Thai Journal of Development Administration, Vol.32, No.4 October – December 1992

RECOGNITION OF THAI CHARACTERS

by

Pipat Hiranvanichakorn, Ph.D.[1]

[1] Lecture, School of Applied Statistics/Information Systems Education Center, The National Institute of Development Administration.

1. Introduction

This paper reports both off-line and on-line character recognition methods of Thai characters being composed of curves, and having many complicated and similar shapes. Both recognition methods are based on the structural analysis approach.

In this off-line system, Freeman chain code and directional differences of contour tracing are utilized for extracting concavities and convexities of character contours. Several local features of arcs are extracted, and are used to calculate similarity of arcs. Then, a pair of the most similar arcs between a model and an input character is detected, and is utilized as standard for determining the matching of other arcs of the characters. Finally, the similarities between arc portions and the similarity between characters are calculated, and the model whose the similarity is maximun value is selected as the identified character with the input one.

On the other hand, in on-line Thai character recognition, each character stroke is segmented into clockwise and counter clockwise arcs according as the stroke tracing is clockwise or counter clockwise, by making use of eight directional codes and directional differences of stroke tracing. Intuitively described features such as the sequence of stroke arcs, types of arcs and relative positions of arcs are extracted and utilized effectively for classifying characters.

By applying the methods to 69 categories of Thai characters, a very high recognition rate has been obtained.

2. Off-line recognition method

2.1 Recognition of printed Thai characters

2.1.1 Feature extraction algorithm

Extraction of concavity and convexity

Concave and convex portions of figures can be extracted by labeling a point as − or + at which counterclockwise boundary tracing turns to the right or left, respectively. In a paper,[2] contour pixels which indicate directional changes of Freeman chain code are detected. The pixels are then coded according to directional differences. The coded pixels are utilized to detect + and − vertices which indicate convexities and concavities of contours. An example of extracting + and − vertices of contours of a Thai character shown in Fig.1 (a) is depicted in Fig.1 (b). In a paper,[5] a concave arc is defined to be an uninterrupted sequence of concave vertices. Accordingly, when only two concave vertices appear between convex ones the portion is defined to be a concave arc inspite of being a straigth line. By avoiding the problem, + and − vertices are used to extract convex and concave arcs of a character as follows.

[Definition of concave and convex arcs]

When, there are successive + vertices between two − vertices, the successive + vertices including the two − vertices are called a convex arc, and a concave arc is the successive − vertices existing between two + vertices.

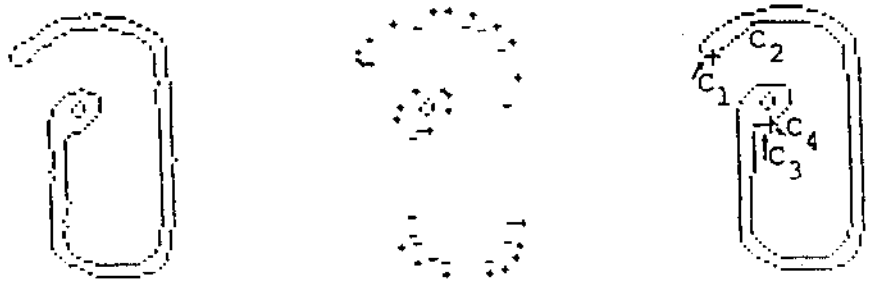
Accordingly, the start and end vertices of a convex (or concave) arc are − (or +) vertices. Further, adjacent concave and convex arcs have a common portion. As shown in Fig.1 (c), the segment passing points C_3, C_4, C_1 and C_2 is regarded as convex arc, and the segment passing points C_1, C_2, C_3 and C_4 is concave arc.

Features of Characters

In a paper[1], several geometric arc features are adopted effectively for the recognition of 128 × 128 dots of Thai characters. However, the features are insufficient for classifying small and similar characters. Thai characters are composed of arcs of complex structures and local information of arcs has important meanings for the classification of similar arcs. A simple extraction of local features of arcs is introduced[2] as follows.

Let $C_j = (C_{xj}, C_{yj})$; j = 1 ~ J be a sequence of + and − vertices belonging to an arc in the x−y plane, The vertices of the arc are transformed to a u−v plane (as shown in Fig.2), in the fashion that the start vertex of arc C_1 is located at the original point of the plane. Further, for avoiding the rotation problem, the first segment $C_1 - C_2$ is located along the positive u−axis.

Further, a feature extraction method[2] in which each character arc is segmented along u−axis and v−axis by points at which



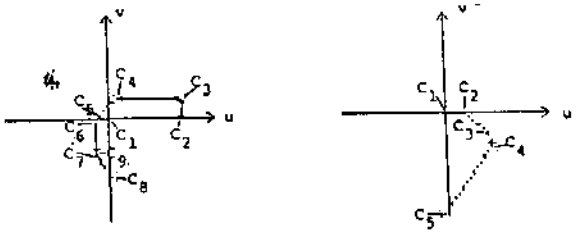
(a) (b) (c)

Fig.1 An example of extracting concavities and convexities.
(a) Original character.
(b) Character with labels + and −.
(c) Extraction of concave and convex arcs of character.

the arc bends, has been proposed. In the example shown in Fig.2 (a) the portions $C_1 - C_3$, $C_3 - C_7$, $C_7 - C_8$, $C_8 - C_9$ are chosen as segmented portions along u−axis(PU) and $C_1 - C_4$, $C_4 - C_8$, $C_8 - C_9$ as segmented portions along v−axis(PV). As for Fig.2(b), portion $C_1 - C_4$, $C_4 - C_5$ as PU and $C_1 - C_5$ as PV

The following parameters are utilized to describe features of arc in the u-v plane.

(1) SS which takes sign + for convex arc and takes sign − for concave arc.
(2) L length of arc.
(3) NU number of segmented portions along the u - axis.
(4) LU_{ku} ; ku = 1 ~ NU length of a segmented portion along the u−axis.
(5) NV number of segmented portions along the v−axis.
(6) LV_{kv} ; kv = 1 ~ NV length of a segmented portion along the v−axis.



(a) Convex arc (b) Concave arc

Fig.2 Examples of arcs drawn in u−v plane. (7)

2.1.2 Matching algorithm

In the papers[2],[3], effective matching methods have been proposed successfully for Thai characters. In the methods, the matching of SS, L, NU and NV between the arc i of an input character I and the arc j of a model MM is tested. If they match, the difference between each segmented portion length of the input arc and that of the model arc is calculated. If the difference is smaller than a threshold, scores which indicate if the considering feature pair matches or not, and link labels which indicate matched lavel are given to the arc pair. By using the given scores and link levels, the similarity between the arcs $SA_{i,j}$ is calculated.

