	ຢ	0	0	0	10	0	0	50	0	7	1293

Also it can be seen from Fig 6.5 that misrecognition of character in the elastic matching method is due to same shape but different direction. Example in the Fig 6.5 (A) shown Lao handwritten character “ປ” and “ຟ” shows that they have the same shape but different direction at head. Character “ປ” has direction at head is counter clockwise but character “ຟ” has direction at head is clockwise.

In Fig 6.6 show misrecognition of elastic matching due to misrecognition of dominant curve. Example in Fig 6.6 (C) Lao handwritten character “ຕ” is misrecognized as “ດ” because elastic matching misses dominant curve of character “ຕ”on the top. These kinds of errors also occur in Lao character as shown in the Fig 6.6 (G) and (H). and shown

It can be seen that elastic matching is difficult to recognize of Lao handwritten character having curve segment with clockwise or counter clockwise direction. But it is easy to recognize character having similar shape.

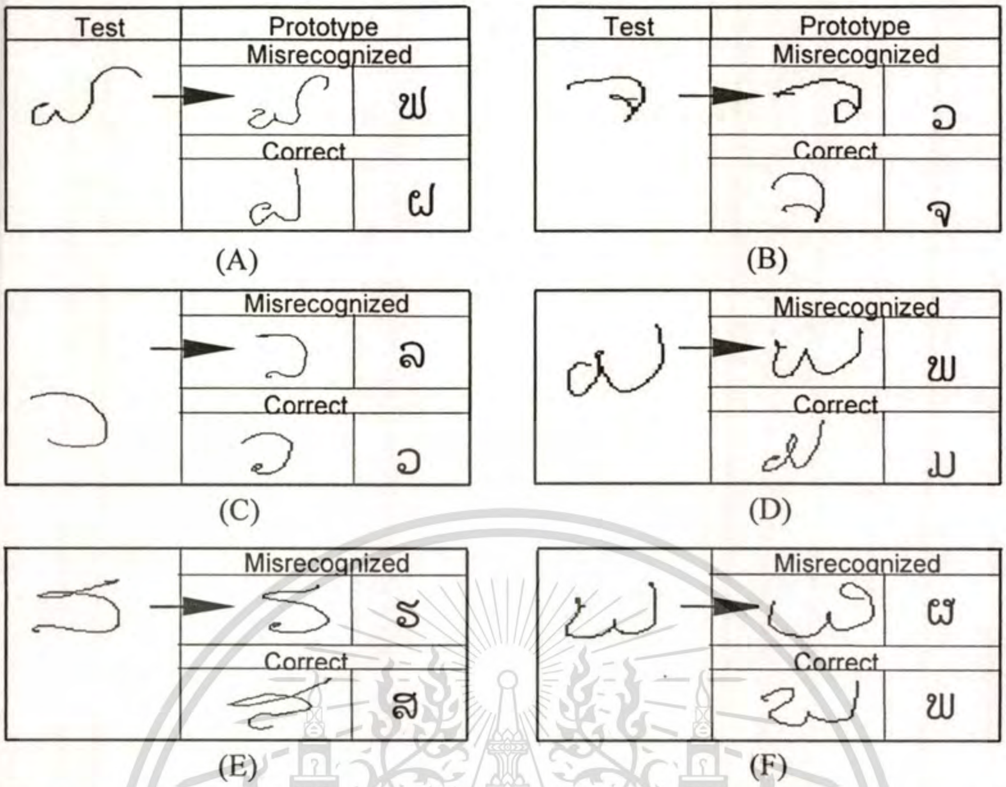


Fig 6.5 Elastic matching misrecognition examples of Lao characters with the same shape but difference direction.

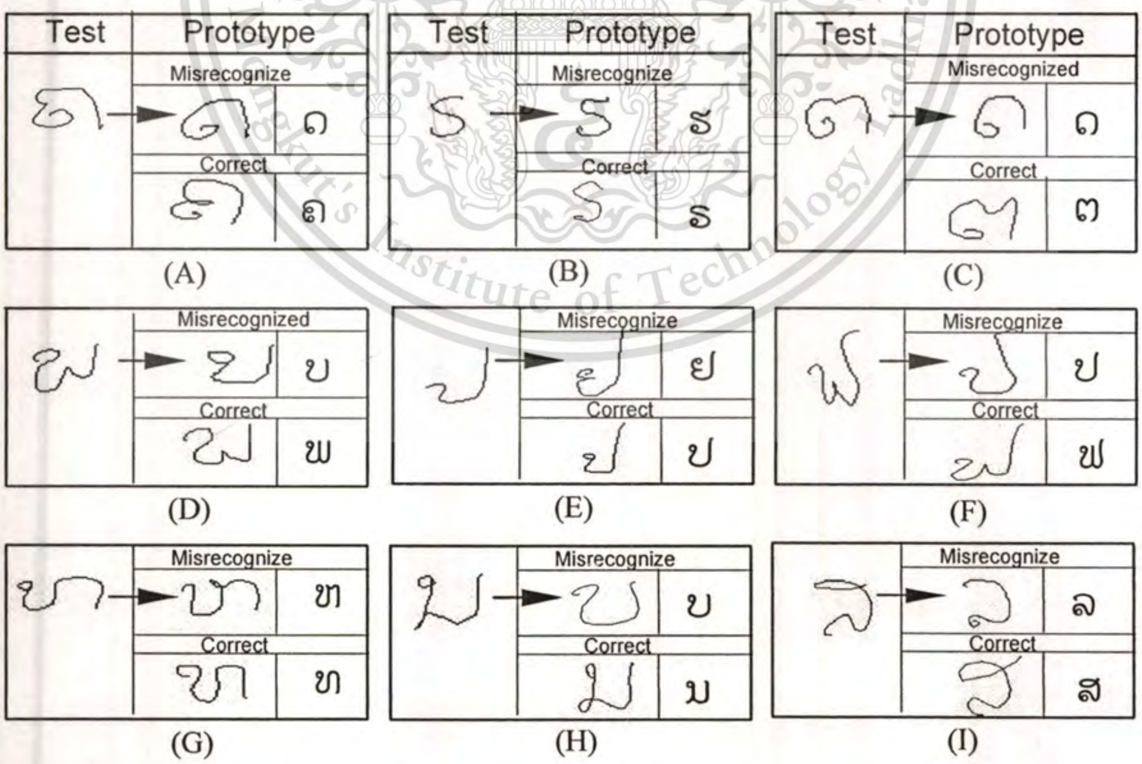


Fig 6.6 Misrecognition of Lao character in elastic matching due to curvature direction.

The misrecognition by single representation method is shown in the Fig 6.7. For Lao handwritten characters, it misses some paths or misses some real curve segments due to fixed threshold as in Fig 6.7 (A) and (B). In Fig 6.7 (C) and (D), it misrecognizes because characters have only single segment with the same direction.

From recognition results, it is difficult for the single representation to identify small real curve from noise curve using fixed threshold as shown in the Fig 6.8.

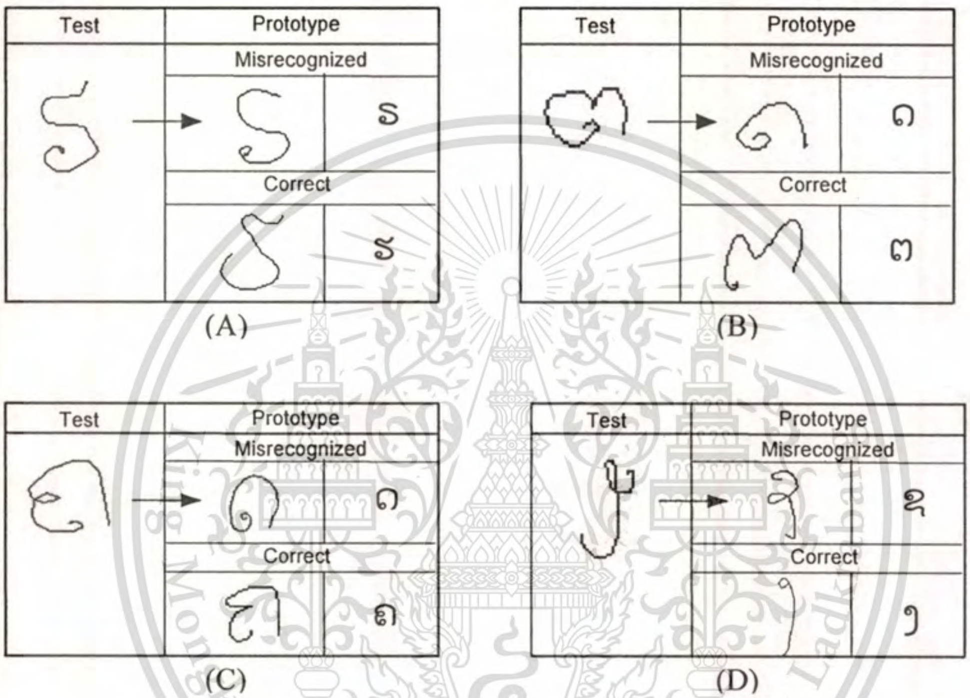


Fig 6.7 Misrecognitions by single representation of Lao handwritten character.

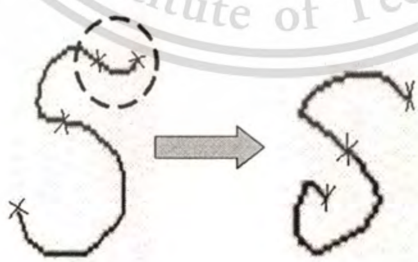


Fig 6.8 Misrecognition of Lao character because of missing the small real curve at tail.

Chapter 7

Conclusions

Handwritten character recognition has been approached in various ways. Many methods have been proposed with high accuracy rate of recognition in many languages. Different languages have difference of characteristic and need different approaches for recognition. For Lao handwritten, very few methods have been proposed.

7.1 Alternative curvature tree feature representation.

In the proposed method, unique features of Lao handwritten character with similar shapes having small or large curve segments with clockwise and counter clockwise direction, are taken into account. Character is divided into segments at the change rotation points in clockwise and counter clockwise direction. These segments are classified to three groups of segment as: real segment, uncertain segment and noise segment. The noise segment is discarded by merging it with adjacent segments (Tree segments will merge together). After merged some segments changed because of merging. Then segments are reclassified again until remain only real segment and uncertain segment. After that tree representation is created. One path in the tree means one representation of character.

7.2 Performance of alternative curvature tree.

The proposed method can recognize correctly Lao handwritten characters with multiple representations even though the elastic matching, single representation and dominant point can not do this. The average recognition rate of the proposed method is 98.97%. It is higher than 96.47% for elastic matching, 96.19% for single representation and 89.443% for dominant point. The average time required to the recognition of the proposed method is 0.00110 second, which is much faster than 0.05416 second for elastic matching. For single representation is 0.00078 second and for dominant point is 0.00222 second.

From experiment, the elastic matching method easily recognize character with similar of shape but has difficulty to recognize character with similar shape but difference rotation of curvature (clockwise, counter clockwise direction) as shown in the Fig 7.1. The elastic matching method also missed dominance of curve of character as shown in the Fig 7.2. The single representation is tried to solve this weak point by including curvature rotation in the feature. But, it faces another problem of uncertainty in small curvature, where it cannot distinguish small noise curve from the real one as shown in the Fig 7.3. The dominant point approach also considers curvature rotation, and again has the problem of variation in number of dominant points of the same character as shown in the Fig 7.4. The proposed method overcomes this weak point by using curvature feature and reduces variation by allowing multiple representations. This representations, allow uncertain noise curve to be in one of the representation. The noise can then easily be identified by comparing with other characters.

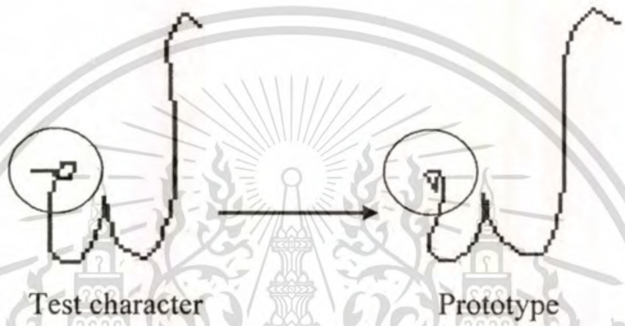


Fig 7.1 Elastic matching method misrecognized due to direction of curve at head

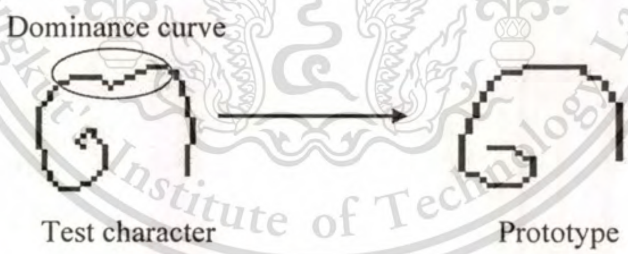


Fig 7.2 Elastic matching method missing dominant curve in the matching

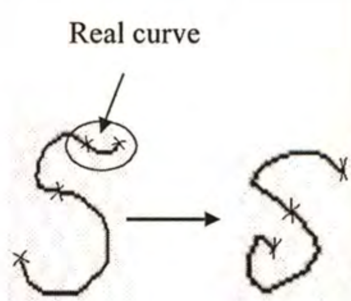


Fig 7.3 Single representation misrecognition due to treating of small curve as a noise

Fig 7.4 Variant in dominant point of the same character

Reference

[1] H. Nishida, "Model-Based Shape Matching with Structural Feature Grouping", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 3, March 1995, pp. 315-320.

[2] E.J. Bellegarda, J.R. Bellegarda, J.H. Kim, "On-line handwritten character recognition using parallel neural networks", Acoustics, Speech, and Signal Processing, IEEE International Conference on , vol. 2, 19-22 April 1994, pp. 605 -608.

[3] J. Hu, M. K. Brown, and W. Turin, "HMM based on-line handwriting recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (10), 1996, pp. 1039-1045.

[4] V. Vuori, J. Laaksonen, E. Oja, J. Kangas, "Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters", International Journal on Document Analysis and Recognition, 3, 2001, pp 150-159.

[5] K. F. Chan and D.Y. Yeung "Recognition On-line handwritten alphanumeric characters though flexible structural matching", Pattern Recognition vol. 32, 1999,pp. 1099-1114.

[6] M. Okamoto, K. Yamamoto, "On-line Handwritten Character Recognition Method using Directional Features and Clockwise/Counterclockwise Direction Change Feature", Fifth International Conference on Document Analysis and Recognition, Bangalore, India, 20 - 22 September, 1999, pp. 491-494.

[7] I. Methasate, and S. Sea-Tang, "On-line Thai Handwriting Character Recognition Using Stroke Segmentation with HMM", in Proceedings of the 20th IASTED International Conference on Applied Informatics (AIA'02), Innsbruck, Austria, 18-21 February. 2002, pp. 59-62.

[8] Y. Ge, F.-J. Guo, L.-X. Zhen, Q.-S. Chen, "Online Chinese character recognition system with handwritten Pinyin input", Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05), 29 Aug.-1 Sept, 2005, Vol. 2, pp 1265 – 1269.

[9] N. Joshi, G. Sita, A. G. Ramakrishnan and S. Madhvanath. "Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition", Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)

[10] S. Al-Emami and M. Usher, "On-Line Recognition of Handwritten Arabic Characters", IEEE Transactions on pattern analysis and machine intelligence, Vol. 12 No. 7, July 1990, pp 704-710.

[11] M. J. Jeo, H. J. Lee, "A Combined Method on the Handwritten Character Recognition", Document Analysis and Recognition, 1995, Proceedings of the Third International Conference on , vol. 1 , 14-16 Aug. 1995 pp. 112 -115.

[12] W. S. Kim and R.-H. Parks "Off-line Recognition of Handwritten Korean and Alpanumeric Character using Hidden Markov Models", Pattern Recognition vol. 29, N0.5, 1996,pp. 845-858.

[13] S. N. Srihari, "Integration of hand-written address interpretation technology into the United States postal service remote computer reader system", In Proceedings of the 4th international conference document analysis and recognition (ICDAR'97) Vols. I and II, 1997, pp. 892–896.

[14] D. Gillevic and C.Y. Suen, "HMM-KNN Word Recognition Engine for Bank Cheque Processing," Proceeding. 14th International Conference Pattern Recognition, Driubaiie, Australia, Aug. 1998, pp. 1526-1.529.

[15] K. Huang, and H. Yan, "Off-line signature verification using structural feature correspondence", Pattern Recognition, Vol. 35, 2002, pp, 2467-2477.

[16] R. plamondon and S.N Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey", IEEE Transactions on pattern analysis and machine intelligence. Vol. 22, No. 1. JANUARY 2000.

[17] C.-K. Lin, B. Jeng, "On-line recognition of handwritten Chinese characters and alphabets", International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 3-6 April. 1990, pp 2029 – 2032.

[18] C. Tappert, C. Y. Suen and T. Wakahara, "The state of the art in on-line handwriting recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(8), 1990, pp 787-808.

[19]. A. Kosmala, G. Rigoll, S. Lavirotte, L. Pottier "On-Line Handwritten Formula Recognition using Hidden Makov Model and Context Dependent Graph Grammars", Fifth International Conference on Document Analysis and Recognition, Bangalore, India, 20 - 22 September, 1999, pp. 107-110.

[20] K. Intawongkam and K. Chamnongthai, "A method of Hand Writing Lao Alphabet Recognition by Using Back-Propagation Neural Networks", 21th Electrical Engineering Conference (EECON), 1998.

[21] V. MOUNGSINH, "The Fuzzy and Back-Propagation Neural Techniques in Recognition of the Printed Lao Characters", Master thesis, Faculty of Engineering, King Mongkut's Institute of Technology Ladhrabang, 2000.

[22] V. Vuori, "Adaptation in On-line Recognition of Handwriting", Master thesis, Helsinki University of Technology, 1999.

[23] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition". In Proceedings of International Conference on Acoustics, Speech and Signal Processing, 1989, pp. 267-295.

[24] J. Hu, M. K. Brown, and W. Turin, "HMM based on-line handwriting recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (10), 1996, pp. 1039-1045.

[25] J. Hu, S. G. Lim, and M. K. Brown, "Writer independent on-line handwriting recognition using an HMM approach", Pattern Recognition 33 (1), 2000, pp133-147.

[26] S. R. Veltman, and R. Prasad, "Hidden Markov models applied to on-line handwritten isolated character recognition", IEEE Transactions on Image Processing 3 (3), 1994, pp. 314-318.

[27] K. Jung, and H. J. Kim, "On-line recognition of cursive Korean characters using graph representation", Pattern Recognition 33, 2000, pp.399-412.

[28] H.-Y. Kim, and J. H. Kim, "Hierarchical random graph representation of handwritten character and its application to Hangul recognition", Pattern Recognition 34 (2), 2001, pp.187-201.

[29] C.-K. Lin, K.-C. Fan, and F. T.-P. Lee, "On-line recognition by deviation expansion model and dynamic programming matching", Pattern Recognition 26 (2), 1993, pp. 259-268.

[30] Y.-Y Yang, "Adaptive recognition of Chinese characters: Imitation of psychological process in machine recognition", IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 28 (3), 1998, pp. 253-265.

[31] H. Yuen, "A chain coding approach for real-time recognition of on-line handwritten characters", ICASSP'96, Atlanta, USA, 1996, pp. 3426 - 3429.

[32] P. Scattolin, "Recognition of handwritten numerals using elastic matching". Master's thesis, Concordia University, Canada, 1995.

[33] X Li,. and D.-Y. Yeung, "On-line handwritten alphanumeric character recognition using dominant points in strokes". Pattern Recognition 30 (1), 1997, pp. 31-44.

[34] S. D. Connell, and A. K. Jain, "Template-based online character recognition", Pattern Recognition 34, 2001, pp. 1-14.

[35] N. Mezghani, A. Mitiche, M. Cheriet, "On-line recognition of handwritten Arabic characters using a Kohonen neural network", Proceedings. Eighth International Workshop on Frontiers in Handwriting Recognition, 6-8 Aug. 2002, pp 490 – 495.

[36] S. Oh and C. Y. Suen, "A class-modular feedforward neural network for handwriting recognition", Pattern Recognition, Vol. 35, (1), January 2002, pp 229-244.

[37] Y. Li, J. Li, L. Meng, "Character Recognition Based on Hierarchical RBF Neural Networks", Sixth International Conference on Intelligent Systems Design and Applications, (ISDA 06), Vol. 1, Oct. 2006, pp127 – 132.

[38] M.F. Zafar, D. Mohamad, R. Othman, "Neural Nets for On-line Isolated Handwritten Character Recognition", IEEE International Conference on Engineering of Intelligent Systems, 22-23 April. 2006, pp 1 – 6.

[39] K. Bounnady, B. Kruatrachue, S. Wangsiripitak, "On-line Lao Handwritten Recognition with Proportional Invariant Feature", The Third World Enformatika Conference, WEC'05'. April 27-29, 2005, Istanbul, Turkey, pp 41-44.

[40] B.V. Dasarathy, "Minimal Consistent Set (MCS) Identification for Optimal Nearest Neighbor Decision Systems Design", IEEE Transactions on Systems, Man and Cybernetics, Vol. 24, No. 3, 1994, pp. 511-517.

[41] R.P.W. Duin, D. de Ridder, and D.M.J. Tax, "Featureless classification", Proc. 1st IAPR International Workshop on Statistical Techniques in Pattern Recognition (STIPR'97), Prague, 1997, pp 37-42.

Appendix

Appendix (A)

Publications

1. Khampheth Bounnady, Boontee Kruatrachue, Somkiat Wangsiripitak "On-line Lao Handwritten Recognition with Proportional Invariant Feature", Proceeding of the 3rd World Enformatika Conference (WEC'05), Istanbul, Turkey, April 27-29, 2005, pp 41-44.

2. Khampheth Bounnady, Boontee Kruatrachue, Takenobu Matsuura, "Curvature Tree For Isolated Unconstrained Lao Handwritten Character Recognition", Proceeding of the 2007 ECTI international conference (ECTI-CON 2007), Chiang Rai, Thailand, May 9-12, 2007, pp 1022-1025.

3. Khampheth Bounnady, Boontee Kruatrachue, Takenobu Matsuura, "Online Freestyle Handwritten Lao Characters Recognition Using Feature Tree", The Journal of Signal Processing, Vol. 11, No. 6, 2007.

ENFORMATIKA

VOLUME 5 2005

Edited By
Cemal ARDIL

On-line Lao Handwritten Recognition with Proportional Invariant Feature

Khampheth Bounnady, Boontee Kruatrachue, Somkiat Wangsiripitak

***Abstrac*—This paper proposed high level feature for online Lao handwritten recognition. This feature must be high level enough so that the feature is not change when characters are written by different persons at different speed and different proportion (shorter or longer stroke, head, tail, loop, curve). In this high level feature, a character is divided in to sequence of curve segments where a segment start where curve reverse rotation (counter clockwise and clockwise). In each segment, following features are gathered cumulative change in direction of curve (- for clockwise), cumulative curve length, cumulative length of left to right, right to left, top to bottom and bottom to top (cumulative change in X and Y axis of segment). This feature is simple yet robust for high accuracy recognition. The feature càn be gather from parsing the original time sampling sequence X, Y point of the pen location without re-sampling. We also experiment on other segmentation point such as the maximum curvature point which was widely used by other researcher. Experiments results show that the recognition rates are at 94.62% in comparing to using maximum curvature point 75.07%. This is due to a lot of variations of turning points in handwritten.

***Keywords*—handwritten feature, chain code, Lao handwritten recognition**

I. INTRODUCTION

THE on-line handwriting recognition has become an area of active research since 1960 [4], And have many researchers in the field of handwritten alphanumeric character recognition to use difference method for recognition such as geometrical and topological feature [5,6], statistic feature [7], and other algorithms [8,9] to perform recognition based on character shape. But for a good handwritten recognition system depends on main two attributes, first selected feature gathering from a handwritten character, second the recognizers that trained to remember feature of each character in order to cluster and recognize each input character. There are many papers on handwritten system proposed various features such as chain code sequence, chain code histogram, height and width ratio, number of transition from black and white pixel [2-3]. Most of these features are gather in zone positions in order to make

Khampheth Bounnady was with National University of Laos, Vientiane, Lao P.D.R. He is now graduate student in Computer Engineering Department, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok Thailand 10520. (e-mail: s6060215@kmitl.ac.th).

Boontee Kruatrachue is with the Computer Engineering Department, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok Thailand 10520. (e-mail: kkboonte@kmitl.ac.th).

Somkiat Wangsiripitak is with the Computer Engineering Department, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok Thailand 10520. (e-mail: somkiat@ce.kmitl.ac.th).

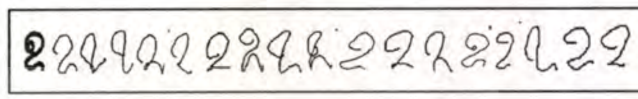


Fig. 1. Variation of handwritten character with and without head, short and long tail.

A lot of these features have been used successfully and reported good recognition rate. But all these good results depend on how recognition characters were written. For example, If a character is written very fast, its proportion changes and all the zone feature and height and width are not working. Fig. 1, show Lao character when written faster it tends to have larger head or some time no headed. It also has longer tail. All these variation destroy zone features. Proportion invariant. A character should have the same feature even if it is out of proportion (bigger or smaller head, shorter or longer tail or stroke, larger or smaller loop or curve).

The handwritten character set tested in this paper is Lao character set shown in Fig 2, Lao handwritten characters are usually written isolated with few connected cursively. All most all character can be written cursively without raised up pen (only one stroke). Although the tested character in this paper is Lao, the proposed feature can be used in English and other character also.

The high-level character feature proposed in this paper is viewing a handwritten character as a sequence of curve segments. Each curve segment is characterized by its degree of curvature which is measure by the cumulative angle difference from all sampling points with in the segments. If the cumulative is minus, it is a clockwise curve. Since some character may consist of the same number of segments with the same curve (ex. one segment with clockwise curve) other features of segments are gather to distinguish these characters.

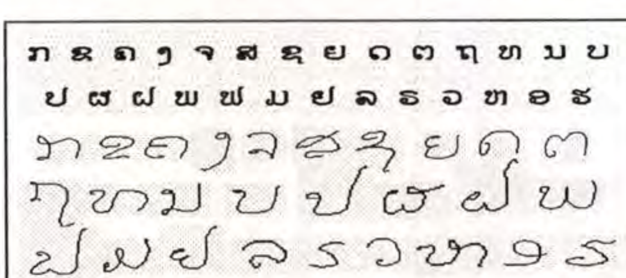


Fig. 2. Example of Lao isolated handwritten and printed characters

This additional segment feature cumulative curve length, cumulative length of left to right, right to left, top to bottom and bottom to top (cumulative change in X and Y axis of segment in both direction increasing and decreasing).

Since the main top level feature is curvature, the character can even be written by 180 degree rotation and still have the same curvature. Unfortunately, some characters when written rotately will become other characters, hence additional feature that are less robust to rotation are added to distinguish them.

II. FEATURE EXTRACTION

The handwritten character input are the sequence of XY points of pen location captured by an electronic writing tablet sampling with the same timing interval (equi-time sampling). Since the degree of curvature is the feature of interested, there is no need to re-sampling the pen location to make equi-distance sampling. The angle between two sampling points is measure in 360 degree and mapping into chain code of 1 to 8.99 [1] as shown in Fig. 3, Since the length of the points between two sampling point are not adjacent to each other the direction number is real number from 1 to 8.99 instead of integer chain code (1.5 direction number for 22.5 degree). If a character is written with a loop as letter 0 with one whole rotation counter clockwise, this character will have the same curvature feature of plus 6 independent from starting location or number of points in the segments. As long as the loop is closed back to the starting point, the curvature feature is 6. The curvature feature of a segment is a summation of direction difference between adjacent points from the starting point to the last point in the segment as shown in "Fig 3".

Before extract segmentation of a character, the noises that usually occur in the start and ending of pen trace are removed. This step is perform near the start and ending of the sequence of direction number where the point that has high direction difference to the previous point is a new start and ending point of a character as shown in Fig. 4.

A character is model as consecutive segments of clockwise and counter-clockwise direction. Consecutive segments of the same rotation (ex. both clockwise) is treated as one segment with one cumulative direction difference. Therefore, the points

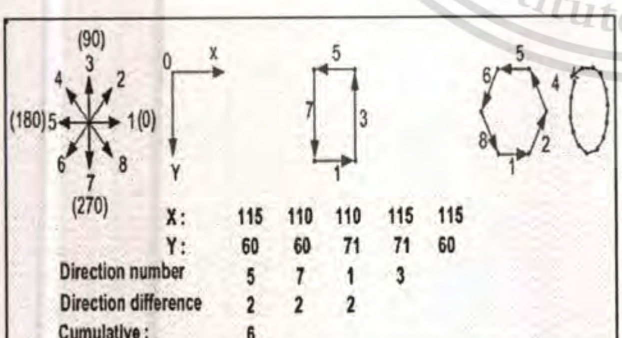
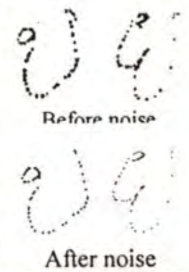


Fig. 3. Mapping of 360 degree to 1- 8.99 chain code and Calculation of Curvature feature.



Before noise reduction			
Index	Direction	Length	Difference direction
0	4	1	0
1	4	1	-2.5
2	1.5	1.414214	0.5
3	2	1.414214	0
4	2	1	
After noise reduction			
0	1.5	1.414214	0.5
1	2	1.414214	0
2	2	1	1
3	3	2	

Fig. 4. Noise at starting and ending of characters.

that separate a handwritten character into segments is a point where pen trace start reverse rotation from clockwise to counter clockwise and vice versa. This point can be identified by monitor the change of sign of the difference at each point. Some researchers [1] used the dominant point or point where the pen traces turn direction dramatically (commonly located at local maximum of direction difference).

We find that number of dominant points varied due to writing style, while the reverse rotation points are more stable (generate less and consistent number of segment from the same shaped character). In order to extract consistent number of segments, the reverse rotation point calculation from direction difference has to cancel noised as follow: If direction difference around point i is diff[i], the reverse rotation point is identified by the sign of diff[i] as follow table 1.

TABLE I
THE REVERSE ROTATION POINT

Sign of diff[i]			Comment
i-1	i	i+1	
+	-	-	The reverse rotation point is at i.
+	-	+	• Reverse rotation point is at i if abs(diff[i])>=1 • In case of diff[i]==diff[i-1] or diff[i]==diff[i+1] point i is not a reverse rotation point.

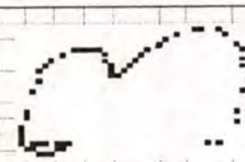
The change from - to + is treated the same as + to -. In some case the character consists of many small turn segments, all the segment with cumulative direction difference less than 1 is add to the adjacent segments to reduce variation due to very small turn, Jitter generated from writing tablet is canceled by delete segment with very small segment length (cumulative length with in the segment, example is segment number 4 in the fig. 5), The calculation of reverse rotation point and cumulative direction difference of segment is show in Fig. 5.

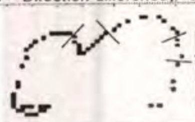
Some different characters may have the same curvature feature (with the same number of segments and cumulative direction difference, angle). In order to distinguish them, we use the differences in the XY coordinate between adjacent points in both X and Y axis.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Direction	1	1	1	1	1	6	5	6	5	5	4	5	4	3	3	3	2.4	3	2.68	2.4	3	2	2	1.59	1.59	1	1	1	1	8	7	8	7
Length	1	1	2	1	1	1.4	1	1.4	1	1	1.41	1	1.41	1	1	1	2.23	1	4.12	2.23	2	1.41	1.41	2.23	2.23	1	1	1	1	1.41	1	1.41	1
Direct_Diff	0	0	0	0	-3	-1	1	-1	0	-1	1	-1	-1	0	0	-0.6	0.59	-0.3	-0.3	0.59	-1	0	-0.4	0	-0.6	0	0	0	-1	-1	1	-1	0
Sign	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Segment	1																																
Cum_direct_diff	-9.98																																
Cum_Length %	48.48																																

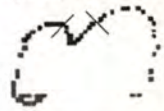
Index	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
Direction	7	1	2	2	2	2	2	1.6	2	2	1.4	1	1	8	8	8	7	7	7.4	6.4	7	7	6.18	5	5
Length	1	1	1	1	1	1.4	1.4	2.2	1	2	3.16	1	3	1	3	1.41	2	1	3.16	2.23	4	2	5	1	1
Direct_Diff	2	1	0	0	0	0	-0	0.4	-0	-0	-0.4	0	-1	0	0	-1	0	0.4	-1	0.27	0	-0.8	-1.2	0	0
Sign	+	+	+	+	+	+	+	+		-	-	-	-	-	-	-	-	-	-	+	+		-	-	-
Segment	2																			4					
Cum_direct_diff	3								-3.58											0.58		-1.99			
Cum_Length %	11.81								23.65											6.65		9.42			

Direct_Diff = Direction difference, Cum_direct_diff = Cumulative direction difference, Cum_Length = Cumulative length





Segment	1	2	3	4	5
Cum direct diff	-9.98	3	-3.58	0.58	-1.99
Cum Length %	48.48	11.81	23.65	6.65	9.42



Segment	1	2	3
Cum direct diff	-9.98	3	-4.99
Cum Length %	48.48	11.81	39.72

Fig. 5. Curvature feature calculation of reverse rotation point and cumulative direction difference with small turn segment reduction.

Index[i]	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
X	350	350	351	352	354	355	356	355	354	353	352	351	350	349	348	348	348	348	349	349	350	351	351	352	353	355	357	358	359	360	361	362	362
Y	361	361	361	361	361	361	361	362	362	363	363	363	362	362	361	360	359	358	356	355	351	349	347	346	345	344	343	343	343	343	343	344	345
X[i+1]-X[i]	0	1	1	2	1	1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	1	0	1	1	0	1	1	2	2	1	1	1	1	1	0	1
Y[i+1]-Y[i]	0	0	0	0	0	0	1	0	1	0	0	-1	0	-1	-1	-1	-1	-2	-1	-4	-2	-2	-1	-1	-1	-1	0	0	0	0	1	1	1
Segment	1																																
	X +								X -									Y +								Y-							
	21								-8									5								-20							
	X + (%)								X - (%)									Y + (%)								Y- (%)							
	72.4137931								-27.5862069									20								80							

Index[i]	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
X	363	363	363	364	365	366	367	368	369	371	372	374	377	378	381	382	384	385	385	385	386	385	384	384	381	380
Y	346	347	348	348	347	346	345	344	343	342	341	340	339	339	339	340	342	343	345	346	349	351	355	357	361	361
X[i+1]-X[i]	0	0	1	1	1	1	1	1	2	1	2	3	1	3	1	2	1	0	0	1	-1	-1	0	-3	-1	
Y[i+1]-Y[i]	1	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	1	2	1	2	1	3	2	4	2	4	0	
Segment	2								3																	
	X +		X -		Y +		Y-		X +				X -				Y +					Y-				
	6		0		2		-5		17				-6				22					-4				
	X + (%)		X - (%)		Y + (%)		Y- (%)		X + (%)				X - (%)				Y + (%)					Y- (%)				
	100		0		28.57		71.42		73.91				26.08				84.61					15.38				

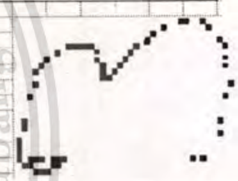


Fig. 6. Calculation of the summation of axis differences.

The summation of difference in X and Y axis in both direction (increasing and decreasing), X+,X-,Y+,Y- are used to characterize the curve. For example if the segment has only X+ and Y+ the segment is written from left to right and top to bottom is show in Fig. 6. The lengths of each segment are another features added to help differentiate character with same curvature and same number of segment. These percentages of length in segment and summation of axis differences reduce the rotation robustness of curvature feature, but they are necessary to distinguish characters with the same curvature.

The character and its summation axis differences is shown in "Fig. 6", "Fig. 7", show the value after normalize of the differences in the X and Y coordinate to distinguish characters with the same curvature, "Fig. 8" shows different writing style can cause the same character to have many different number of segments.

	Character	Segment 1				Segment 2			
		Cum direct diff	%Length	% X+	% Y+	Cum direct diff	%Length	% X+	% Y+
1	ฉ	-0.5842	0.3904	0.7500	0.4167	0.5842	0.6096	0.1026	0.2188
2	ฉ	-0.3686	0.2783	1.0000	0.3750	0.5174	0.7217	0.1795	0.1071
3	ฉ	-0.5529	0.3411	0.6316	0.5000	0.5529	0.6306	0.1633	0.2286
4	ฉ	-0.5013	0.2539	0.5833	0.2000	0.5013	0.7275	0.2250	0.1905
5	ฉ	-0.5842	0.3500	0.6471	0.4286	0.5661	0.6500	0.1739	0.1818
6	ฉ	-0.5013	0.3570	0.6875	0.4286	0.5718	0.6081	0.1750	0.1200
1	ฌ	-0.2921	0.1597	1.0000	0.0000	0.6122	0.8403	0.5500	0.0417
2	ฌ	-0.5529	0.1554	1.0000	0.1111	0.6274	0.8446	0.5362	0.1875
3	ฌ	-0.3399	0.1529	1.0000	0.0000	0.6274	0.8471	0.5373	0.1282
4	ฌ	-0.3686	0.1507	1.0000	0.0000	0.6375	0.8493	0.5000	0.1429
5	ฌ	-0.3686	0.1572	1.0000	0.0000	0.6314	0.8247	0.4630	0.2000
6	ฌ	-0.3686	0.1464	1.0000	0.0000	0.6696	0.8412	0.6000	0.1111

Cum_direct_difff =Cumulative direction differnce

Fig. 7. The use of summation axis differences to distinguish similar segments character.