When a rotated character is input to a computer, the arcs constituting the character appear in different order from those of a model. By the use of $SA_{i,j}$, a pair of the most similar arcs is detected from the model and the character.

The most similar arcs are such i and j as satisfy the following conditions.

(a) $(SA_{i-1,j-1} > S_b) \& (SA_{i,j} > S_c) \& (SA_{i+1,j+1} > S_b)$

(b) $\{ i,j \mid \max_{i=1}^{Z_{II}} (\max_{j=1}^{Z_{MM}} (SA_{i-1,j-1} + SA_{i,j} + SA_{i+1,j+1})) \}$

Where, S_b and S_c are thresholds. Z_{II}, Z_{MM} are the total number of concavities and convexities of the input data and the model, respectively.

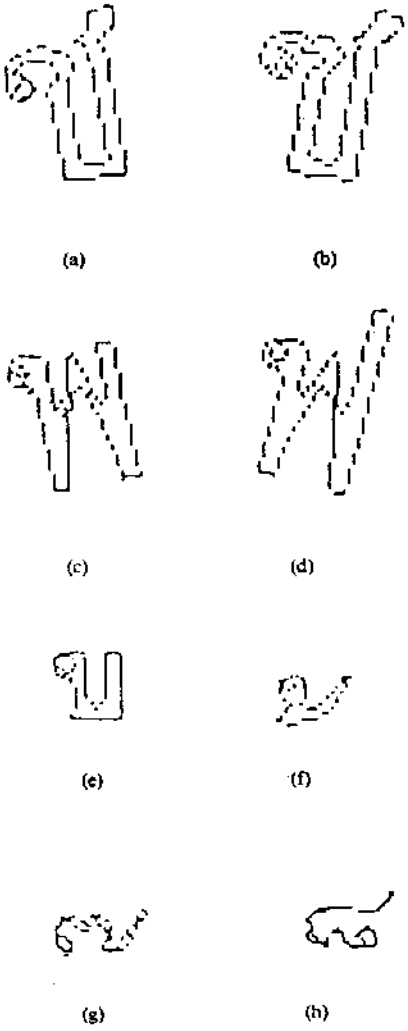
The most similar arc pair is utilized as standard for determining the matching of the next arc pairs. Finally, the similarity between characters $SC_{II,MM}$ is calculated as follows.

$$SC_{II,MM} = \sum_{i,j} (SA_{i,j}) * 100 / (2 * Z_{II}) \%$$

The model MM whose $SC_{II,MM}$ is maximum value and larger than a threshold is selected as the identified character with the input one.

2.1.3 Experimental results

The recognition method is applied to 69 categories (345 data) of 50 × 50 dots of printed Thai characters rotated with five kinds of arbitrary angles as shown in Fig.3. As for the experimental results, all characters except two errors were correctly identified. In fig.4, the ill-identified characters are shown.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig.3 Examples of 50x50 dots of Thai characters.

the data was identified as ก the data was identified as ไ

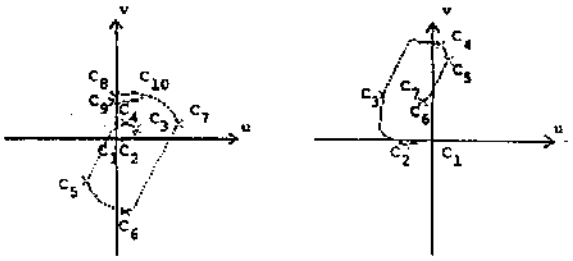
(a) (b)

Fig.4 Ill-identified characters.

2.2 Recognition of handprinted Thai characters,

The above recognition method was applied to handprinted Thai characters and the recognition rate of 70% was obtained. The feature extraction, in which the first segment of arc is used as standard for solving the rotation problem (as shown in Fig.2), is affected by the distortion of handprinted characters. Moreover, the model making, in which a single model is generated for each category, does not work well for handprinted Thai characters.

The above method is modified and applied to handprinted Thai characters. In the handprinted recognition method[3], the average direction of all segments of an arc is utilized to transform the arc vertices to a u−v plane as shown in Fig.5. Then segmented portions of arc along u−axis(PU) and those along v−axis(PV) are extracted, and are utilized to calculate the similarities between arc portions and the similarity between characters in the matching algorithm as described above.



(a) convex arc (b) concave arc

Fig.5 Example of arcs drawn on the u−v plane. (8)

2.2.1 Model making for handprinted characters

Model making is a difficult point of structural analysis methods. Further, for using practically, an effective dictionary of models having compact size is required. As several Thai characters have similar arc portions, the idea of using common arc portions among different models is considerable.

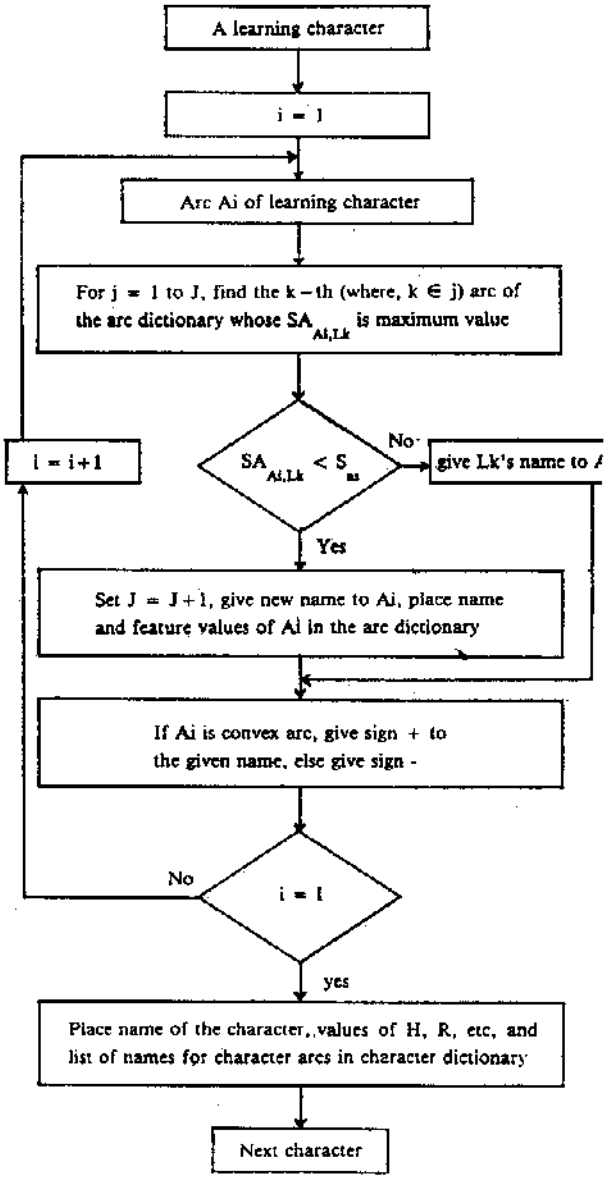
Dictionary of arcs and dictionary of characters.

In this method, the dictionary of models is divided into a dictionary of arcs and a dictionary of characters. The algorithm which provides the dictionary is shown in Fig.6.

Each arc of the learning character is compared to all arcs of the dictionary of arcs in order to find out a similar arc for the character arc. If a similar arc is found, the name of the arc is given to the character arc. Otherwise, a new name is given to the character arc. The given name and feature values of the arc are placed in the dictionary of arcs. After all arcs of the character have been named, name of the character and a list of the names of the character arcs which describe the character are placed in the dictionary of characters.

2.2.2 Experimental results

By applying the method to 828 data of 128 x 128 dots of handprinted Thai characters as shown in Fig.7, a recognition rate of 99.3% for learning data and a recognition rate of 88.9% for test data have been obtained. In Fig.8, examples of rejected and ill-identified characters are shown.



L_j:arc in arc dictionary, S_{as}:threshold

J:number of arcs of arc dictionary, I:number of character arcs

H:Number of holes, R:number of disjoint regions

Fig.6 A flowchart of model making.

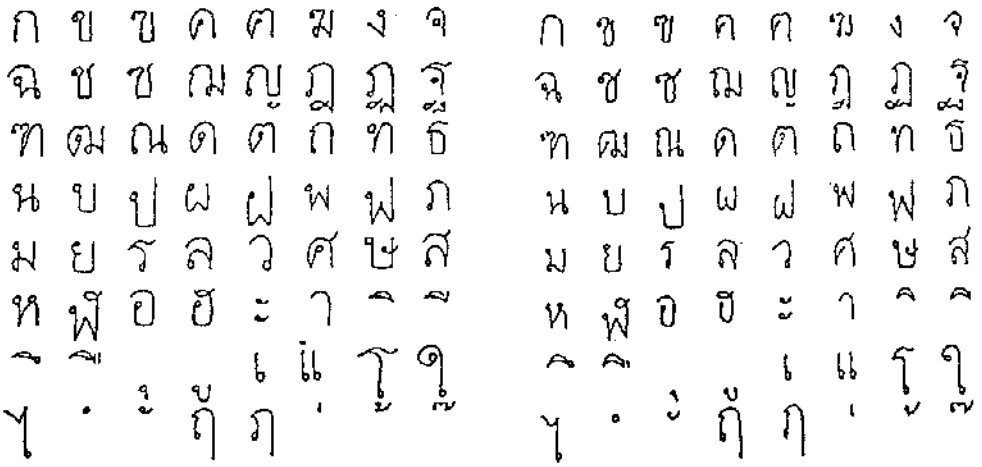


Fig.7 Example of data.

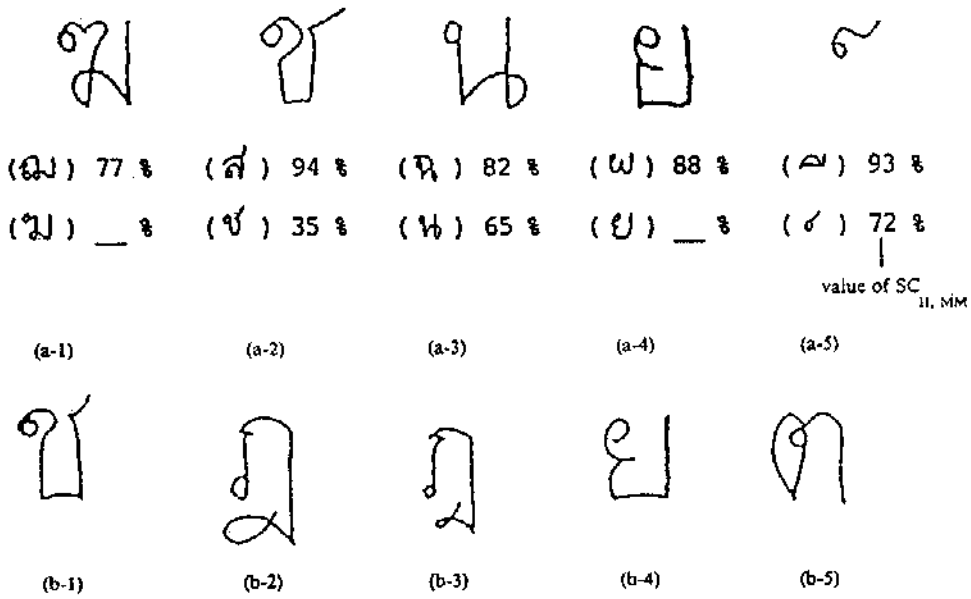


Fig.8 (a) Example of ill-identified characters
(b) Example of rejected characters.

3. On-line recognition method

3.1 Feature extraction algorithm

3.1.1 Segmentation of character strokes

Most Thai characters are single-stroke characters having similar and complex shapes. Therefore, the idea of segmenting a character stroke into several arcs and using features of arcs for the classification is considered to be effective for the recognition of the characters.

The extraction method of convexity and concavity described in 2.1.1 is modified and utilized to segment character strokes. In the on-line recognition method[4], a character is input to a computer through a digitizing tablet in the form of sequences of point coordinates representing character strokes. Stroke points of each character stroke are coded by eight directional codes. The points which indicate directional changes of the codes are detected. The points are then utilized to detect + and − vertices which indicate if

the stroke tracing is counter clockwise or clockwise. The + and − vertices are used to segment a stroke into counter clockwise and clockwise arcs.

A counter clockwise arc is the stroke portion passing successive + vertices, and the stroke portion passing successive − vertices is called a clockwise arc.

In the Fig.9, an example of detecting + and − vertices of a Thai character stroke is shown. The segment passing points C_3^*, C_4^* and C_5^* is regarded as a counter clockwise arc. Further, the segment passing points C_1^*, C_2^*, C_3^* and C_4^*, and that passing points C_4^*, C_5^*, C_6^*, C_7^*, C_8^*, C_9^*, C_{10}^* and C_{11}^* are clockwise arcs.