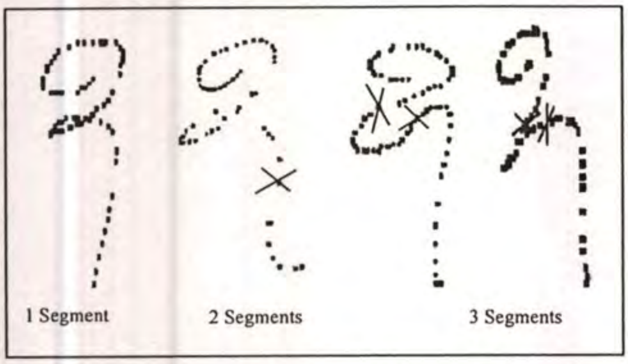


Fig. 8. Show the segments of character that write reference style.

III.RECOGNITION RESULTS

Each character of the same letter is divided into groups that have the same number of segments. In order to recognition, an unknown character number of segments must be determined first, then a template matching is performed for all the character in the training set that has the same number with the unknown character. The closet match will be the recognition character. The number of prototype for each number of segment on average is 268 (max 864, min 5). The recognition time is not quite fast due to the simple feature extraction and comparison time to small number of prototype.

The trained characters set consists of 27 letters with the total of 3755 characters. The test characters are 14183 characters. The number of correct characters recognition is 14183/13420 (94.62%), when compare with the dominant point the recognition is 14183/10647 (75.07%) as show in table2 and "Fig. 9", show recognition comparison of segmentation point between reverse rotation point and dominant point.

TABLE II
COMPARISONS OF RECOGNITION

Method	Characters set	Recognition rate			
		Number characters	Right	wrong	% Recognition
Reverse rotation point	Train	3755	3755	0	100
	Test	14183	13420	763	94.63
Dominant point	Train	3755	3755	0	100
	Test	14183	10647	3536	75.07

IV. CONCLUSION

The curvature and summation axis differences features have been proposed for isolated on-line handwritten recognition. The features is simple and robust for rotation and proportion invariant with high recognition rate.

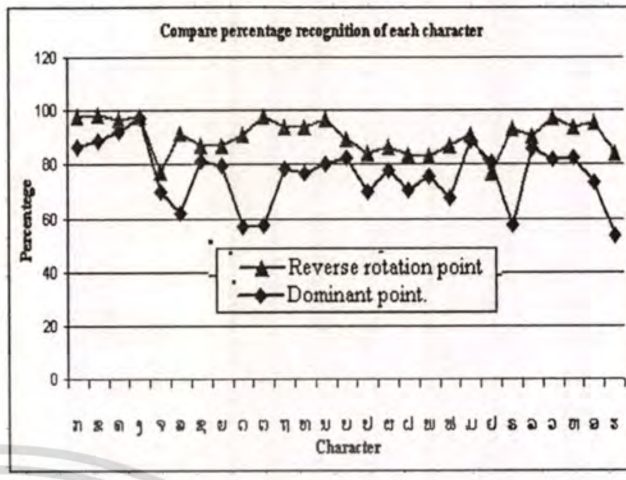


Fig. 9. Comparison recognition of segmentation point between reverse rotation point and dominant point.

REFERENCE

[1] *H. Yuen, A chain coding approach for real-time recognition of on-line handwritten characters, ICASSP'96, Atlanta, USA, 1996.*

[2] *S. R. Veltman and R. Prasad, Hidden Markov models applied to on-line isolated character recognition, IEEE trans. Image Processing, Vol. 3, no. 3, May 1994, pp. 314-318.*

[3] *Y. H. Tay, P.M. Lallican, M. Khalid, C. Viard-Gaudin, S. Kneer, An offline cursive handwritten word recognition system, Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology, TENCON 2001, Vol. 2, 2001, pp. 519 -524*

[4] *C. C. Tappert, C. Y. Suen ans T. Wakahara, The state of art in on-line handwriting recognition, IEEE Transactions on Pattern Analysis and Machin Intelligence, Vol. 12, no. 8, 1990, pp. 787-808.*

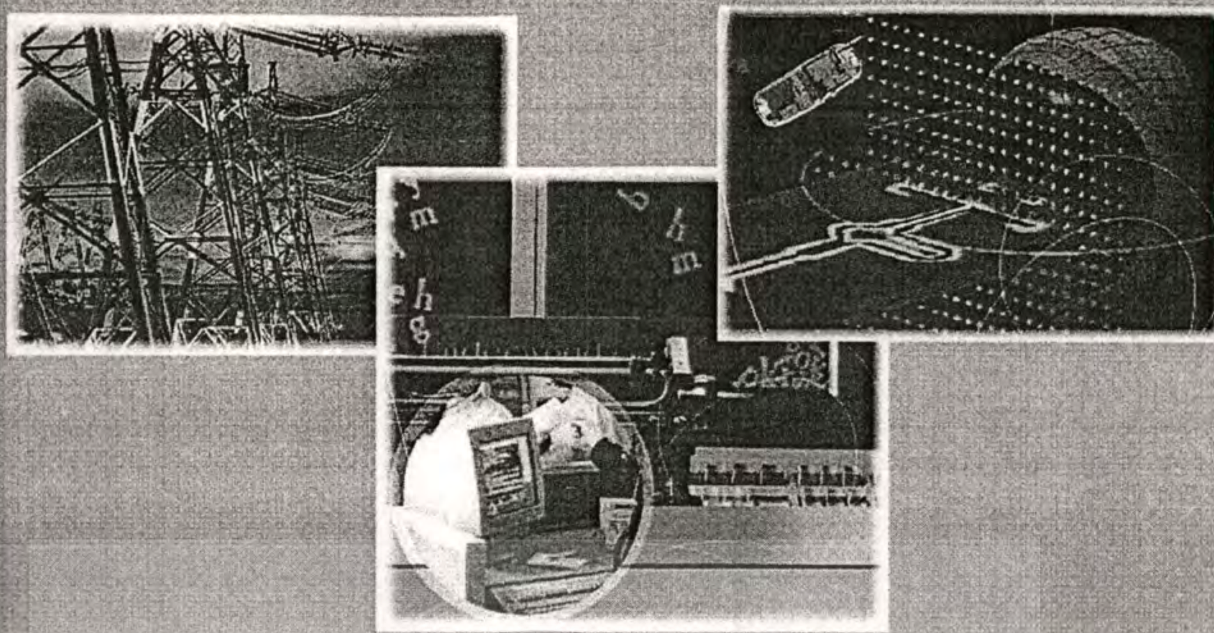
[5] *D. D. Kerrick and A. C. Bovik, Microprocessor-base recognition of handpirnted characters from a talet input, Pattern recognition, Vol. 21, no. 5, 1988, pp. 525-537.*

[6] *K. F. Chan and D. Y. Yeung, Recognition on-line handwritten alphanumberic characters through flexible structural matching, Pattern recognition, Vol. 32, no. 7, 1999, pp. 1099-1114.*

[7] *T. Text, J. B. Olafsdottir and M. Daehlen, Recognition of handwritten symbol, Pattern recognition, Vol. 23, no. 11, 1990, pp.1155-1166.*

[8] *D. J. Burr, Designing a handwriting reader, IEEE Transaction on Pattern Analysis ans Macihine Intelligence, Vol. 5, no. 5, pp. 554-559.*

[9] *J. S. Huang and K. Chuang, Heuristic approach to handwrittern numberal recognition, Pattern Recognition, Vol. 19, no. 1, 1986, pp. 15-19.*

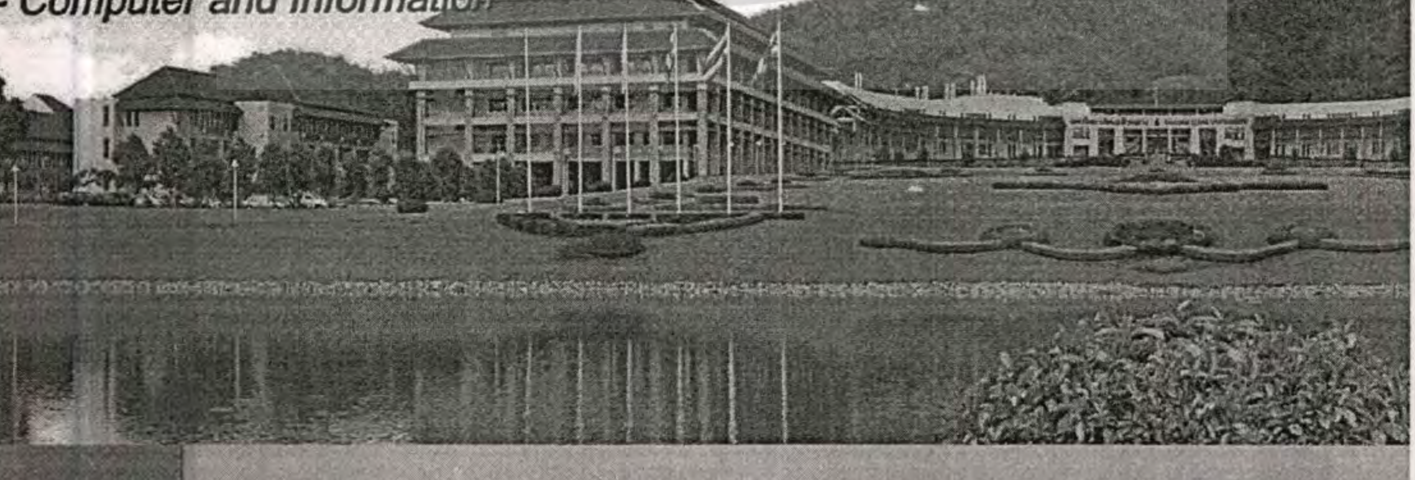


ECTI-CON 2007

Mae Fah Luang University, Chiang Rai, Thailand
May 9-12, 2007

VOLUME 1
- *Circuits and Systems*
- *Control Engineering*
- *Electrical Power Engineering*
- *Other Related Fields*

VOLUME 2
- *Communication Systems*
- *Signal Processing*
- *Computer and Information*



ECTI Association · IEEE THAILAND SECTION · NECTEC a member of NSTDA · WD Western Digital

Curvature Tree for Isolated Unconstrained Lao Handwritten Character Recognition

Khampheth Bounnady*, Boontee Kruatrachue*, Takenobu Matsuura **
*Computer Engineering Department, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang, Bangkok Thailand 10520.
**Communications Engineering Department, Faculty of Information Technology and Electronics,
Tokai University 259-1292, Japan. E-mail: s6060215@kmitl.ac.th, kkboonte@kmitl.ac.th, matsuura@dt.u-tokai.ac.jp

Abstract- **Lao handwritten characters can naturally be represented as a sequence of curve in clockwise and counter clockwise directions. This paper proposed the use of this curve sequence as a main feature in Lao handwritten character recognition system. Since, there can be variation and noise in a handwritten that usually generates small curve segment. Also, some curve segments when written in fast speed can be decreased to small curve segment. Hence, it is hard to distinguish between real curve and noisy curve. To overcome this problem, small curve segment is treated as both real curve and noise curve. The noise curve segment is omitted from the sequence while the real curve is remained. Each uncertain curve segment causes two possible representations. So, instead of single curve sequence, a tree of curve segment is form and use as a main feature in recognition. The proposed method can recognize different handwritten Lao character very well in comparing to the conventional method such as elastic matching.**

Keywords- *feature tree, curvature feature, distinguish feature, handwritten Lao character, Lao recognition.*

I. INTRODUCTION

Many methods for online character recognition have been proposed [1]. However a few methods for Lao character recognition have been proposed [6,7]. In this paper, a new method for unconstrained Lao handwritten recognition is proposed. Fig 1. shows the example of Lao characters and Fig. 2 shows the unconstrained Lao handwritten, where characters are written out of proportion and varied in amount of rotation (slant) and writing style. Most zone features [1] are less effective due to the out of proportion since some characters will have big, small or omitted head.

The proposed method will be compared with elastic matching which provides a very good recognition rate and has been widely used since the early [2] and recently handwritten recognition research [3,4]. The elastic matching can be used to performed matching the unknown writing with the prototype character and the unknown writing is recognized as the character with the closest match.

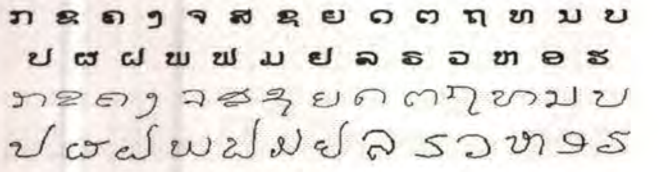


Fig 1. Example of Lao isolated handwritten and printed characters

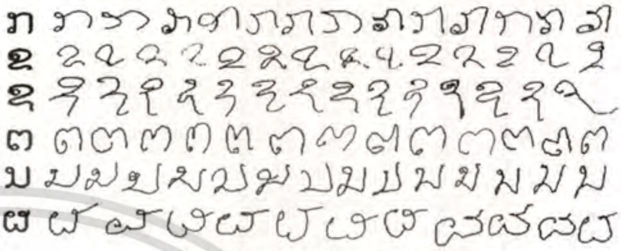


Fig 2. Variation of Lao handwritten.

The elastic matching by finds the optimal alignment distance among all the n points of both matched characters by summing distance of all the matched points. The disadvantage of elastic matching is the overall shape matching of character which has also been observed by other researcher [3]. This is a problem in Lao handwritten as shows in Fig. 3, where the overall shape matches to the wrong character and miss the small curvature in the top. The elastic matching misses dominant point due to equally treatment of all the point in the matching.

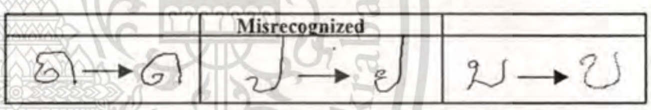


Fig 3. Misrecognized characters by elastic matching method.

Handwritten Lao character is usually continuous drawing of curves (mostly single stroke). It is naturally to represent handwritten Lao as a sequence of curve segment with clockwise and counter clockwise direction as the main feature. Even though this feature quite coincides with the way writers learn to write Fig. 4, it may consist of noisy curves. Usually, the noise curve segment can be distinguished from the real curve by the high amount of curvature of the real signal. But when the written speed is fast, the curvature is reducing close to the noisy curve as shown in Fig. 4. These noisy curves are hard to distinguish with normal noise reduction technique such as Fourier expansion or Fourier descriptor due to the similarity in frequency and amplitude.

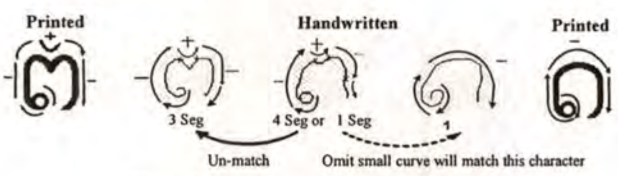


Fig 4. Uncertain segment with small curvature.

In order to successfully use the curvature sequence in the recognition, one may have to identify real curvature from the noisy one. As shown in Fig. 4, the character ຕ consist of three curvatures in sequence. If all small curvature is treated as noise and omitted. It will match to another character ດ that has one clockwise curvature. On the other hand, if treated all curvature as real curve the character will have 4 curvatures and can't match to the correct character with three curvature segments. So in the proposed method, we classify curve feature into three groups, noise, real and uncertain segments. The noise group consists of the noise curves that can be identified and are ignored. The uncertain group consists of noise and real signal that can't be distinguished. The real group has high curvature enough to be certain as a real signal. A binary tree of curve sequence is formed, the branch in the tree represents uncertain curve segment. So the feature of a handwritten character is not just a single sequence of curve but a tree of curve segments. The noisy curves are discarded and the real curves still remain in the tree. In other word, we discard the certain noise out but keep all possibility of the uncertain and real curve segment and form them in the tree. For detail of tree generation will be descript later in section three.

Since all possible segments still remain in the tree, we postpone the determination of the uncertain segment until we compare with another character. So in order to find a similarity (or dissimilarity) with another character, it is liked comparing two trees for all possible representation of curve sequence of both characters. Since this method needs to compare between two characters in order to select the closet representation of the two characters, this technique is suitable for similarity based pattern recognition.

The tree representation guarantees to have one correct sequence of curve, but also contains a lot of other noisy sequences. Hence, more additional curve features are added in order to make noisy curve dissimilar enough to make the correct sequence pair the most similar. In order to show the effectiveness of this approach, the comparison of the recognition results obtained from the proposed method using the curvature tree and the elastic matching are given.

II. CURVATURE SEQUENCE EXTRACTION OF LAO HANDWRITTEN

Lao handwritten is input as a sequence of X,Y coordinates samples from input device with the rate of 100 samples per second. In order to extract curvature sequence, the following steps are performed: 1) Translate the X,Y coordinate sequence into the direction sequence and the length sequence. 2) Translate the direction sequence into the angle sequence by subtracting the current direction from the next direction in the sequence. 3) Find segmentation points and then curvature sequence is defined by adding the angle between consecutive segmentation points.

The direction of pen trajectory is represented as real number call directional code [5]. Examples of the directional code calculation are shown in Fig. 5. The 45-degree means one unit (360-degree is 8 units). The directional code can be calculated

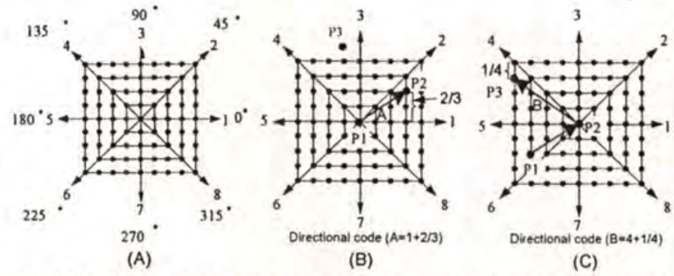


Fig 5.The directional code calculation,(A) The directional code,(B)Directional code calculation of vector A, (C) Directional code calculation of vector B.

as a real number. As shown in Fig. 5, the directional code of vectors A and B are given as 1 + 2/3 and 4+1/4, respectively.

The angle between two consecutive segments of pen trajectory is defined as the difference between the corresponding directional codes, the current directional code minus the previous one. With the above angle, the directions of pen trajectory, counter clockwise and clockwise can be represented as positive and negative angles, respectively. The angle of a curve segment represented as the sum of the angles of the consecutive segments and the length of curve segment is represented as the sum of the length of the segments in the curve segment. It can be seen that the angle for curve segment is rotation invariant and shape invariant. As shown in Fig.6, the directional code for the rotation of any counter clockwise 90 degree turn is 2.0 and the directional code for whole circle 270 turn is 6 independent of exact shape. In the following the whole character is represented as the sequences of curve segments with clockwise and counter clockwise directions.

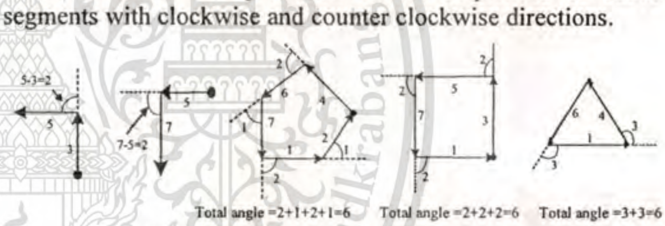


Fig 6. Rotation invariant and shape invariant.

Consecutive segments with the same direction of clockwise or counter clockwise are treated as one angle. A point separating pen trajectory into two curve segments with clockwise and counter clockwise directions is defined here as a segmentation point. In other words, a segmentation point is the point changing the direction of pen trajectory from the clockwise to the counter clockwise and vice versa. Some researchers [5] have used the dominant point or point where the pen trajectory turns the direction quickly. We consider the segmentation points in the pen trajectory of Lao handwritten by improving the above dominance point in the literature [5]. In order to represent Lao handwritten with segmentation points effectively, the noise such as small variation or fluctuation in the pen trajectory should be removed. In this paper the noise cancellation is made as show in Fig. 7(A) and 7(B). When the direction of pen trajectory is not in alignment but in zigzags (clockwise and counter clockwise curve segments with small equal angle), the zigzags is considered as a straight line. In other words, the pen trajectory with the zigzags segments is

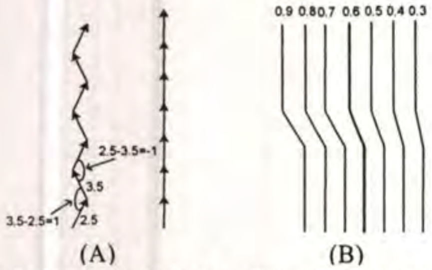


Fig 7. (A) Quantized noise cancellation, (B) The threshold 0.6 curvature

considered as a straight line if sum of the angles of curve segments between the consecutive segmentation points is equal to zero. As show in Fig. 7(B), a pen trajectory has a noise such as small and quick change of curvature. In this case pen trajectory is considered as a straight line or zigzag line according to the amount of the angle of the curve segments between two segmentation points. That point is not considered as the segmentation point if the angle is less than a selected threshold angle. In the example in Fig. 7(B), pen trajectory is considered as straight line if amount of the angle of the curve segments is less than threshold value (0.6), otherwise a zigzag line. It can be seem that the direction of pen trajectory is determined using the angle of curve segment in the trajectory. However some Lao characters such as ດ and ງ have same clockwise direction of pen trajectory. In this case we can not recognize two characters as different using the angle. Thus we introduce the change of the x, y coordinates and the length of the curve segment of pen trajectory as other features of handwritten Lao character.

III. CURVATURE TREE REPRESENTATION

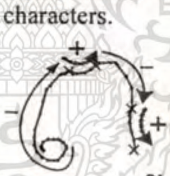
This section explains construction of curvature tree for all possible uncertain segments. The sequence consisting of clockwise and counter clockwise curve segments is used to construct curvature tree using the following steps:

1) Classify each curve segment as one of the following segments, noise, uncertain and real using length and angle of the segment. Then discard the noise segment by merging noise segment with the previous segment and subsequent segment.

2) Re-classify each merged segment as one of the following segments, uncertain and real. If any adjacent uncertain segments has the same amount of angles but different signs (-clockwise,+ counter clockwise), then both uncertain segments are treated as a noises with zero angle and merge both adjacent segments as mentioned above.

Construct binary tree of curve segments consisting of the real and uncertain segments.

The noise of segment can be reduced using the combinations of the angle of the curve segment and the length of the segment. The real curve segment should have high curvature and significant size. Some noise segments may have high curvature with short length or small curvature with long length, but not both. Hence, the noise segment can be classified here using angle-length feature, the angle multiplied by the length (the ratio of the length over the summation of the entire segments length). If the angle-length feature is less than a threshold value (0.017) then the segment is considered as a noise segment. If it is between 0.017 and 0.4, then it is considered as an uncertain segment and furthermore if it is greater than 0.4, then it is considered as a real segment. These threshold values 0.017 and 0.4 got from our handwriting database. Since our database is from 18 writers and we have rotation invariant feature, this threshold should be suitable for other writer. The segment contaminated with noise is removed from the pen trajectory and then the angle and the length of segment are added to both adjacent segments (previous and subsequent segments). After noise removal, the angle and the length of the merged segment change due to the addition. The noise classification of the merged segments is repeated until this classification results in only uncertain and real segments without noise. Up to now the curve sequence has been represented by a sequence of clockwise and counter clockwise curve segments with some uncertain segment. In the next step a binary tree of all possible representation is formed, where each uncertain segment can be treated as a noise or real segment and forms the binary branch of the tree. When uncertain segment is treated as a noise segment, the adjacent segments (three segments) are merged together, as shown in Fig. 8. The figure also shows the tree representation of character ຕ. Each node of tree consists of sum angle and sum length. As mentioned in the introduction, the purpose of feature tree is to keep all the possible representations of a character. At this moment, we still don't know which the real curve sequence is because of some uncertainty. But, the real representation can be identified by comparing with the other characters.



State	S1	S2	S3	S4
Sum angle (SA)	-11.3333	1.1666	-3.2083	0.4519
Sum Length (SL)	224.542	24.581	100.023	54.164
|SA*SL(%)|	6.3098	0.0711	0.7956	0.0606
Segment type	Segment	Uncertain	Segment	Uncertain

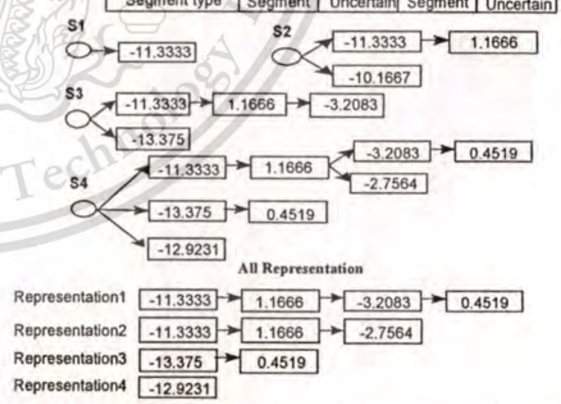


Fig 8. Curvature tree representation and merge of uncertain segment treated as a noise.

IV. RECOGNITION USING PROTOTYPE MATCHING

It is assumed that the character to be recognized has some segments, and the prototypes are prepared in each segment. For the convenience of explanation, it is assumed the character to be recognized has p segments and the number of character classes with p segments, is s_p. Then the character X is represented as

$$X \equiv X_p = (\xi_1, \ldots, \xi s_p)^T \qquad (1)$$

X_p is corresponding to the branch of the alternate curvature tree for the character X then the character X is recognized as follows (Fig. 9)

(Step.1) calculate the distance between prototype P_k and X as

$$d\left(X_p, P_k\right) = \left\| X_p - P_k \right\| \qquad (2)$$

where $d(X_p, P_k)$ is the Euclidean distance between X_p and prototype P_k of the *k*th character class with p segments.

(Step.2) calculate the minimum distance as

$$\underline{d}_{s_p}(r) = \min_{1 \le k \le s_p} d\left(X_p, P_k\right) \qquad (3)$$

(Step.3) character x is recognize as $X \in C_r$ (The *r*th character class)

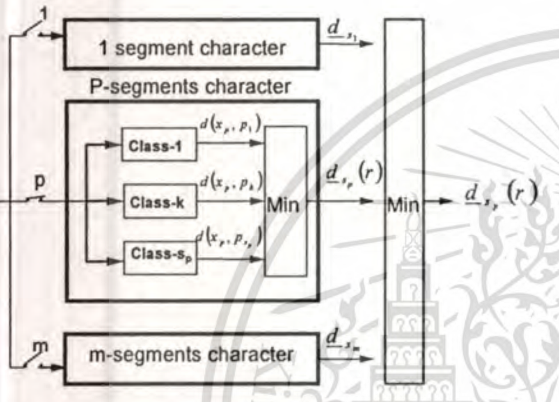


Fig 9. Lao handwritten recognition system

V. EXPERIMENTAL RESULTS

The proposed curvature tree and the elastic matching method are compared. In experiment we used PC with 1.4 GHz Pentium 4 and using our database consisting of 24321 handwritten Lao characters belonging to 27 classes. The prototypes of characters consisting of 3729 handwritten Lao characters. The recognition rates of each method are shown in Table I

TABLE I
COMPARISONS OF RECOGNITION METHODS

Method	Recognition Rate (%)	Average Time (s)
curvature tree	96.78	0.0204
Elastic Matching	94.05	1.2356

Some of characters misclassified by our method and elastic matching method in the experiment are shown in Fig. 10.and Fig.11, respectively. It can be seem from Fig.10 that it is rather difficult for out method to recognize some of characters having small loop or curve, but it easy to recognize curve segment with clockwise or counter clockwise direction in pen trajectory. While it can be also seem from Fig.11 that the elastic matching method can not recognize a curve segment with clockwise or counter clockwise direction in pen trajectory but it is easy to recognize a character with similar shape except for the direction, clockwise or counter clockwise of pen trajectory.

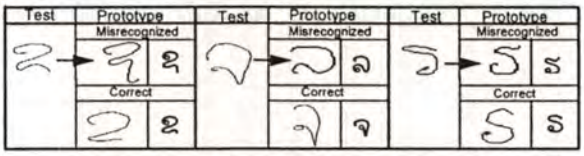


Fig 10. Example of misrecognized in our method.

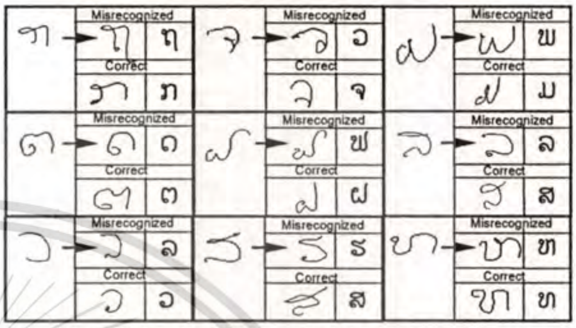


Fig 11. Example of misrecognized in elastic matching.

VI. CONCLUSIONS

This paper proposed an unconstrained isolated Lao handwritten recognition method by using curvature tree. In the proposed method, as unique features of Lao handwritten, different characters with similar shapes having small or large curve segments with clockwise and counter clockwise directions, were take into account. The proposed method can recognize correctly Lao handwritten with above features though the elastic matching method having been used widely can not do this. It was found from our experiments using our data base that the recognition rate of our method was higher than that of elastic matching method and average time requited to the recognition of our method is faster than elastic matching method.

REFERENCES

[1] Rejean Plamondon, Sargur N. Srihari, "On-Line and Off-Line Handwriting Recognition A Comprehensive Survey", IEEE Trans, vol.22, no.1, pp.63-83, January 2000.

[2] D.J Burr, "Designing a Handwriting reader ", IEEE Trans. PAMI vol. PAMI-5, no 5, pp. 555-559, 1983.

[3] Seiishi Uchida, Hiroaki Sakoe, "Eigen deformation for elstic matching base handwritten character recognition", Pattern Recognition, Vol. 36, Issue 9, pp 2031-2040. September 2003.

[4] Claus Bahimann and Hans Burkardt, "The Writer independent Online Handwriting Recognition System frog on hand and cluster Generative Statistical Dynamic Time Warping", IEEE Trans, vol.26, no.3, pp.299-310, March 2004.

[5] H. Yuen, "A chain coding approach for real-time recognition of on-line handwritten characters", ICASSP'96, Atlanta, USA, 1996.

[6] Khampheth Bounnady, Boontee Kruatrachue, Somkiat Wangsiripitak "On-line Lao Handwritten Recognition with Proportional Invariant Feature", WEC'05, April 27-29, 2005, Istanbul, Turkey, pp41-44.

[7] Khingtong Intawongkam and Kosin Chamnongthai, "A method of Hand Writing Lao Alphabet Recognition by Using Back-Propagation Neural Networks", 21th Electrical Engineering Conference (EECON), 1998.

Online Freestyle Handwritten Lao Character Recognition Using Feature Tree

Khampheth BOUNNADY[1], Boontee KRUATRACHUE[1], and Takenobu MATSUURA[2]

[1]Computer Engineering Department, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

[2] Faculty of Information Technology and Electronics, Tokai University, *1117 Kitakaname, Hiratsuka, Kanagawa, 259-1292, Japan*

E-mail: [1]s6060215@kmitl.ac.th, [1]kkboonte@kmitl.ac.th, [2]matsuura@dt.u-tokai.ac.jp

Abstract In this paper, we propose an online writer-independent method of recognizing freestyle handwritten Lao characters using a feature tree. Each path to the leaves of the feature tree represents one of the possible representations of a handwritten character. A character may have more than single representation due to uncertainty (noise) in the input feature. By allowing the multiple representation of handwriting, the proposed method enables us to recognize different handwritten Lao characters having small curve segments in the clockwise or counterclockwise direction, although conventional methods such as the elastic-matching method have difficulty in recognizing them. In the experimental results, the recognition rate was 98.97% and the average recognition time using a PC with a 1.4 GHz Pentium 4 processor was 0.0011 seconds.

Keywords: feature tree, curvature feature, handwritten Lao character recognition, distinguishing feature

1. Introduction

Many methods for online handwriting recognition have been proposed [1]. However, very few methods for recognizing freestyle online handwritten Lao characters have been proposed. Intawongkam proposed a recognition method for the handwritten Lao alphabet using back-propagation neural networks. This method involves offline recognition of the image of the Lao alphabet. A set of lines, curves and circles encoded by a chain code were used as the input of the back-propagation neural networks for recognition. The recognition rate of this method was about 80% [2], [3].

In general, Lao characters are mostly single stroke and consist of a sequence of curves starting with a circular curve called the "head" and continuing with sequences of small and large curves in the clockwise and counterclockwise directions. The difficulties of recognizing handwritten Lao characters are that

1) Many freestyle handwritten Lao characters have similar shapes which small differences in "heading" or the small curves with is not common in other languages, as shown in Fig. 1(b).
2) The head may be omitted in some letters and may vary considerably in size and curvature among the many writing styles. In some characters, the head is the main feature in identifying the characters, as shown in Figs. 2 and 3.
3) It is difficult to distinguish an intentional small curve from noise in a handwritten Lao character since they may have the same amount of curvature. Misidentification of this small curve may result in misrecognition, as shown in Fig. 4.

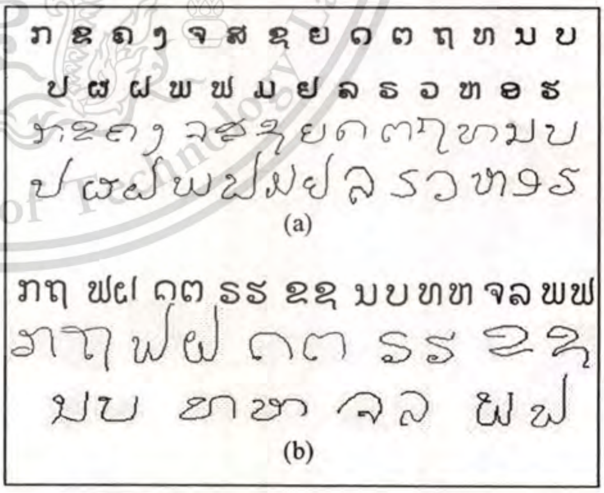


Fig. 1 (a) Printed and handwritten Lao characters. (b) Examples of similarly shaped Lao characters with differences in the head and tail.

In this paper, a new method for writer-independent recognition of freestyle handwritten Lao characters is proposed. Some examples of freely handwritten Lao characters are shown in Fig. 2, where characters are written out of proportion and vary in the amount of rotation (slant) and writing style. Most zone features [1],[4] are less effective due to their being out of proportion since some characters will have a large, small or omitted head and cause shifting in zone.

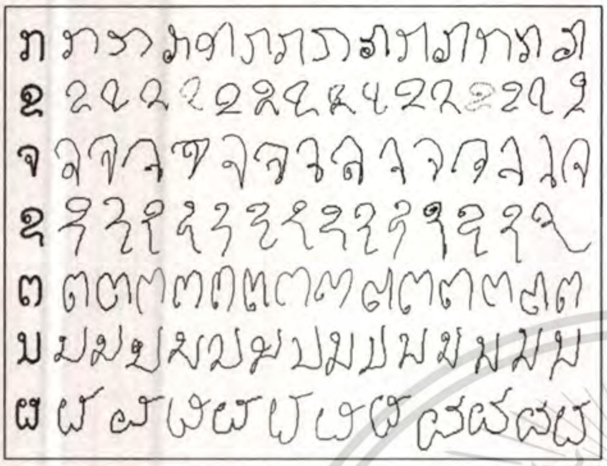


Fig. 2 Examples of freely handwritten Lao characters.

Since a handwritten Lao character is a continuously drawn set of curves, it is natural to represent a handwritten Lao character as a sequence of curve segments in the clockwise and counterclockwise directions as the main feature. Even though this feature coincides with the way children learn to write, the characters may consist of noisy curves, as shown in Fig. 3.

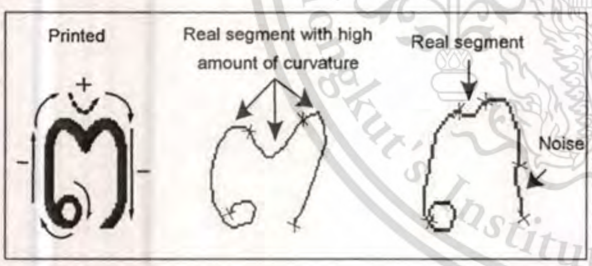


Fig. 3 Lao character and variation in curvature of handwritten Lao character.

Usually, a noisy curve segment can be distinguished from a real curve by the high amount of curvature of the real signal. However, when the written speed is fast, the curvature is reduced and becomes close to that of the noisy curve, as shown in Fig. 3. These noisy curves are difficult to distinguish using a normal noise reduction technique such as a Fourier expansion or Fourier descriptor due to their similarity in frequency and amplitude. To successfully use a sequence of curves in the recognition, one may have to distinguish real curves from a noisy ones. As shown in Fig. 4, the character ຕ consists of three curves in sequence. If all the small curvature segments are treated as noise and omitted, the character will match another character ດ with one clockwise curve. On the other hand, if all the curves are treated as real curves, the character will have 4 curves and it can not match the correct character with three curves segments. In our previous research [3], small curves with curvature less than some threshold were treated as noise. The average recognition rate was 94.63%. The main error was due to prototype characters and test characters having different numbers of curve segments due to noisy, as shown in Fig. 4.

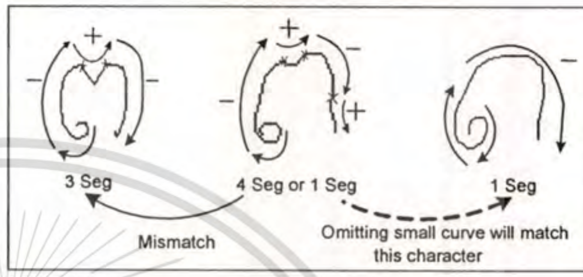


Fig. 4 Uncertain segment with small curvature.