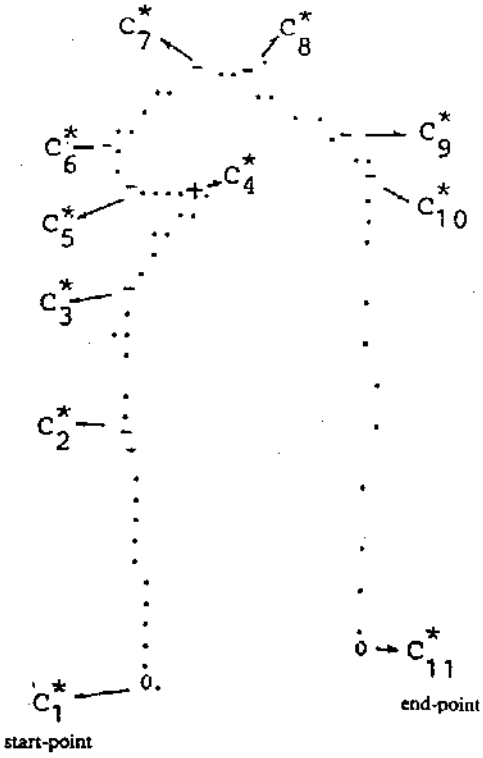
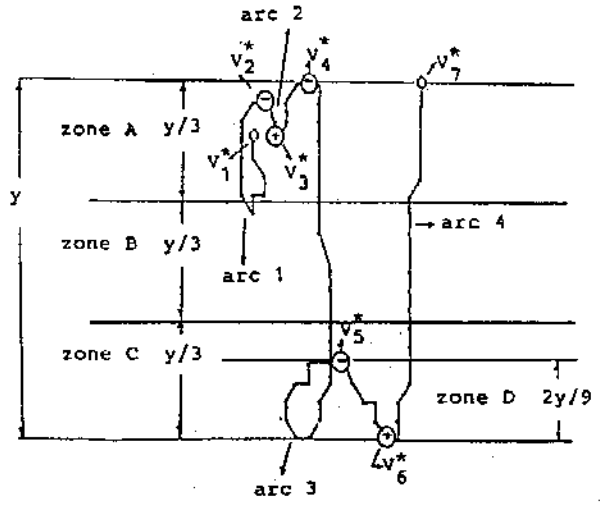


Fig.9 An example of assignment a character with + and −

3.1.2 Features of characters

Our feature extraction method is modified from the methods in Ref. (5), (6) in which character height is divided into zones, and the zones are utilized to extracted character features. As shown in Fig.10, the distance between the uppermost and lowermost points of each character stroke is divided along y-direction into three zones, i.e., zone A, zone B and zone C. One of the reasons is that most Thai characters are written from the portions existing at the uppermost or lowermost parts of the characters, and few characters are written from the middle part. Further, the distance along y-direction from the bottom of zone C to two third of the zone is called zone D.

Then, intuitively described features, such as the sequence of stroke arcs, types of arcs (i.e. clockwise or counter clockwise), and relative positions of loops, arcs, and the start and end points of the character stroke are extracted by making use of the zones. The features are used effectively for classifying Thai characters. For example, zone D is used effectively for classifying such similar characters as ก and ถ, ฝ and ฟ.



arc 1: the portion passing points V_1^*, V_2^* and V_3^*.
arc 2: the portion passing points V_2^*, V_3^* and V_4^*.
arc 3: the portion passing points V_3^*, V_4^*, V_5^* and V_6^*.
arc 4: the portion passing points V_5^*, V_6^* and V_7^*.

Fig.10 An example of dividing distance along y-direction of a character stroke into zones.

3.2 Experimental results

As for the classification of characters, the extracted features of an input character are compared with those of the models in the multi-step matching method[4]

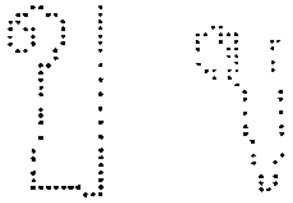
By applying the method to 414 data having patterns of 64x64 dots as shown in Fig.11, a recognition rate of 100% for learning data, and a recognition rate of 96.4% for test data have been obtained. In the Fig.12, the ill-identified characters are shown.

4. Conclusions

In this paper, an effective combination of feature extraction, matching process and model making is made for the recognition of Thai characters.

As for the feature extraction of both off-line and on-line recognition methods, we can say that Freeman chain code and directional differences of contour tracing and stroke tracing are utilized effectively for segmenting characters into meaningful portions such as concave (or convex) arcs and clockwise (or counter clockwise) arcs. Further, the local features of characters are used effectively for the classification of characters.

(a-1) (a-2)

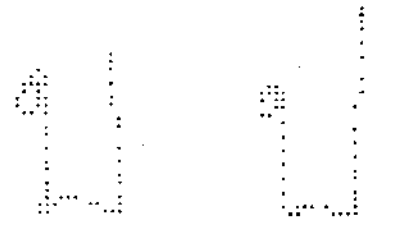


(b-1) (b-2)

(c-1) (c-2)

(d-1) (d-2)

Fig.11 Examples of characters



was identified as (a)

was identified as (b)

was identified as

(c)

Fig.12 Examples of ill-identified characters.

In off-line recognition method, the matching process in which the most similar arc pair is detected and utilized as standard for the matching of corresponding arcs of the considering characters, is proved to be effective for rotated characters.

From each kind of experimental results, the proposed recognition methods are considered to be effective for Thai characters.

References

(1) Hiranvanichakorn, P., Agui, T. and Nakajima, M. : "A recognition method of Thai characters", Trans. IECE Japan, E 65, 12, pp. 737-744 (Dec. 1982).

(2) Hiranvanichakorn, P., Agui, T. and Nakajima, M. : "Recognition method of Thai characters by using local features", Trans. IECE Japan, E 67, 8, pp. 425-432 (Aug. 1984).

(3) Hiranvanichakorn, P., Agui, T. and Nakajima, M. : "A recognition method of handprinted Thai characters by local features", Trans. IECE Japan, E 68, 2, pp. 83-90 (Feb. 1985).

(4) Hiranvanichakorn, P., Agui, T. and Nakajima, M. : "An on-line recognition method of Thai characters", Trans. IECE Japan, E 68, 9 (Nov.1985).

(5) Pavlidis, T. and Ali, F. : "Computer recognition of handwritten numerals by polygonal approximations", IEEE Trans. Syst., Man & Cybern., SMC-5, 3, pp.610-615 (1975).

(6) Sakai T. and Nagao M. : "Recognition machines of characters and shapes", pp. 63-68, KYOURITSUSHUPPAN (1967).