Thus, in the proposed method, we classify curve features into three groups, noise, real and uncertain segments. The noise group consists of the noisy curves that can be identified and removed. The uncertain group consists of noisy and real signals that cannot be distinguished. The real group has a sufficiently high curvature for certain identification as a real signal. A feature tree of curve sequences is formed, and each branch in the tree represents an uncertain curve segment. Thus, the features of handwritten characters are not simply a single sequence of curves but a tree of curve segments. The noisy curves are removed and the real curves still remain in the tree. In other words, we remove the certain noise but keep the possibly real curve segments and use them to create a tree. Since all possible segments remain in the tree, we postpone the determination of an uncertain segment until we can compare it with another character. Thus, to find a similarity (or dissimilarity) with another character, two trees containing all possible representations of the curve sequence of both characters are compared. The tree representation guarantees the existence of a correct curve sequence, but also contains a lot of other noisy sequences. Hence, additional curve features are added to make a noisy curve sufficiently dissimilarity to a real curve to make the correct sequence the most similar. To show the effectiveness of this approach, a comparison of the recognition results obtained from the proposed method using the feature tree and the elastic-matching method is given.

2. Curve Sequence Extraction of Handwritten Lao Characters.

In this section, we explain how to extract the curve segment from the pen trajectory of a handwritten Lao character. The handwritten is input as a sequence of x,y coordinate samples from an input device at a rate of 100 samples per second. To obtain the curve sequence, the following steps are performed.

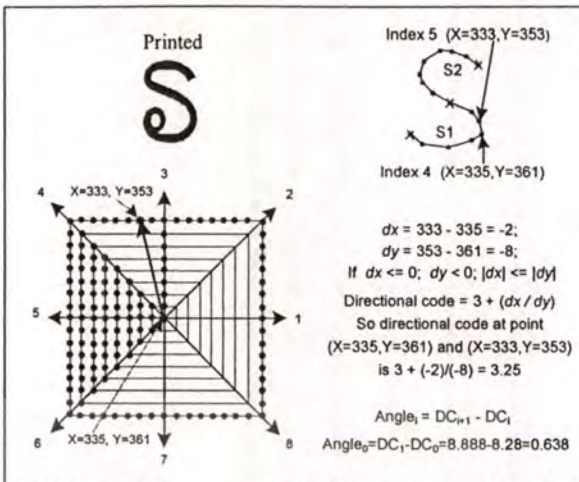
1: Translate the x, y coordinate sequence into a direction sequence and a length sequence.
2: Translate the direction sequence into an angle sequence by subtracting the current direction from the next direction in the sequence.
3: Find segmentation points and then a curve sequence by adding the angles between consecutive segmentation points.

The direction of the pen trajectory is represented as a real number called the directional code (DC) [5]. The DC can be calculated from the displacements in the x and y directions between two sampling points of the pen trajectory, as shown in the following equation:

$$DC=\begin{cases} 1+\dfrac{|dy|}{dx}; & dx>0; dy\le 0; dx\ge |dy| \\ 3-\dfrac{dx}{|dy|}; & dx>0; dy<0; dx<|dy| \\ 3+\dfrac{dx}{dy}; & dx\le 0; dy<0; |dx|\le |dy| \\ 5-\dfrac{dy}{dx}; & dx<0; dy<0; |dx|>|dy| \\ 5+\dfrac{dy}{|dx|}; & dx<0; dy\ge 0; |dx|\ge dy \\ 7-\dfrac{|dx|}{dy}; & dx<0; dy>0; |dx|<dy \\ 7+\dfrac{dx}{dy}; & dx\ge 0; dy>0; dx\le dy \\ 9-\dfrac{dy}{dx}; & dx>0; dy>0; dx>dy \end{cases} \tag{1}$$

where $dx_i=x_{i+1}-x_i$; dx is the difference in the x direction of the sequence.
$dy_i=y_{i+1}-y_i$; dy is the difference in the y direction of the sequence.

Examples of the DC calculation are shown in Fig. 5. An angle of 45 degrees is equal to one unit (360 degrees is 8 units). Figure 5 also shows the raw (x, y) data of the sampling pen trajectory of a handwritten ຣ along with the DC, and its lengths and angles. The angle between two consecutive segments of a pen trajectory is defined as the difference between the corresponding DCs, the current DC minus the previous one. From this angle, the direction of the pen trajectory, whether it is counterclockwise or clockwise, can be represented as a positive or negative angle, respectively, as shown in Fig. 5.



DC=directional code SL=Sum Length SA=Sum Angle

Index	X	Y	DC	Length	Angle	Sign	SA	SL	Segments
0	287	361	8.25	10	0.638	+	4.392	67.753	S1
1	295	367	8.888	18.11	0.377	+			
2	313	369	1.266	15.524	0.304	+			
3	328	365	1.571	8.062	1.678	+			
4	335	361	3.25	8.246	0.583	+			
5	333	353	3.833	7.81	0.809	+			
6	328	347	4.642	14.866	-0.142	-	-4.357	92.254	S2
7	314	342	4.5	13.416	-0.5	-			
8	302	336	4	9.899	-0.6	-			
9	295	329	3.4	5.385	-0.775	-			
10	293	324	2.625	8.544	-0.988	-			
11	296	316	1.636	13.038	-0.761	-			
12	307	309	8.875	8.062	-0.175	-			
13	315	310	8.7	10.44	-0.414	-			
14	325	313	8.285	8.602					
15	332	318							

Fig. 5 Example of the feature extraction of DC, lengths, angles, and segmentation point of handwritten Lao character ຣ.

The angle of a curve consisting of a sequence of segments of the pen trajectory is described as the sum of the angles of the consecutive segments. The length of the curve is represented as the sum of the lengths of the segments in the curve. It can be seen that the angle for the curve is rotation and shape invariant.

As shown in Fig. 6, the DC for any counterclockwise rotation of 90 degrees is 2 and the DC for a 270 degrees turn is 6, independent of the shape. In the following, the whole character is represented as a sequence of curve segments with clockwise and counterclockwise directions. Consecutive segments with the same clockwise or counterclockwise angle are treated as one angle. A point separating a pen trajectory into two curve segments with clockwise and counterclockwise directions is defined here as a segmentation point. In other words, a segmentation point is a point where the direction of the pen trajectory changes from clockwise to the counterclockwise or vice versa. As shown in Fig. 5, point 6 is a segmentation point, where the direction changes from positive to negative. This separates the handwritten ຣ into two consecutive segments. The first one is counterclockwise with a curvature of 4.392, followed by the second one, which is clockwise with a curvature of -4.357.

Some researchers [5] have used a dominant point, a point where the pen trajectory rapidly changes direction,

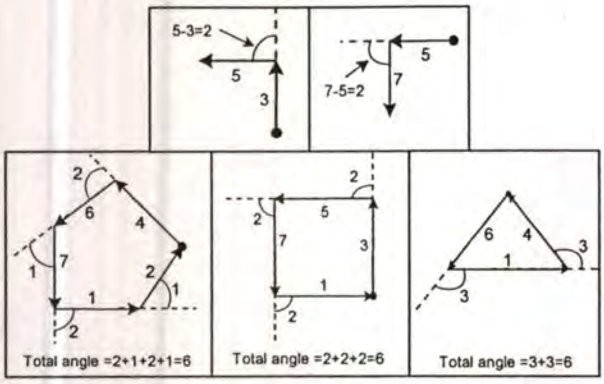
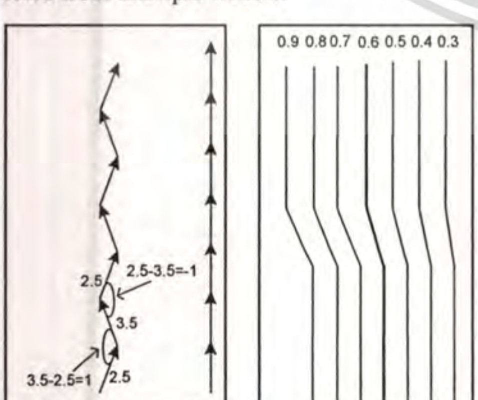


Fig. 6 Rotation-invariant and shape-invariant curves.

as a segmentation point. It was found from our experiment that the use of dominant points as segment points causes greater variation in the number of segments for the same character for different writing styles. To represent handwritten Lao characters with segmentation points effectively, noise such as small variations or fluctuations in the pen trajectory should be removed. In this paper, the noise cancellation is performed as shown in Figs. 7(a) and 7(b).

When the direction of the pen trajectory is not constant but in zigzags (clockwise and counterclockwise curve segments with small equal angles), the zigzags are considered as a straight line. In other words, a pen trajectory with zigzag segment is considered as a straight line if the sum of the angles of the curve segments between the consecutive segmentation points is equal to zero. In Fig. 7(b), the pen trajectory has noise due to small and quick changes in curvature. In this case, the pen trajectory is considered as a straight line or a zigzag line according to the angle of the curve segments between two segmentation points. These points are not considered as segmentation points if the angle is less than a selected threshold angle. In the example in Fig. 7(b), the pen trajectory is considered as a straight line if the angle of the curve segments is less than the threshold value (0.6), otherwise it is a zigzag line. This threshold value of 0.6 was obtained from our handwriting database collected from multiple writers.



(a) (b)

Fig. 7 (a) Quantized noise cancellation. (b) Threshold 0.6 curvature.

It can be seen that the direction of the pen trajectory is determined using the angles of the curve segments in the trajectory. However, some Lao characters have the same counterclockwise direction of the pen trajectory such as ວ and ອ. In this case, we cannot recognize two characters as being different using only the angle. Thus, we introduce the length of the curve segment of the pen trajectory and changes in the x, y coordinates of the character as other features of handwritten Lao characters.

3. Feature Tree Representation

In this section, we explain the construction of alternative curve segment trees for all possible uncertain segments. The sequence consisting of clockwise and counterclockwise curve segments is used to construct a feature tree using the following steps:

1: Classify each curve segment or one of the following segments, noisy, uncertain or real, using the length and angle of the segment. Then, discard the noisy segments by merging them with the previous and subsequent segments.

2: Reclassify each merged segment as one of the following segments, uncertain or real. If any adjacent uncertain segments have the same angle but different signs (- clockwise, + counterclockwise), then both uncertain segments are treated as a noise with a zero angle and are merged with adjacent segments, as mentioned above.

3: Construct a binary tree of curve segments consisting of the real and uncertain segments.

A noisy segment can be detected using the combination of the angle and length of the segment. Real segments should have high curvature and be of significant size. Some noisy segments may have a high curvature with a small length or small curvature with a large length, but not both. Hence, the noisy segments can be classified using the angle-length feature of the angle multiplied by the length (the ratio of the length to the summation of the total length of the segments). If the angle-length feature is less than a threshold value (0.017) then the segment is considered as a noisy segment. If it is between 0.017 and 0.4, then it is considered as an uncertain segment, and if it is greater than 0.4, then it is considered as a real segment. These threshold values 0.017 and 0.4 were obtained from our handwriting database.

The merging of noisy segments is shown in Fig. 8.

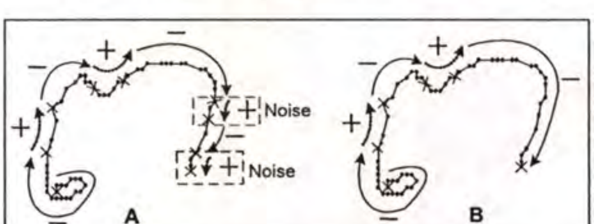


Fig. 8 Merging of noisy segments

The segment contaminated with noise is removed from the pen trajectory, and then the angle and length of the segment are added to both adjacent segments (the previous and subsequent segments). The purpose of the addition is to preserve the length feature since some segments may include a small rotation but the length is still significant in the feature representation of the character.

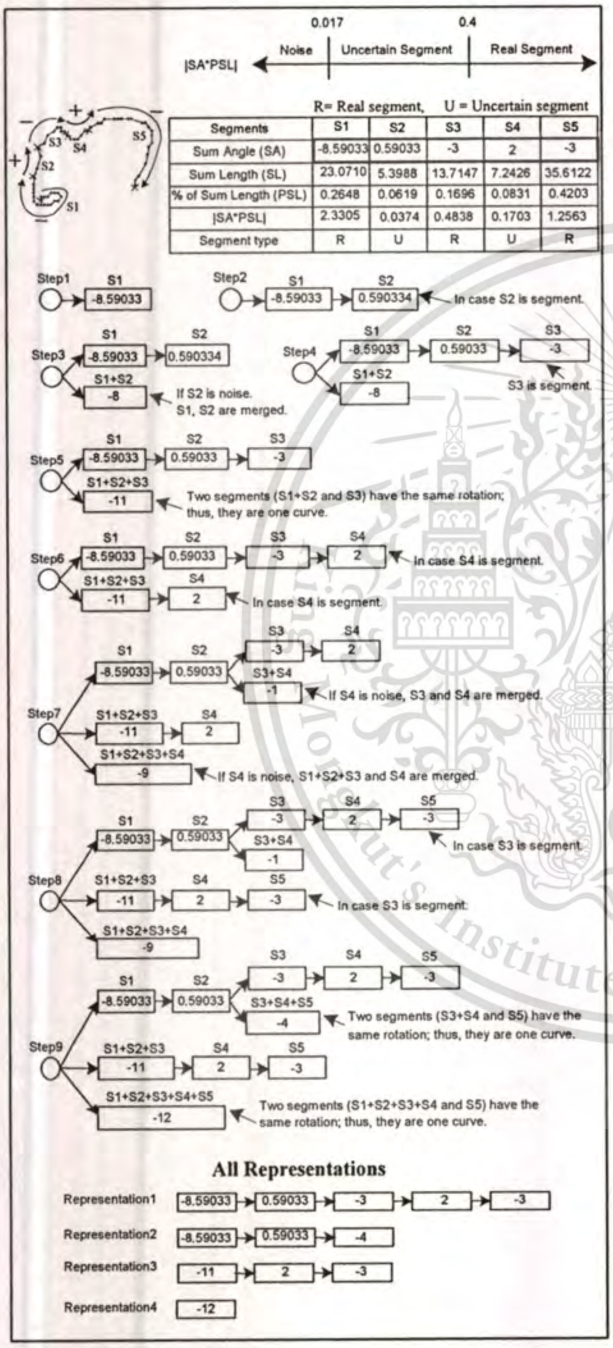


Fig. 9 Formation of tree representation from segment sequence.

After noise removal, the angle and length of the merged segment change upon the addition. The noise classification of the merged segments is repeated until the classification results in only uncertain and real segments without noise. Now, the curve sequence is represented by a sequence of clockwise and counterclockwise curve segments with some segments labeled as uncertain segments. In the next step, a binary tree of all possible representations is formed, where each uncertain segment can be treated as noisy or real and forms a binary branch of the tree. When an uncertain segment is treated as noisy, the adjacent segments (three segments) are merged together, as shown in Fig. 9.

The figure also shows the tree representation of a handwritten character ถ. As mentioned in the introduction, the purpose of the feature tree is to keep all the possible representations of a character. At this moment, we still do not know which is the real curve sequence because of some uncertainty. However, the real representation can be identified by comparison with the other characters.

4. Recognition Using Prototype Matching

It is assumed that the character to be recognized has some segments, and that prototypes are prepared for each segment. For the convenience of explanation, it is assumed that the character to be recognized has p segments and that the number of character classes with p segments is s_p. Then, the character X is represented as

$$X \equiv X_p = (\xi_1, \ldots, \xi s_p)^T \tag{2}$$

X_p corresponds to a branch of the alternative curve tree for character X; then character X is recognized as follows (Fig. 10).

(Step 1) Calculate the distance between prototype P_k and X as

$$d(X_p, P_k) = \left\| X_p - P_k \right\| \tag{3}$$

where $d(X_p, P_k)$ is the Euclidean distance between X_p and prototype P_k of the kth character class with p segments.

(Step 2) Calculate the minimum distance as

$$\underline{d}_{s_p}(r) = \min_{1 \le k \le s_p} d(X_p, P_k) \tag{4}$$

(Step 3) Character X is recognized as $X \in C_r$ (the rth character class).

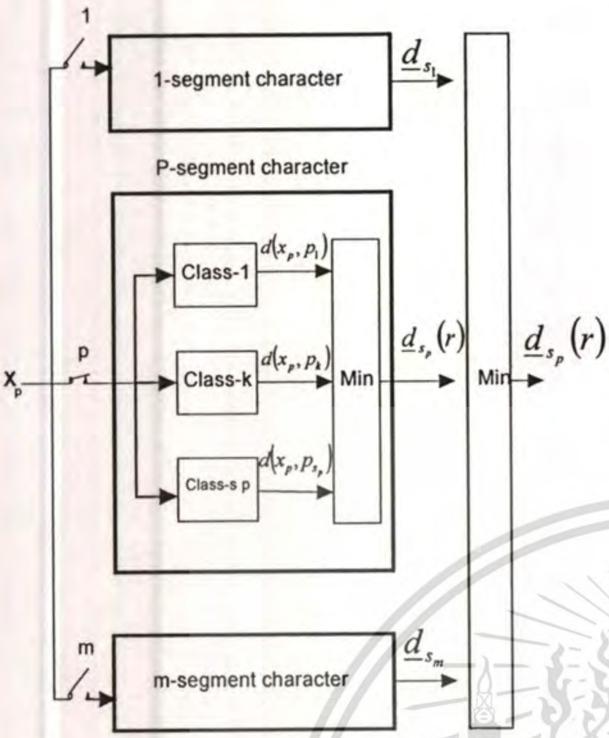


Fig. 10 Handwritten Lao character recognition system.

5. Prototype Selection

The prototype selection used the maximum-cover-first method [6], as shown in Fig. 11. For the convenience of explanation, we define data points A1, A2 and A3 to be of class A, and B1 and B2 to be of class B. All data points are in two-dimensional space (x, y).

1. The distance list for each data from all the data points is calculated and sorted from closest to furthest. This list is used to identify the candidate prototypes. For example, for A3, the distance list is A3 B1 B2 A2 A1; A3 is a reference point. This suggests that only A3 itself must be used as a prototype for correctly recognizing A3, since A3 is the only A closer (more similar) to A3 than B1. Otherwise, A3 may be misclassified as B1 if B1 is selected to be a prototype.
2. The cover list for each data point is constructed from the distance lists (cover list 1). If A3 is selected, it can also be used as a prototype to correctly classify A1 and A2, since A1 is closer to A3 than it is to B1 (the closest other class of A1). A2 is also closer to A3 than it is to B1. Hence, A1, A2 and A3 are all covered by A3. In other words, if A3 is selected as a prototype, A1, A2 and A3 will be closer prototypes than any other non-A prototype.
3. Select the prototype with the maximum number of data points in the cover list (list A3, which covers 3 data points A1, A2 and A3).
4. Remove the data points (A1, A2 and A3) in the selected prototype cover list (list A3) from the entire cover list. The cover list after removing prototypes A1, A2 and A3 is shown in cover list 2. Since A3 is already selected to be one of the prototypes, data points A1, A2 and A3 can be classified correctly as prototype A3.
5. Repeat steps 3 and 4 until the entire cover list is empty. (Select B1 and remove B1 and B2).

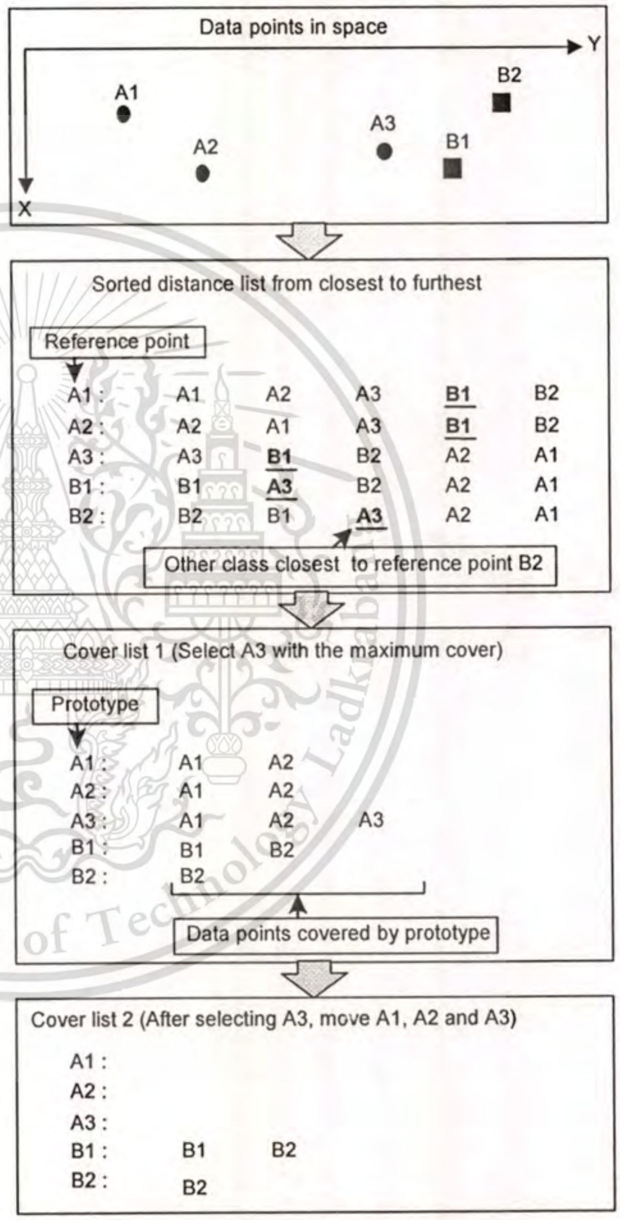


Fig. 11 Prototype selection.

The selection algorithm cannot generate the minimum number of prototypes nor the "best" prototype with the highest rate of recognition of the unseen data (test data). However, it will generate a set of prototypes that will 100% correctly classify the data used for prototype selection. Hence, our data can be separated into two groups namely, training and testing data. The prototypes are selected from the training data and are used in the recognition of the test data.

6. Elastic Matching

The proposed method will be compared with elastic matching, which provides a very good recognition rate and has been widely used in early [7] and recent handwritten recognition research [8]. Elastic matching can be used to match an unknown character with the prototype characters, and choose the closest match as the recognized character. The elastic matching finds the optimal alignment using the Euclidian distance among all the n points of both matched characters by the summation distance of all the matched points along the Viterbi path. This requires $O(n^2)$ computational cost, proportional to the Viterbi matrix size [9]. Even though the Viterbi search reduces the computational cost from exponential to $O(n^2)$, the cost is still too high to perform real-time recognition, particularly for a large number of prototype characters.

For the convenience of explanation, it is assumed here that the location of a handwritten Lao character is normalized. The x and y components of the handwritten Lao character are denoted here as x(t) and y(t), respectively.

Define the complex sequences, $\{F(t_i)\}$ and $\{G(t_i)\}$, as

$$F(t_i) = X_F(t_i) + jY_F(t_i), \quad i = 1,2,\ldots,I$$

and

$$G(t_i) = X_G(t_i) + jY_G(t_i), \quad i = 1,2,\ldots,J, \tag{5}$$

where $j \equiv \sqrt{-1}$

Then the dissimilarity between two characters F and G is defined as

$$D(F,G) = \frac{1}{I+J} \min_P \left[\sum_{k=1}^{K} d(p_k)\Delta_k \right] \tag{6}$$

where

$$F = [F(t_1),\ldots,F(t_I)], G = [G(t_1),\ldots,G(t_J)]$$
$$P_k = (i_k, j_k), \quad 1 \le i_k \le I, \quad 1 \le j_k \le J$$
$$d(P_k) = d(i_k, j_k) \equiv \left\| F(t_{i_k}) - G(t_{j_k}) \right\|$$
$$\Delta_k = (i_k - i_{k-1}) + (j_k - j_{k-1}), \quad i_0 = j_0 = 0.$$

The dissimilarity between F and G can be calculated by dynamic programming as follows:

Step 1: Define

$$h(p_k) = \min_{p_{k-1}} \left[h(p_{k-1}) + d(p_k)\Delta_k \right] \tag{7}$$

where

$$h(p_1) = d(p_1)\Delta_1 = 2d(p_1)$$
$$h(p_k) = \min \begin{pmatrix} h(i_{k-1}, j_{k-1}) + 2d(i_k, j_k) \\ h(i_{k-1}, j_k) + d(i_k, j_k) \\ h(i_k, j_{k-1}) + d(i_k, j_k) \end{pmatrix}.$$

Step 2: Calculate the dissimilarity between two characters F and G:

$$D(F,G) = \frac{1}{I+J} h(p_K) \tag{8}$$

The pattern G is recognized as

$$G \in class \;\; F \;\; if \;\; D(F,G) \le \eta_F$$
$$G \notin class \;\; F \;\; if \;\; D(F,G) > \eta_F \tag{9}$$

η is the threshold value determined from the training set.

The main problem in using elastic matching for handwritten Lao characters is shown in Fig. 12. Elastic matching matches the overall shape of a character resulting in incorrect character matches. It misses the small dominant curvature at the top due to its equal treatment of all the points in the matching.

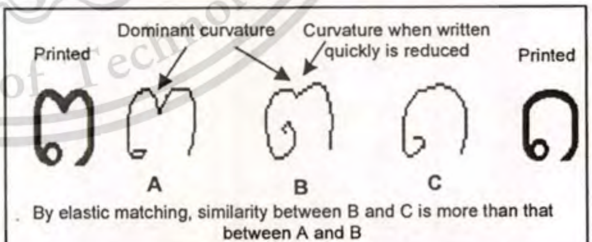


Fig. 12 Problem of elastic matching in recognition of handwritten Lao character.

7. Experimental Results

In this section, the proposed feature tree, the elastic-matching method and the dominant points are compared using our database consisting of 19,7481 handwritten Lao characters belonging to 27 classes from 18 writers. In the dominant-point method, the segmentation points are the dominant points where the pen trajectory changes direction quickly. The feature in each segment is the same as in the feature tree except that there are only noisy and real segments without uncertain segments. Hence, there is no tree, simply a list of consecutive segments. The prototypes of characters were determined from the prototype selection algorithm in Sect. 5. We randomly partitioned the data into 5 sets of training and test data, as shown in Table 1.

Table 1. Recognition result for handwritten Lao characters.

Method	Data Set	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Data Set 5	Average
	Test Data	15356	15357	15357	15357	15357	15357
	Training Data	4125	4124	4124	4124	4124	4124
Feature tree	Prototypes	325	326	329	327	319	325
	Recognition (%)	99.08	98.87	98.94	99.10	98.88	98.974
	Average Time (s)	0.00119	0.00071	0.00107	0.00098	0.00152	0.00110
Elastic matching	Prototypes	434	403	379	406	401	405
	Recognition (%)	96.31	96.62	96.48	96.60	96.37	96.476
	Average Time (s)	0.05780	0.05372	0.05125	0.05398	0.05404	0.05416
Dominant point	Prototypes	325	326	329	327	319	325
	Recognition (%)	89.61	90.05	89.01	90.53	87.98	89.442
	Average Time (s)	0.00226	0.00210	0.00220	0.00210	0.00240	0.00221

The average recognition rates were 98.97% for our method, 96.47% for the elastic-matching method and 89.44% for the dominant-point method. Although the 2.5% difference in the recognition rates of the proposed method and the elastic matching seems small, the number of errors is 3.5 times lower using the proposed method (1 error for every 100 characters versus 3.5 errors). The average recognition time per sample using a PC with a 1.4 GHz Pentium 4 processor was 0.00110 seconds for our method, 0.05416 seconds for the elastic-matching method and 0.00221 seconds for the dominant-point method. Table 2 shows the number of identification errors of each method.

Table 2. Comparison of number of recognition errors.

Method	Data set 1	2	3	4	5
Test data	15356	15357	15357	15357	15357
Feature tree	140	173	162	137	171
Elastic matching	558	518	540	521	557
Dominant point	1562	1517	1678	1445	1783

Some of the characters misclassified by our method and the elastic-matching method in the experiment are shown in Figs. 13 and 14, respectively. It can be seen from Fig. 13 that it is rather difficult for our method to recognize some of the characters having the same curve, but it can easily to recognize curve segments with a clockwise or counterclockwise pen trajectory.

Examples in Fig. 14 show where the same overall shape results in matching the wrong character by missing the small distinguishing curves. The elastic matching misses dominant points due to its equal treatment of all the points in the matching. In the elastic-matching method it is difficult to recognize small curve segments with a clockwise or counterclockwise pen trajectory, but it can easily to recognize a character with a similar shape except for a clockwise or counterclockwise pen trajectory.

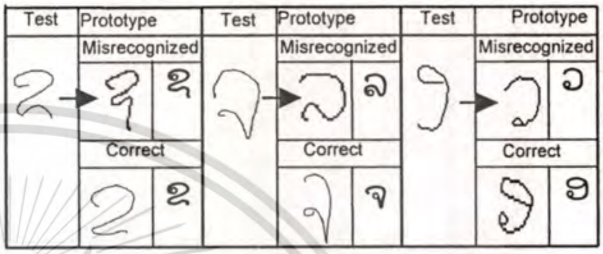


Fig. 13 Examples of misrecognition in our method.

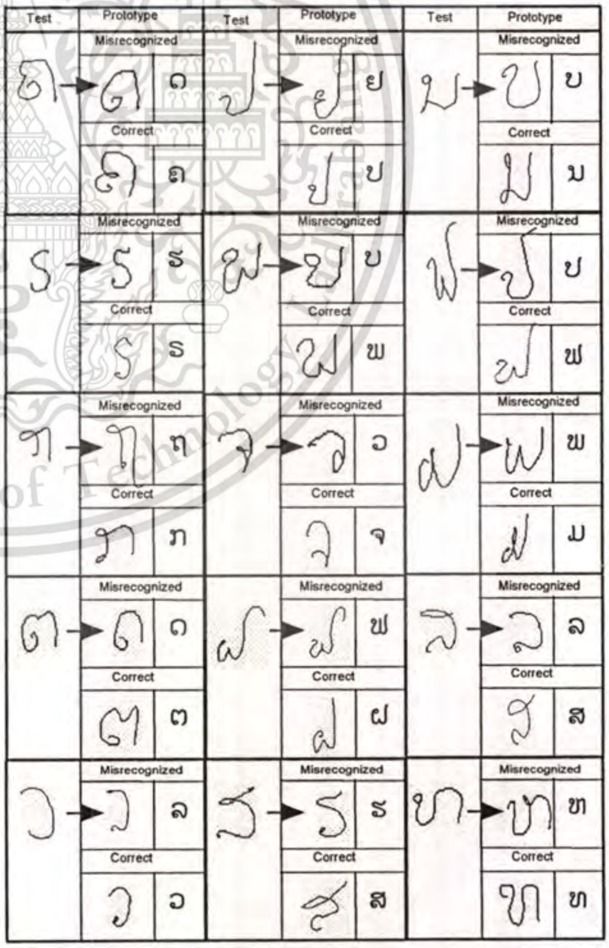


Fig. 14 Examples of misrecognition in elastic matching.

8. Conclusions

In this paper, we proposed a method of recognizing freestyle writer-independent handwritten Lao characters using a feature tree. In the proposed method, the unique features of handwritten Lao characters, such as different characters with similar shapes having small or large curve segments in the clockwise or counterclockwise direction, were taken into account. The proposed feature tree can systematically handle the intrinsic ambiguities found in handwritten Lao characters. The present method can correctly recognize handwritten Lao characters with the above features, although the widely used elastic-matching method cannot do this. It was found from experiments using our database that the recognition rate of our method was higher than that of the elastic-matching method and that the time required for recognition was much less than that using the elastic-matching method.

Acknowledgments

The authors thank JICA for supporting this work under the AUN/SEED-Net Project.

References

[1] R. Plamondon and S. N. Srihari: On-line and off-line handwriting recognition: a comprehensive survey, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, pp. 63-84, 2000.

[2] K. Intawongkam and K. Chamnongthai: A method of hand writing Lao alphabet recognition by using back-propagation neural networks, Proceeding of 21st Electrical Engineering Conference (EECON), pp. 421-424, 1998.

[3] K. Bounnady, B. Kruatrachue and S. Wangsiripitak: On-line Lao handwritten recognition with proportional invariant feature, Proceeding of The 3rd World Enformatika Conference (WEC'05), pp. 41-44, 2005.

[4] S. Impedovo, M. G. Lucchese, and G. Pirlo: Optimal zoning design by genetic algorithms, IEEE Trans. Systems, Man and Cybernetics, Vol. 36, No. 5, pp. 833-846, 2006,.

[5] H. Yuen: A chain coding approach for real-time recognition of on-line handwriting characters, Proceeding of International Conference On Acoustics, Speech and Signal Processing (ICASSP'96), pp. 3426 – 3429, 1996.

[6] B. V. Dasarathy: Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design, IEEE Trans. Systems, Man and Cybernetics, Vol. 24, No. 3, pp. 511-517, 1994.

[7] D. J. Burr: Designing a handwriting reader, IEEE Trans. Pattern Analysis and Machine Intelligence Vol. 5, No 5, pp. 555-559, 1983.

[8] N. Joshi, G. Sita, A. G. Ramakrishnan, and S. Madhvanath: Comparison of elastic matching algorithms for online Tamil handwritten character recognition, Proceeding of 9th International Workshop on Frontiers in Handwriting Recognition, (IWFHR-9. 2004), pp. 444 – 449, 2004.

[9] L. R. Rabiner and B. H. Juang: Fundamentals of Speech Recognition, Prentice-Hall PTR, 1993.

Khampheth Bounnady received his B.E. and M.E. degrees in Computer Engineering from King Mongkut's Institute of Technology Ladkrabang, Thailand, in 2001 and 2003 respectively. He is currently a doctoral student under an AUN/SEED-Net scholarship to study Electrical Engineering at King Mongkut's Institute of Technology Ladkrabang. He is also a Lecturer at the Department of Electronics Engineering, National University of Laos. His research interests include on-line handwriting recognition and data mining.

Boontee Kruatrachue received his B.S. degree in Electrical Engineering from Kasetsart University, Thailand, in 1981, and his M.S. and Ph.D degrees in Electrical Engineering from Oregon State University, USA, in 1984 and 1987, respectively. During 1988-1990, he was a Software Engineer at Astronautics Corporation of America, Wisconsin, USA. He is now an Associate Professor at the Computer Engineering Department, King Mongkut's Institute of Technology Ladkrabang, Thailand. His research interests include pattern recognition, data mining and machine learning.

Takenobu Matsuura received his M.E. and D.E. degrees in Electrical Engineering from Tokai University, Hiratsuka, Japan, in 1976 and 1983, respectively. From 1988 to 1989, he was a Visiting Professor at the ESE Department, the University of Essex, Colchester, United Kingdom. He is currently a Professor at the Department of Communications Engineering, Tokai University. His research interests include pattern recognition, image processing and signal processing.

Appendix (B)

Lao character set

For Lao characters there are consisting of alphabet 33 characters, vowel 28 characters, special vowel 4 characters, and 4 tonal marks. For alphabet 33 characters divide to single alphabet and pair alphabet. Single alphabet there are 27 characters and pair alphabet there are 6 characters. As shown in the Fig below.

ກ ຂ ຄ ງ ຈ ສ ຊ ຍ ດ ຕ ຖ ທ ນ ບ

ປ ຜ ຝ ພ ຟ ມ ຢ ລ ຣ ວ ຫ ອ ຮ

Show single alphabet of Lao characters

ຫງ ຫຍ ຫນ ຫມ ຫຼ or ຫລ ຫວ

Show pair alphabet of Lao characters

Xະ Xາ Xິ Xີ Xຶ Xື Xຸ Xູ ເXະ ເX ແXະ ແX ໂXະ ໂX

ເXາະ Xໍ ເXິ ເXີ ເXັຍ ເXຍ ເXຶອ ເXືອ Xົວະ Xົວ

Show vowel of Lao characters

ໄX ໃX ເXົາ Xຳ

Show special vowel of Lao characters

X່ X້ X໊ X໋

Show tonal mark of Lao characters

Appendix (C)

Recognition result of unconstraint Thai handwritten characters

We also experimented with the Thai handwritten character as shown in the Fig 1. Thai handwritten characters consist of 16,703 Thai handwritten characters belonging to 37 classes from 26 writers. From 16,703 characters we randomly partitioned data into 5 sets of training and testing data as shown in Table 1.

Fig 1 Thai handwritten character

Table 1 Data set of Thai handwritten character used in experiments.

Data	Number Test set	Number Train set
Data set 1	10677	6026
Data set 2	10677	6026
Data set 3	10678	6025
Data set 4	10678	6025
Data set 5	10679	6024

We used training set 1-5 to train for the prototypes. We got the prototypes of Thai handwritten character of each method is shown in the Table 2.

1. Recognition results of Thai handwritten character

From our experiment by using prototypes above and testing with test data set 1-5, we get the recognition rate and recognition time of each methods of Thai handwritten character as shown in the Table 2. And Table 3 shows the number of Thai handwritten character error of each method in each test data set 1-5.

Table 2 Recognition result

Methods	Data set	Set 1	Set 2	Set 3	Set 4	Set 5	Average
	Test	10677	10677	10678	10678	10679	10677.8
	Train Data	6026	6026	6025	6025	6024	6025.2
Proposed method	Prototype	687	808	798	685	681	731.80
	Recognition (%)	98.192	98.127	98.043	98.202	98.305	98.174
	Average Time (s)	0.004863	0.005215	0.005523	0.004925	0.005106	0.005126
Elastic matching	Prototype	782	809	810	778	787	793.20
	Recognition (%)	95.036	94.980	94.934	95.111	94.700	94.952
	Average Time (s)	0.22138	0.22129	0.22747	0.22757	0.22550	0.22464
Single representation	Prototype	1006	1065	810	1006	1005	978.4
	Recognition (%)	92.19	92.51	93.32	92.19	92.68	92.578
	Average Time (s)	0.00346	0.00357	0.00354	0.00349	0.00341	0.00350
Dominant point	Prototype	687	808	798	685	681	731.8
	Recognition (%)	69.77	72.34	72.07	70.01	68.92	70.622
	Average Time (s)	0.00522	0.00504	0.00516	0.00523	0.00557	0.00524

Table 3 Comparison number of handwritten error.