Supplements

A Recognition Method of Thai Characters

Pipat HIRANVANICHAKORN, Takeshi AGUI *and* Masayuki NAKAJIMA, *Regular Members*

UDC 003.337.4.055 : 159.937.52 : 681.32.05

SUMMARY A recognition method of printed Thai characters by the structural analysis of their contours is described. Digital contours of characters are encoded according to directional differences of contour tracing, then, simple arithmetic operations are applied to eliminating noises and extracting concavities and convexities of the contours. Several geometric features of concave and convex arcs are used to calculate similarities of the arcs, then, a pair of the most similar arcs between a model and an input character is determined from the similarities. Finally, for recognizing characters, the similarity between each arc pair of the character portion and the similarity between characters are calculated. By applying the method to 67 categories (670 data) of Thai characters, a very high recognition rate has been obtained.

1. Introduction

Recently, studies of character recognition of not only alphanumerics, Chinese characters, Japanese characters (Katakana), but also Korean[1], Arabic[2], Tamil characters[3] and others[4] have been made by many researchers. However, the study of Thai characters is not almost made, because they have many complicated and similar characters, and they are mainly composed of curves different from Chinese characters and Katakana being mainly composed of straight lines.

Recognition methods of characters and figures can be classified into two categories, one is the correlation method and the other one is the structural analysis method. The former is difficult to apply to rotated and distorted characters, and characters having many categories, though it is slightly affected by local noises[5]. The latter can be classified into the structural analysis of strokes and that of contour lines. The structural analysis of character strokes has been effectively applied to the recognition of Katakana[6] and Korean characters[1], in which characters are mainly composed of straight lines. However, for applying this method to characters composed of curves, there is difficulty in the extraction of strokes. In this paper, a structural recognition method of fundamental printed Thai characters shown in Fig. 1(a) is described.

As for the recognition of characters, many methods using the structural information of contours have been proposed. Especially, methods based on concavity and convexity information[7]-[10] are considered to be effective for such characters as alphanumerics which are composed of curves.

Manuscript received April 7, 1982.
Manuscript revised July 19, 1982.

The authors are with Imaging Science and Engineering Laboratory, Tokyo Institute of Technology, Yokohama, 227 Japan.

57—99 [P—72]

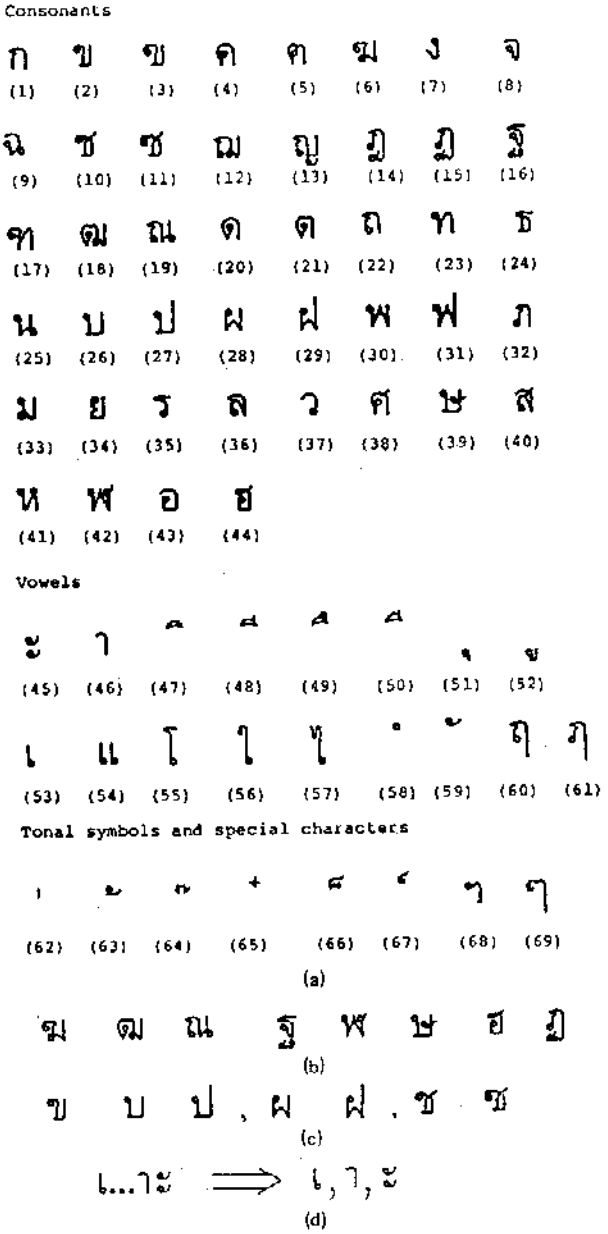


Fig. 1 (a) A set of typical Thai characters.
(b) Complicated characters.
(c) Similar characters.
(d) Combined characters.

In the paper[8], a successful recognition method of Katakana and alphanumerics has been proposed, in which concave and convex portions of characters are extracted by the outermost-point method, and many kinds of features of concavities and convexities constituting characters are utilized to calculate similarities among characters. However, this method is difficult to apply to such complicated characters as Thai ones shown in Fig. 1(b), because the simultaneous extraction of both large and small concavities and convexities in single process is difficult by it, besides, these features have important information in Thai characters. Further, because the local information is lost through the method, it requires and extra feature, i.e., angular distribution of edges in line segments by Freeman's code,* for recognizing such characters as D and O which are composed of only convex curves.

In the paper[10], the recognition of shapes by very simple matching is made by the use of high level attributes[9] of extracted concave and convex arcs. However, the extraction of features being used for the high level ones is complicated, and the matching method is not available for such very complicated and similar characters as shown in Fig. 1(b) and (c).

A simple and effective recognition method of Thai characters is proposed, in which an effective extraction of concavities, convexities and features, and matching process are dealt with. In the method, digital contours of characters are coded by directional differences in the process of contour tracing, and concavities and convexities of characters are extracted after simple processing of noise elimination. Then, a pair of the most similar arcs is detected between a model and an input data by using a few geometric features of concave and convex arcs. Finally, for recognizing characters, the similarity between each arc pair of the character portion and the similarity between the characters are calculated.

2. Thai Characters

Thai characters are composed of forty four consonant characters, thirty two vowel ones, four tonal symbols and four special characters. As shown in Fig. 1(d), several characters of the vowels are composed of the combination of fundamental characters. The decomposition of the combined characters into fundamental ones results in seventeen fundamental vowel characters.