Method	Data set				
	Set 1	Set 2	Set 3	Set 4	Set 5
Tests data	10677	10677	10678	10678	10679
Proposed method	193	200	209	192	181
Elastic matching	530	536	541	522	566
Single representation	832	790	712	832	780
Dominant point	3209	2935	2971	3184	3307

In the Fig 2 shows comparison average recognition rate of Thai handwritten character from 5 sets in each character of each method. We experimented on test data set 1-5 as shown below.

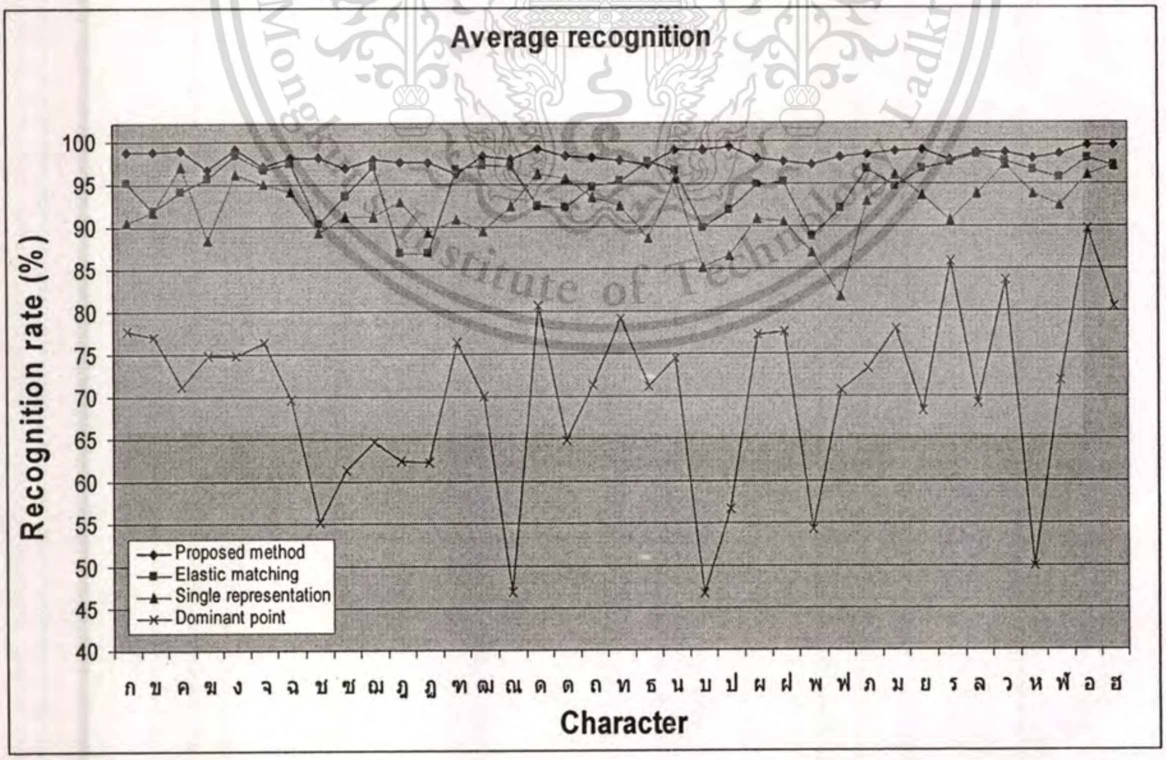


Fig 2 Average recognition rate of Thai handwritten character from 5 data sets.

And in the Table 4 to Table 7 shows the right recognition and wrong recognition of Thai handwritten character in each method. The highlight is wrong character.

Table 4 Confusing matrix of Thai handwritten character recognition of the proposed method (5 sets)

Number character	Tests	ก	ข	ฅ	ฃ	ง	จ	ฉ	ช	ซ	ฌ	ฎ	ฏ	ฑ	ฒ	ณ	ค	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ	ภ	ม	ย	ร	ล	ว	ห	ฬ	อ	ฮ
1605	ก	1585	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	4	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
891	ข	0	879	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	18	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1932	ฅ	10	0	1911	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0
1464	ฃ	0	0	0	1415	0	0	0	0	7	9	0	0	5	14	3	0	0	0	0	0	0	1	0	0	0	2	0	0	8	0	0	0	0	0	0	0	0
1793	ง	0	3	0	0	1776	6	0	3	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0
1550	จ	0	2	0	0	0	1503	6	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	16	0	14	1	0	0
1690	ฉ	0	0	0	1	0	10	1658	0	0	0	3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	7	0	0	8	0	0
1431	ช	0	8	0	0	0	0	0	1402	4	0	0	0	0	0	0	0	0	0	0	2	0	0	6	0	0	0	0	0	0	0	0	0	0	6	3	0	0
1495	ซ	0	0	0	7	0	0	0	12	1448	0	3	2	6	1	0	0	0	0	2	4	0	1	0	0	2	0	5	0	0	0	0	0	0	0	0	0	0
1401	ฌ	0	0	0	20	0	0	0	0	0	1372	0	0	0	4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
1807	ฎ	1	0	0	0	0	0	0	1	0	0	1761	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2
1379	ฏ	0	0	0	0	0	0	0	0	0	0	30	1346	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
1340	ฑ	0	0	0	3	2	0	0	0	22	2	0	0	1288	0	1	0	5	0	15	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0
1587	ฒ	0	4	0	10	0	0	2	0	0	2	0	0	0	1559	0	0	0	0	0	0	0	1	0	0	0	6	0	0	3	0	0	0	0	0	0	0	0
1537	ณ	2	0	0	1	0	0	0	0	3	13	0	0	3	0	1504	0	3	1	0	0	5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1676	ค	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1660	3	1	6	0	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1737	ต	0	0	0	2	0	0	0	0	0	0	0	0	1	0	2	3	1707	1	11	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
1540	ถ	13	0	0	0	0	0	0	0	0	1	0	0	0	0	2	3	2	1509	7	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0
1497	ท	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	16	4	11	1463	0	4	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
1463	ธ	0	0	3	0	0	0	0	4	0	1	0	0	1	0	0	0	0	1	0	1421	1	3	1	0	0	0	2	2	0	0	22	0	0	0	0	0	1
1587	น	0	0	0	0	0	0	4	0	0	0	0	0	0	0	2	0	0	0	3	0	1570	0	6	1	0	0	2	0	0	0	0	0	0	0	0	0	0
749	บ	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	741	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
961	ป	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	954	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1076	ผ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	1053	13	0	0	0	0	3	2	0	0	0	0	0	2
1086	ฝ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	1060	0	0	0	0	9	0	0	0	0	0	0	0
669	พ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	650	11	0	4	0	0	0	0	0	2	0	0
806	ฟ	0	1	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	12	790	0	0	0	0	0	0	0	0	0	0
1686	ภ	3	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0	3	0	0	1659	0	0	11	0	0	0	0	0	0
1526	ม	0	0	0	3	5	0	0	0	0	0	0	0	0	0	0	1	4	0	0	0	1	0	1	0	0	1	4	0	1506	0	0	0	0	0	0	0	0
1477	ย	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	9	3	0	0	0	0	1462	0	0	0	0	0	1	0
1417	ร	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	6	1	0	0	8	0	4	1385	0	0	0	0	0	2
1366	ล	0	0	0	0	1	11	5	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1346	0	0	1	0	0
1615	ว	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	1591	0	0	21	0
1565	ห	0	0	0	0	0	15	0	1	0	0	0	0	0	0	0	0	0	0	7	0	0	0	1	0	0	0	0	1	0	0	0	2	0	1531	0	0	0
1637	ฬ	0	0	0	0	0	3	0	0	4	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2	2	0	0	0	0	0	0	3	1611	0	0
1581	อ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	2	0	1573	0
1770	ฮ	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	2	1760

Table 5 Confusing matrix of Thai handwritten character recognition of the elastic matching (5 sets)

Number character	Tests	ก	ข	ฅ	ฃ	ง	จ	ฉ	ช	ซ	ฌ	ฎ	ฏ	ฑ	ฒ	ณ	ค	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ	ภ	ม	ย	ร	ล	ว	ห	ฬ	อ	ฮ
1605	ก	1527	0	0	1	0	0	0	0	0	0	0	0	0	4	4	0	0	45	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0
891	ข	0	818	0	6	0	0	0	13	5	0	0	0	0	0	0	0	0	0	0	0	0	42	3	0	0	0	0	0	2	3	0	0	0	0	0	0	0
1932	ฅ	0	0	1819	0	0	0	0	0	0	0	0	0	0	0	0	92	13	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1464	ฃ	0	2	0	1401	0	0	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	55	0	0	0	0	0	0	0	0
1793	ง	0	0	0	0	1764	0	0	0	0	0	1	27	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1550	จ	0	0	0	0	0	1499	15	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	20	0	0	8	0
1690	ฉ	0	0	0	0	0	24	1644	1	0	0	0	0	0	0	0	0	0	0	0	0	8	0	3	0	0	0	3	0	2	0	0	0	0	0	0	0	0
1431	ช	0	8	0	0	0	0	0	1293	117	0	0	0	0	0	0	0	0	0	0	8	0	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1495	ซ	0	1	0	2	0	0	0	88	1399	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	2	0	0
1401	ฌ	0	0	0	0	0	0	0	0	0	1359	0	0	0	8	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1807	ฎ	1	2	0	0	0	0	0	0	2	0	1569	233	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1379	ฏ	0	0	0	0	0	0	0	0	0	0	181	1198	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1340	ฑ	0	0	0	0	0	0	0	0	3	0	0	0	1295	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
1587	ฒ	0	0	0	17	0	0	0	0	0	21	0	0	0	1542	0	0	4	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
1537	ณ	0	0	0	1	0	0	0	0	0	44	0	0	0	0	1490	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1676	ค	0	0	44	0	0	0	0	0	0	0	0	0	0	0	0	1547	81	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1737	ต	0	0	9	0	0	0	0	0	0	0	0	0	2	0	1	110	1602	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1540	ถ	34	0	0	0	0	0	0	0	0	1	0	0	0	0	0	6	8	1457	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0	0	0	0
1497	ท	0	0	1	0	0	0	0	1	0	0	0	0	44	1	0	1	0	4	1427	0	0	0	0	0	0	2	0	2	0	0	0	0	0	14	0	0	0
1463	ธ	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	1428	0	0	1	0	0	0	0	0	0	0	12	0	0	0	2	0	14
1587	น	0	2	0	0	0	0	2	2	1	0	0	0	0	0	0	0	0	0	0	0	1532	13	5	0	1	2	17	0	8	0	0	0	0	0	2	0	0
749	บ	0	24	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	672	28	0	0	10	4	0	8	0	0	0	0	0	0	0	0
961	ป	0	0	0	0	0	0	0	9	3	0	0	0	0	0	0	0	0	0	0	0	0	31	882	0	2	0	30	0	0	0	0	0	0	0	2	0	2
1076	ผ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	1019	39	6	1	0	0	8	0	0	0	0	0	0	0
1086	ฝ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	32	1032	0	13	0	0	0	0	0	0	0	0	0	0
669	พ	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	15	0	6	2	594	41	0	6	0	0	0	0	2	0	0	0
806	ฟ	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	2	29	1	6	10	742	0	3	0	0	0	0	0	9	0	0
1686	ภ	20	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	0	0	1629	0	0	0	0	0	0	0	0	0
1526	ม	0	2	0	37	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	8	2	0	0	7	21	0	1444	0	0	0	0	0	0	0	0
1477	ย	0	5	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	7	2	21	9	0	2	0	0	1428	0	0	0	0	0	1	0
1417	ร	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0	0	1	0	0	1379	0	0	0	0	0	3
1366	ล	0	0	0	0	0	11	1	0	0	1	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1344	2	0	0	0	0
1615	ว	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	6	0	0	0	0	0	0	1	0	1574	0	0	11	0
1565	ห	0	1	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	5	37	0	0	0	0	0	0	1	0	4	0	0	1	0	0	1511	0	0	0
1637	ฬ	0	0	0	0	0	0	0	10	10	0	0	0	0	0	0	0	0	0	0	0	2	0	4	0	0	9	33	0	0	0	0	0	0	2	1565	0	2
1581	อ	0	0	0	0	0	5	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	2	0	0	0	0	0	0	0	21	0	0	1546	0
354	ฮ	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	30	0	0	2	0	0	0	0	0	0	0	12	0	0	2	1	1	1718

Table 6 Confusing matrix of Thai handwritten character recognition of single representation (5 sets).

Number		Prototypes																																			
character	I	ก	ข	ค	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ	ภ	ม	ย	ร	ล	ว	ห	ฬ	อ	ฮ
1605	ก	1453	0	13	0	0	0	0	0	0	1	0	0	8	0	3	6	15	58	6	4	0	0	0	0	0	5	1	29	0	0	3	0	0	0	0	0	0
891	ข	0	816	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	36	18	0	0	0	0	0	4	0	1	0	0	0	2	0	0
1932	ค	17	0	1875	0	0	0	0	0	0	0	0	1	0	0	0	3	2	1	0	2	0	0	0	2	0	0	0	21	0	0	3	0	0	0	0	5	0
1464	ง	0	0	0	1295	0	0	0	1	15	62	3	0	11	25	10	0	0	3	1	1	3	0	0	0	0	7	4	0	22	0	0	0	0	0	1	0	0
1793	จ	2	4	2	4	1725	4	2	1	0	4	0	0	0	4	3	4	5	3	0	0	4	2	8	0	0	0	2	4	2	0	0	2	0	2	0	0	0
1550	ฉ	0	0	2	0	4	1472	14	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	38	0	8	2	4	0
1690	ช	0	0	0	4	0	14	1591	0	1	1	3	2	1	3	0	0	3	2	2	0	15	0	2	0	0	2	3	2	3	0	0	20	0	2	11	3	0
1431	ซ	0	16	0	3	0	0	0	1278	78	2	2	6	0	0	0	0	0	0	0	12	2	3	20	0	0	0	0	0	0	0	3	2	0	1	3	0	0
1495	ฌ	0	0	0	19	0	0	0	46	1358	3	0	0	22	2	0	0	0	0	0	6	0	0	4	0	0	2	13	0	3	0	0	0	0	0	13	0	0
1401	ญ	0	0	0	44	0	0	1	0	1	1276	0	0	2	16	35	0	0	4	0	1	2	0	0	3	0	7	0	0	7	0	0	0	0	2	0	0	0
1807	ฎ	0	0	0	0	0	0	0	0	0	0	1678	97	1	1	0	0	0	0	0	0	0	0	0	6	1	0	0	7	0	0	4	0	0	0	1	2	8
1379	ฏ	0	0	2	0	0	0	0	2	1	0	118	1224	0	0	0	0	2	0	0	10	0	0	0	0	0	4	0	0	0	0	0	0	0	2	2	0	4
1340	ฑ	0	0	0	21	0	0	0	0	35	4	0	0	1216	2	3	6	11	0	36	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	2	0	0
1587	ฒ	0	5	0	61	0	0	2	2	2	29	2	0	1	1418	14	0	5	0	0	9	6	1	0	0	1	3	2	0	21	0	1	0	0	0	2	0	0
1537	ณ	0	0	0	3	0	0	0	0	3	35	1	2	6	5	1420	0	4	12	2	1	38	0	0	0	0	0	1	0	2	0	0	0	0	2	0	0	0
1676	ด	4	3	2	0	6	3	0	0	0	1	0	0	2	0	0	1610	10	11	18	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0
1737	ต	0	0	0	4	1	0	0	0	4	0	0	0	5	5	5	13	1660	12	10	0	1	0	2	0	0	2	1	0	8	0	0	0	0	4	0	0	0
1540	ถ	27	0	3	0	0	0	2	0	0	3	0	0	2	0	9	29	17	1439	7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1497	ท	3	0	2	0	0	0	0	5	0	0	0	0	23	0	0	24	5	3	1382	0	6	0	4	0	0	25	4	4	0	0	0	0	0	7	0	0	0
1463	ธ	0	0	6	11	0	0	0	35	7	2	1	14	6	2	0	0	2	0	1	1297	2	2	2	0	0	7	2	11	1	2	40	1	0	0	2	0	7
1587	น	0	5	0	0	1	0	7	0	0	0	0	0	0	7	12	0	2	0	9	0	1514	2	1	0	0	11	1	0	10	0	0	0	0	0	0	0	0
749	บ	0	26	0	0	2	0	0	3	0	2	0	0	0	0	0	0	0	0	0	1	0	637	47	0	3	10	13	0	5	0	0	0	0	0	0	0	0
961	ป	0	24	0	0	0	4	0	11	2	2	0	0	0	1	1	0	0	0	5	1	4	57	831	0	0	3	10	0	5	0	0	0	0	0	0	0	0
1076	ผ	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	977	69	4	0	1	0	13	10	0	0	0	0	0	0
1086	ฝ	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	2	0	0	0	68	982	0	1	3	0	16	6	0	0	0	0	1	4
669	พ	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	1	6	2	2	2	1	581	48	0	13	0	0	0	0	0	2	0	0
806	ฟ	2	3	0	0	0	0	0	0	4	2	0	1	3	3	2	0	0	0	3	0	2	9	5	0	0	89	658	0	6	1	0	0	0	0	13	0	0
1686	ภ	17	0	31	0	0	0	0	0	0	0	9	2	0	0	0	0	0	0	0	6	3	0	0	11	3	0	0	1565	0	4	27	0	0	6	0	0	2
1526	ม	0	9	0	9	0	0	0	0	2	6	0	0	0	1	0	0	7	0	0	0	4	7	1	0	0	11	3	0	1464	0	0	2	0	0	0	0	0
1477	ย	0	0	0	0	0	0	0	2	1	0	9	0	0	0	0	0	0	0	0	0	0	0	0	20	32	4	0	3	0	1383	2	0	0	3	0	3	15
1417	ร	2	0	6	0	0	0	0	0	0	0	4	2	0	0	0	0	0	0	0	65	0	0	0	12	6	0	0	23	0	7	1285	0	0	1	0	0	4
1366	ล	0	2	0	2	0	37	17	0	3	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1281	0	7	15	0	0
1615	ว	0	0	2	0	0	15	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	1572	0	0	21	2
1565	ห	0	2	2	0	0	30	5	3	0	0	2	0	0	0	1	2	11	0	3	0	3	0	1	0	0	1	1	3	1	0	2	8	1	1466	13	0	4
1637	ฬ	0	0	0	4	0	7	4	5	19	0	3	0	3	3	0	0	3	0	0	3	0	2	3	0	1	12	35	0	5	0	0	6	0	6	1516	0	0
1581	อ	0	0	2	0	0	3	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	4	0	0	16	5	4	1518	8
1770	ฮ	0	0	0	0	0	4	0	0	0	0	12	5	0	0	0	0	0	0	0	2	0	0	0	1	4	0	0	0	0	9	4	0	2	4	5	1	1717

Table 7 Confusing matrix of Thai handwritten character recognition of the dominant point (5 sets)