3. Feature Extraction Algorithm

3.1 Directional Differences

Efficient coding of digital contours has been done by many researchers for the expression of pictures and figures, data compression[12,13] and picture analysis[10]. Concave and convex portions of figures can be extracted by labeling a point as + at which counterclockwise boundary tracing turns to the left and a point to the right as −. In this paper, signs (labels) are assigned to all contour pixels constituting a character according to directional changes shown in Fig. 2

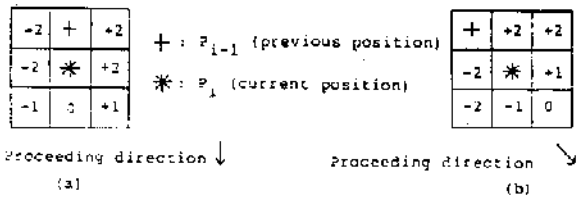


Fig. 2 Sign assignment according to directional differences.

(a) and (b).

$$P_n = (P_{xn},\ P_{yn}) : n = 1 \sim N,$$

$$A_n \in \{0,\ \pm 1,\ \pm 2\},$$

where, P_n is a contour element of a character in an $x-y$ plane, N is the number of the contour elements and A_n is an element of a set of signs for P_n.

Further, as the elements having a sign 0 are not used for the extraction of concavities and convexities, the following symbols are defined.

$$C_k = (C_{xk},\ C_{yk}) : k = 1 \sim K,$$

$$V_k \in \{\pm 1,\ \pm 2\},$$

where, C_k is a contour element which has not the sign 0, K is the number of the contour elements not having the sign 0, and V_k is an element of a set of signs for C_k

3.2 Extraction of Concavity and Convexity

Usually, contours of a character are obtained from clockwise or counterclockwise boundary tracing, and they include such noises as intrusions and protrusions. For eliminating such noises, many methods have been proposed[12,13]. However, in the recognition of Thai characters, sometimes, small intrusions and protrusions give important meanings to the character. In a paper[10], a successful method of eliminating contour noises for Thai characters has been proposed. Here, the method is applied to eliminating noises, then concavities and convexities of contours of characters are extracted.

Further, in the paper[9], vertices of concavities and convexities of an approximated character are labeled as − and +, respectively. In our method, after the elimination of noises, the labels, − and + are given to contour points according to the following definitions.

[Definition 1] When a contour point C_i satisfies either condition (a) or (b), a label + is assigned to C_i

(a) $(V_i > 0)\ \&\ (V_{i-1} + V_i + V_{i+1} > 0)$.

(b) $(V_i > 0)\ \&\ (V_{i-1} + V_i + V_{i+1} < 0)$. $\&\ (D_{i-1,i}\ \&\ D_{i,i+1} \geq k_1)$.

where, $D_{i,j} = \sqrt{(C_{xi} - C_{xj})^2 + (C_{yi} - C_{yj})^2}$

and k_1 is a threshold.

[Definition 2] When a contour point C_i satisfies either condition (a) or (b), a label − is assigned to C_i.

(a) $(V_i < 0)\ \&\ (V_{i-1} + V_i + V_{i+1} < 0)$.

(b) $(V_i < 0)$ & $(V_{i-1} + V_i + V_{i+1} > 0)$
& $(D_{i-1,i}$ & $D_{i,i+1} \geq k_1)$.

In Fig. 3(a) and (b), labeling examples of + are illustrated, and in Fig. 4(a) a Thai character with labels + and − assigned by the definitions is given with its original one shown in Fig. 4(b).

In a paper [11], a concave arc is defined as an uninterrupted sequence of concave vertices. Accordingly, when only two concave vertices appear between convex ones, the portion is defined to be a concave arc in spite of being a line. For avoiding the problem, a convex (concave) arc is defined as follows.

When there are successive + vertices between two − vertices, the successive + vertices are called a convex arc, and a concave arc is the successive − vertices put between two + vertices.

Accordingly, the start and end vertices of a convex (or concave) arc are − (or +) vertices, and adjacent concave and convex arcs have a common portion.

For vertices $C_j = (C_{xj}, C_{yj})$: $j = 1 \sim J$ belonging to an arc, the length L, distance L_v between two end vertices, curvature CE and centroid $G = (G_x, G_y)$ of the arc are defined as follows.

$$L = \sum_{j=1}^{J-1} D_{j,j+1}\ ,\ L_v = D_{1,J}\ ,\ CE = L / L_v\ ,$$

$$G_x = \left(\sum_{j=1}^{J-1} (C_{xj} + C_{x,j+1}) \times D_{j,j+1} \right) / 2L\ ,$$

$$G_y = \left(\sum_{j=1}^{J-1} (C_{yj} + C_{y,j+1}) \times D_{j,j+1} \right) / 2L\ .$$

3.3 Approximation by Line Segment

As shown in Fig. 5, some portions constituting a character have low curvature. When rotated characters are considered to be recognized, these low curvature portions of the characters give different number of arcs to the characters of the same category. For avoiding such difficulty, an approximation method by straight line segments is applied to Thai characters.

When an arc satisfies the following conditions, the arc is approximated by a straight line segment.

(a) $CE < C_{CE}$

(b) $(D_S < C_{DS})$ or $(D_L < 10 \times D_S)$ or $(\theta \geq \theta_A)$

where, D_L is the longest distance between two vertices constituting an arc, D_S is the remainding distance of the arc, $\theta = \cos^{-1}\left[(D_L^2 + D_S^2 - L_v^2) / (2 \times D_L \times D_S)\right]$, and C_{CE}, C_{DS} and θ_A are thresholds, respectively.

In Fig. 6(a) and (b), approximated straight line segments L_i are shown, and in Fig. 7, an example of a character after line approximation is illustrated.

3.4 Features of Characters

Both of global features and local features of contours are required for the recognition of Thai characters, because they have many similar characters and categories. As for the global features, the number of holes H, that of disjoint regions R, the total number of concavities and convexities Z, and the number of contour pixels (perimeter of character) N are used.

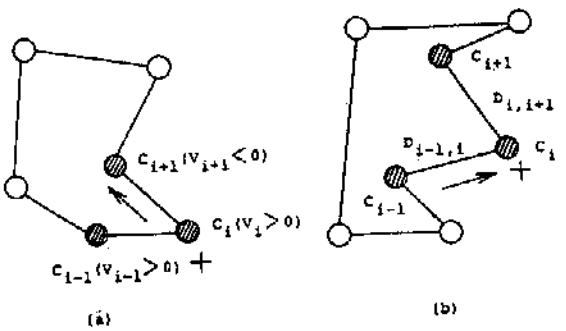


Fig. 3 Labeling of contour elements with +.