Number		Prototypes																																			
character	I	ก	ข	ค	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ	ภ	ม	ย	ร	ล	ว	ห	ฬ	อ	ฮ
1605	ก	1245	0	74	30	0	3	0	4	15	13	11	22	2	0	3	7	13	59	18	19	1	3	1	12	1	17	9	13	9	7	0	0	0	1	0	0	0
891	ข	1	687	2	6	3	0	0	32	1	2	0	0	0	6	0	3	3	0	22	7	0	86	15	1	0	0	3	0	3	0	0	2	0	0	6	0	0
1932	ค	26	16	1375	0	3	33	7	16	16	9	39	23	0	0	0	47	20	4	4	18	0	2	43	8	3	7	0	68	2	9	11	0	0	112	0	0	6
1464	ง	0	0	0	1095	0	0	7	9	8	126	3	3	7	58	30	13	11	6	16	2	8	0	0	0	0	6	5	0	44	0	0	3	0	2	2	0	0
1793	จ	0	31	5	7	1341	13	106	7	35	19	4	2	9	41	30	21	12	10	10	2	27	7	6	0	0	0	0	0	6	0	0	31	0	4	7	0	0
1550	ฉ	0	10	5	0	0	1183	35	7	5	0	2	2	0	0	0	7	3	1	34	14	0	0	7	0	0	0	4	20	4	0	0	92	16	48	2	9	0
1690	ช	0	1	0	16	1	31	1173	5	16	20	7	0	24	4	41	0	9	2	14	1	90	1	4	0	4	0	46	0	21	9	0	36	0	1	53	6	0
1431	ซ	2	195	1	10	2	5	4	788	120	17	5	5	11	3	15	0	13	6	19	26	9	36	16	0	1	5	1	1	5	3	1	3	0	31	12	0	0
1495	ฌ	1	33	0	14	0	4	50	150	909	26	5	9	108	17	23	2	2	5	3	9	6	10	15	0	1	4	21	0	1	8	3	2	0	16	30	0	0
1401	ญ	4	0	0	124	0	0	7	6	12	906	7	4	8	188	91	3	9	4	3	0	1	0	0	0	0	4	4	0	11	0	0	0	0	5	3	0	0
1807	ฎ	0	6	19	2	0	5	3	4	2	3	1124	367	0	2	3	0	0	4	0	10	0	7	0	24	7	5	18	56	22	24	33	3	0	16	10	2	23
1379	ฏ	0	0	4	4	0	0	2	0	3	8	406	832	0	1	1	0	0	0	0	11	1	0	0	5	0	0	2	15	0	18	7	0	0	14	3	9	6
1340	ฑ	0	0	4	23	0	0	2	2	37	17	3	3	1021	34	27	8	9	2	54	9	3	4	0	0	0	23	21	0	2	0	0	3	0	10	16	0	0
1587	ฒ	0	0	0	101	3	0	3	4	7	136	4	13	11	1106	85	12	48	7	13	0	0	0	0	0	2	0	3	6	5	0	0	2	0	3	8	0	1
1537	ณ	9	0	0	96	2	0	8	5	16	277	3	1	56	99	720	5	104	22	6	4	81	0	0	2	2	5	2	0	7	0	0	0	0	1	4	0	0
1676	ด	0	0	21	16	5	0	0	0	1	36	0	1	18	15	24	1351	47	38	76	0	0	0	6	0	0	4	0	0	17	0	0	0	0	0	0	0	0
1737	ต	3	6	3	30	0	0	1	5	3	55	0	0	20	106	115	32	1124	72	78	3	4	0	0	0	0	25	2	0	40	0	0	2	0	2	3	0	0
1540	ถ	26	0	4	25	2	0	2	1	4	56	1	2	18	38	21	72	61	1099	67	5	0	3	3	0	0	11	6	4	4	0	0	0	0	3	2	0	0
1497	ท	4	2	2	5	2	3	0	3	0	3	4	0	62	2	0	82	6	12	1185	0	2	4	0	0	0	51	6	0	1	0	0	1	0	55	0	0	0
1463	ธ	0	1	3	7	0	10	10	26	18	8	3	17	4	4	3	2	11	7	3	1041	0	3	19	9	28	0	2	5	2	13	129	3	0	5	33	7	27
1587	น	3	2	5	9	1	0	11	6	3	10	0	0	20	5	57	4	5	4	64	10	1181	3	5	0	0	20	24	0	129	0	0	3	0	1	2	0	0
749	บ	5	87	1	0	4	0	0	8	0	0	0	0	10	2	0	0	5	0	1	12	6	350	143	6	0	30	48	3	19	0	2	0	0	4	3	0	0
961	ป	0	58	8	4	0	23	2	66	12	3	0	0	4	0	2	2	9	0	13	14	3	104	544	0	1	9	40	0	3	0	2	15	0	5	11	2	0
1076	ผ	0	0	0	0	0	0	0	0	1	0	4	17	0	0	0	0	0	0	0	8	0	0	0	831	158	0	0	4	0	20	9	0	1	4	4	1	14
1086	ฝ	0	0	0	0	0	0	2	0	0	0	0	16	0	0	0	0	0	0	0	7	0	0	0	138	841	0	3	7	0	30	9	0	2	2	0	7	21
669	พ	3	3	0	6	0	0	1	3	2	1	0	0	6	5	2	4	3	0	29	3	4	12	2	4	0	363	179	0	17	0	1	0	0	6	8	0	2
806	ฟ	0	0	0	10	0	3	5	7	9	4	0	0	3	6	1	0	3	0	13	0	8	1	41	2	3	70	569	0	10	2	1	6	0	2	27	0	0
1686	ภ	2	0	29	0	0	2	1	6	1	0	52	52	0	0	0	0	7	2	0	57	0	7	0	31	6	3	0	1231	0	24	76	3	4	80	2	0	5
1526	ม	0	15	0	14	0	0	2	14	4	16	0	0	3	11	20	24	19	0	19	1	70	4	0	0	0	29	61	0	1180	0	0	0	0	1	11	0	0
1477	ย	0	0	8	0	0	0	1	0	1	0	22	66	0	0	0	0	0	0	0	26	0	1	0	73	80	0	0	5	0	1004	40	0	4	15	1	8	119
1417	ร	0	1	19	0	0	0	0	0	0	1	7	20	0	0	0	0	0	0	0	41	0	0	0	11	7	2	0	31	0	7	1215	0	6	4	0	8	37
1366	ล	0	9	0	3	10	207	45	5	1	12	4	0	3	6	4	1	2	0	4	4	11	2	3	0	0	0	7	2	15	0	1	945	0	18	39	3	0
1615	ว	0	0	0	0	0	21	0	0	0	0	4	9	0	0	0	0	0	0	0	7	0	0	0	0	2	0	0	7	0	2	10	0	1349	2	0	189	13
1565	ห	0	19	61	1	0	56	4	40	22	7	12	44	9	5	10	5	0	3	172	31	10	11	10	9	0	16	11	53	2	0	23	13	0	783	74	10	39
1637	ฬ	0	0	1	24	0	9	7	5	50	8	12	4	9	9	7	0	6	0	0	56	2	1	0	3	12	19	120	29	2	3	11	5	0	39	1175	2	7
1581	อ	0	0	2	2	0	0	0	1	0	0	2	24	0	0	0	0	0	0	0	6	0	0	0	2	29	0	2	3	0	8	6	0	58	2	3	1414	14
1770	ฮ	0	0	0	0	0	0	0	0	2	0	4	42	0	0	0	0	3	0	0	37	0	0	0	23	23	0	0	8	0	30	66	0	36	8	2	61	1427

2. Analysis of recognition results

The misrecognition examples of Thai characters are shown in the Fig 3. It can be seen that is rather difficult for the proposed method to recognize some of characters having same curve. But it is easy to recognize curve segment with clockwise or counter clockwise direction. In the table 8 and table 9 is shown recognition of character has the same of shape but different in tail is short and long.

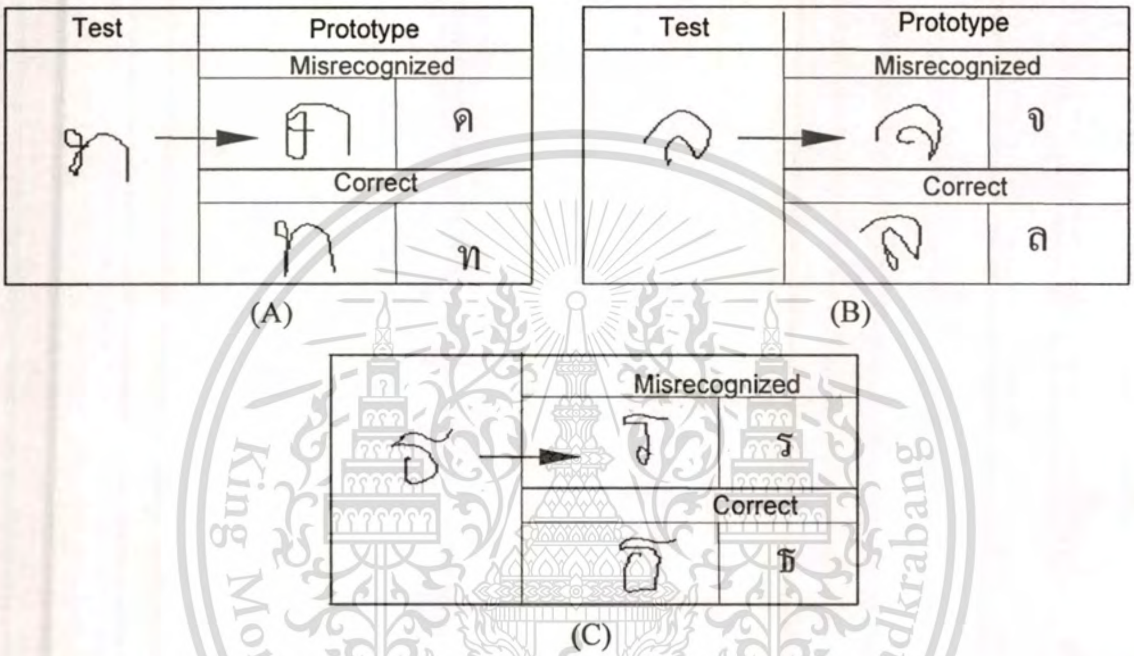


Fig 3 Misrecognition examples of Thai handwritten characters.

Also it can be seen from Fig 4 that misrecognition of character in the elastic matching method is due to same shape but different direction. In Fig 5 shows misrecognition of elastic matching due to misrecognition of dominant curve.

Table 8 The proposed method

Tests		prototype					
		บ	ป	ผ	ฝ	พ	ฟ
749	บ	741	2	0	0	0	0
961	ป	5	954	0	0	0	0
1076	ผ	0	0	1053	13	0	0
1086	ฝ	0	0	17	1060	0	0
669	พ	0	1	0	0	650	11
806	ฟ	0	0	0	0	12	790

Table 9 Elastic matching method

Tests		prototype					
		บ	ป	ผ	ฝ	พ	ฟ
749	บ	672	28	0	0	10	4
961	ป	31	882	0	2	0	30
1076	ผ	2	1	1019	39	6	1
1086	ฝ	0	9	32	1032	0	13
669	พ	15	0	6	2	594	41
806	ฟ	2	29	1	6	10	742

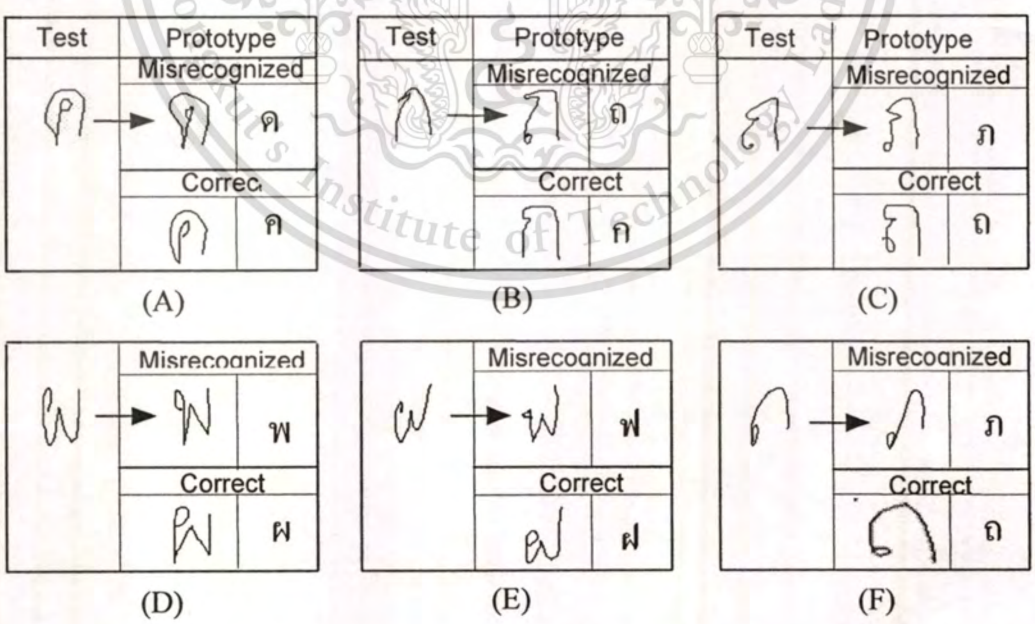


Fig 4 Elastic matching misrecognition examples of Thai characters with the same shape but difference direction.

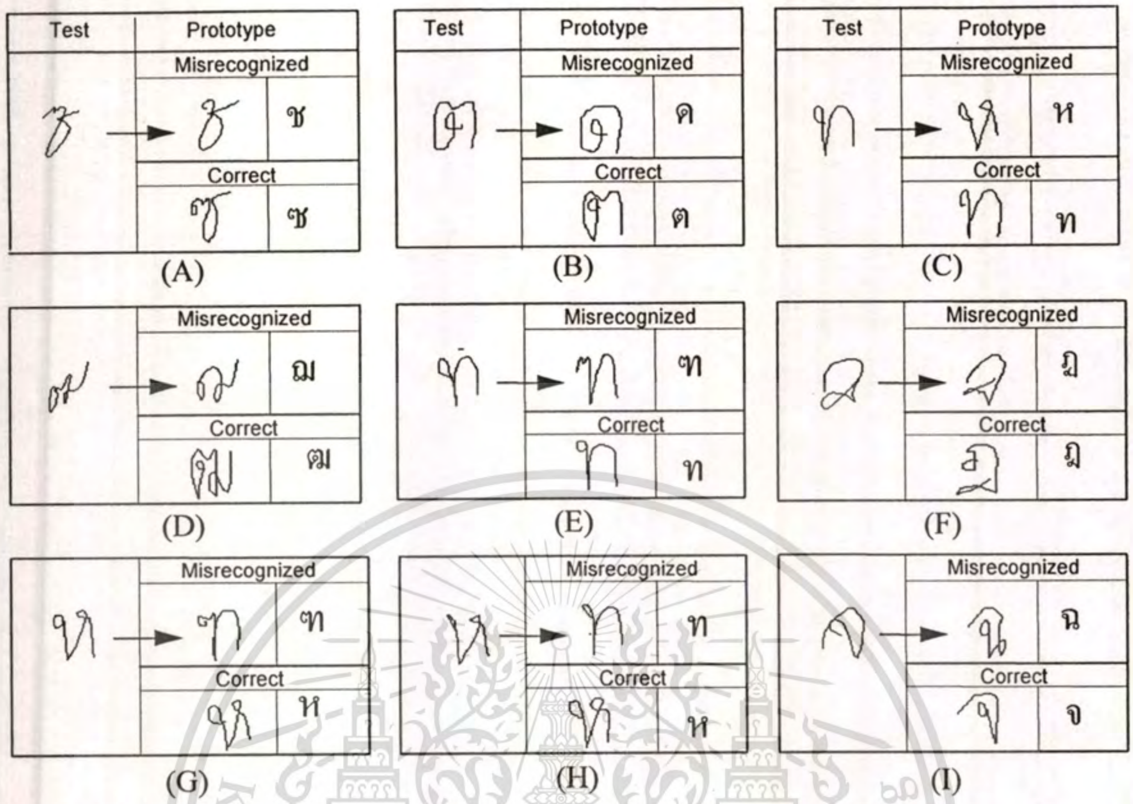


Fig 5 Misrecognitions in elastic matching method due to misrecognition of dominant curve

The misrecognition by single representation method is shown in the Fig 6 and in the Fig 7. For Thai handwritten characters, it misses some paths or misses some real curve segments due to fixed threshold.

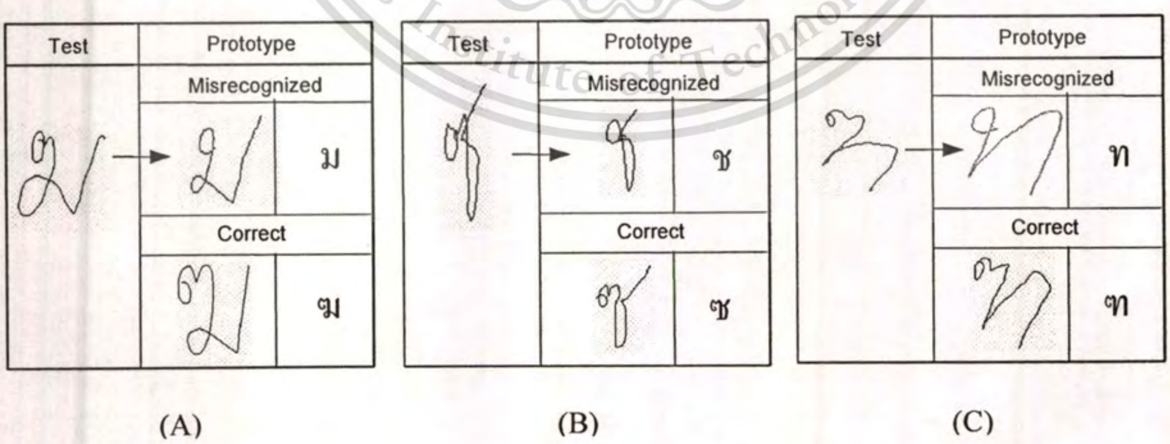


Fig 6 Misrecognitions by single representation of Thai handwritten character.

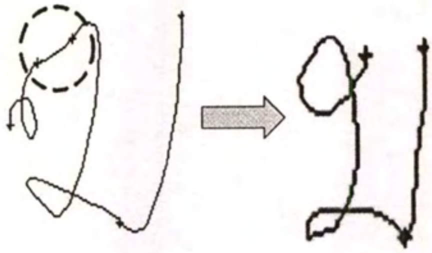


Fig 7 Misrecognition of Thai character because of missing the small real curve at head.

BIOGRAPHY

Mr. Khampheth Bounnady was born in May 28, 1977 at Samnue, Houaphanh Province, Laos P.D.R. He received his B.E. and M.E. in Computer Engineering department, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2001 and 2003 respectively. He is currently a doctoral student under an AUN/SEED-Net scholarship to study Electrical Engineering at King Mongkut's Institute of Technology Ladkrabang. He is also lecturer at the Department of Electronics Engineering, Faculty of Engineering, National University of Laos. His research interests include on-line handwriting recognition and data mining.