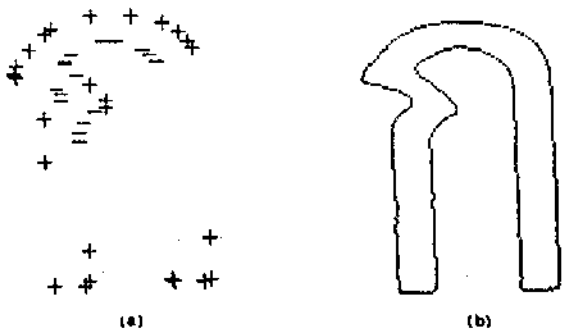


Fig. 4 An example of a Thai character with labels + and −
(a) A character with labels + and −
(b) Original character of (a).

Fig. 5 Example of characters having low curvature portions.

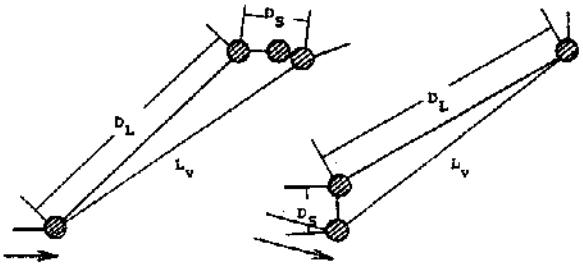


Fig. 6 Approximation of low curvature arcs by line segments.

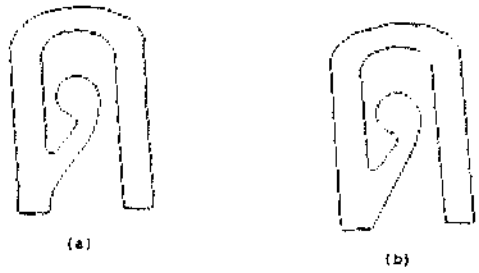


Fig. 7 Example of a character after line approximation processing.
(a) Original character.
(b) Approximated character.

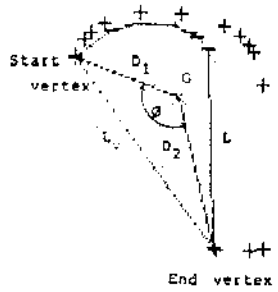


Fig. 8 Five features of an arc.

Fig. 9 Examples of characters composed of only convexities.

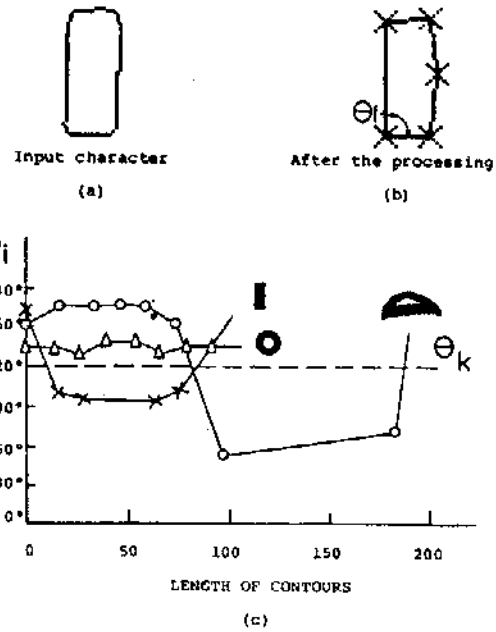


Fig. 10 An example of the detection of θ_i and the distribution of θ_i.

In the paper[8], many kinds of features of an arc are utilized for the recognition of Katakana and alphanumerics. However, the use of many features increases the computation time of matching. Further, for recognizing rotated characters, features which are independent of rotations are required. Here, as for features of an arc, the following five geometric parameters are adopted.

(1) length of arc: L.

(2) distance between the start and end vertices of arc: L_e.

(3) distance between the start vertex and the centroid of arc: D_1.

(4) distance between the end vertex and the centroid of arc: D_2

(5) angle of the start vertex, the centroid and the end vertex: $\emptyset$.

In Fig. 8, these parameters are illustrated.

In Thai characters, there are characters which are composed of only convex arcs as shown in Fig. 9. For such characters, the following processing is made.

(i) When the distance between successive two + vertices of a character is short, they are merged and make a single vertex whose position is given by the average of the coordinates of the two vertices.

(ii) After the approximation of curves by straight line segments, the number of vertices of the approximated polygon whose vertex angles θ_i are smaller than a threshold θ_k is taken up as a feature parameter.

In Fig. 10, θ_i at every vertex is shown for characters composed of only convexities.

4. Matching Process

As matching methods of models with input data, flexible matching[7],[8] and relaxation matching method[9] are widely applied. However, these methods are complex and require much computation time which exponentially depends on the complexity of objects. Further, it is difficult to apply them to the recognition of rotated objects. In the paper[10], a simple and fast matching method has been proposed, in which pairs of similar arcs of model and data were linked and the whole similarities were calculated. However, this method requires complicated features[9] and is difficult to apply to the recognition of characters composed of similar arcs.

Our matching method is related to that in the Ref. (10), and it is proposed to recognize Thai characters which have many categories and characters composed of similar arcs, and moreover, to solve the problem of rotated characters.

For decreasing matching time, first, the matching of H, R and Z between a model and an input data is made. If they match, and the difference of the number of contour pixels N is smaller than a threshold (denoted by DIFN),

the matching of arcs is made according to the following rules.

(i) The matching between concave arc and convex one is not made.

(ii) Suppose $II_{i,k}$ and $MM_{j,k}$ ($k = 1 \sim 5$, where k is the number of geometric parameters of an arc, and $k=1$ means the feature L, $k=2$; L_v, $k=3$; D_1, $k=4$; D_2, $k=5$; θ) be the feature values of the i-th arc of an unknown input data II, and those of the j-th arc of a model MM, respectively. Then the following scores T_k which indicate if the considering feature pair matches or not, and link levels R_k which indicate matched level are given to the arc pair according to the conditions (a) and (b).

(a) $T_k = 1$, $R_k = 2$

for $|II_{i,k} - MM_{j,k}| \leq \epsilon_{1k}$

(b) $T_k = 1$, $R_k = 1$

for $|II_{i,k} - MM_{j,k}| \leq \epsilon_{2k}$

where, ϵ_{1k} and ϵ_{2k} are the allowance values of features, and determined experimentally by the values of $MM_{j,k}$.

(iii) By the use of T_k and R_k, a similarity $SA_{i,j}$ between arcs is determined as follows.

$$SA_{i,j} = \sum_{k=1}^{5} (T_k / 5 + R_k / 10)$$

$SA_{i,j}$ is given in the range from 0.0 to 2.0, however, when $SA_{i,j}$ has value smaller than a threshold (denoted by S_{ak}), zero is given to it. Here, the value of 0.9 is selected for S_{ak}, and it means that the matching arc pair is considered to be similar if at least three of their features match.

When a rotated character is inputted to a computer, the arcs constituting the character appear in different order from those of a model. By the use of $SA_{i,j}$, similar arc pairs between the data and the model can be detected. However, as Thai characters have many characters composed of similar arcs, the information of currently considering arcs and that of connecting arcs to them are utilized for detecting the similar arc pairs. Further, for facilitating the process, at first a pair of the most similar arcs is detected from the input data and the model. Then, by using the detected arc pair, the next similar arcs are sequentially found.

The most similar arcs are such i and j as satisfy the following conditions.

(a) $(SA_{i,j} > 0) \,\&\, (SA_{i-1,j-1} > 0) \,\&\, (SA_{i+1,j+1} > 0)$.

(b) $$\left\{ i, j \;\middle|\; \max_{i=1}^{Z_{II}} \cdot \left(\max_{j=1}^{Z_{MM}} \cdot (SA_{i,j} + SA_{i-1,j-1} + SA_{i+1,j+1}) \right) \right\},$$

where i is an arc of an input data, j is that of a model and Z_{II}, Z_{MM} are the total number of concavities and convexities of the input data and the model, respectively. When the number of Z_{II} is large, it increases the computation time for detecting the most similar arc pair. However, in practice, a pair of the most similar arcs between two similar characters can be detected by using only several arcs constituting the characters. When no pair of the most similar arcs is found between an input data and a model, the model is changed to other one. On the other hand, when a pair of the most similar arcs i, j is found, then corresponding arcs, . . . , $(i-2, j-2)$, $(i-1, j-1)$, $(i+1, j+1)$, $(i+2, j+2)$ . . . are determined between the input data and the model. By calculating the similarity between every determined arc pair, the similarity between the characters $SC_{II,MM}$ is calculated according to the following equation.

$$SC_{II,MM} = \sum_{i,j} (SA_{i,j} \times 100) / (2 \times Z) \quad (\%)$$

This similarity is calculated for all models whose parameters, H, R, Z and N are matched to those of the input character. The model MM, which has the maximum $SC_{II,MM}$ value and satisfies the following inequality, is selected as the identified character with the input one.

$SC_{II,MM} > S_{ck}$ (S_{ck} : a threshold).

5. Models

Model making is a difficult point of structural analysis methods, especially, that for rotated characters. For facilitating it, the following model making method is utilized, in which the model of a category is made from the input characters belonging to the category, and having the same numbers of parameters H, R and Z.

Suppose that the i_1-th arc of a character $I1$ is the most similar arc to the i_n-th ($n = 2, 3, \cdots, m$) arc of a character In, in which m is the number of characters belonging to the same category and having the same numbers of parameters H, R and Z. Then, the i-th arc features of the model of the category are determined as follows.

$$MM_{i,k} = \sum_{n=1}^{m} In_{i_n,k} / m, \quad (k = 1 \sim 5)$$

In the same way, other arc features are defined by the following equations.

$$\vdots$$

$$MM_{i-1,k} = \sum_{n=1}^{m} In_{i_n-1,k} / m,$$

$$MM_{i+1,k} = \sum_{n=1}^{m} In_{i_n+1,k} / m,$$

$$\vdots$$

These arc features of the model are calculated as many as the number of arcs, and a single model is obtained for each category.

6. Experimental Results

Recognition experiments were made according to the algorithm described above. Sixty seven of Thai characters were inputted to a mini-computer (OKITAC-4300b, 32 kW, 16 bits/W), as a binary data of 256 × 256 dots. These data were reduced to 128 × 128 dots. In the process of taking the characters into the computer, they were rotated with arbitrary angles. In total, 670 data, i.e., 335 (arbitrary five kinds of rotations for each character) for fine (256 × 256 dots) characters and that for coarse (128 × 128 dots) characters, were used for the experiments. In Fig. 11 (a), (b), (c), (d) and (e), examples of five kinds of arbitrary

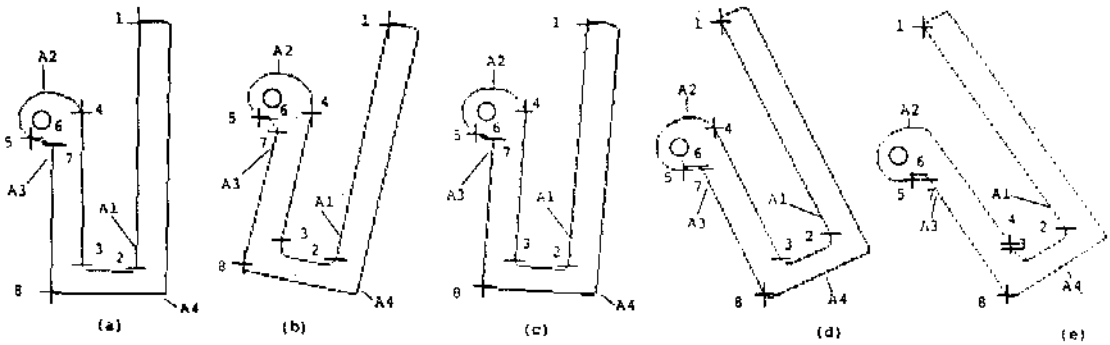


Fig. 11 (a), (b), (c), (d) and (e) Examples of five kinds of arbitary rotations for a Thai character.
arc A1: The segment passing points 1, 2, 3 and 4.
arc A2: The segment passing points 3, 4, 5 and 6.
arc A3: The segment passing points 5, 6, 7 and 8.
arc A4: The segment passing points 7, 8, 1 and 2.

Table 1 Feature values of input characters shown in Fig. 11.
(H = 1; R = 0; Z = 4; N = 632)

	L	L_v	D_1	D_2	Ø
A1	254	59	89	39	36
A2	154	70	64	9	134
A3	95	86	40	47	168
A4	449	82	45	50	120

where;

arc A1 : The segment passing points 1,2,3 and 4.

arc A2 : the segment passing points 3,4,5 and 6.

arc A3 : the segment passing points 5,6,7 and 8.

arc A4 : the segment passing points 7,8,1 and 2.

(a)

(H = 1;R = 0;Z = 4;N = 618)

	L	L_v	D_1	D_2	Ø
A1	246	65	86	33	41
A2	147	67	60	12	124
A3	90	82	38	45	167
A4	445	77	44	47	118

(b)

(H = 1;R = 0;Z = 4;N= 627)

	L	L_v	D_1	D_2	Ø
A1	253	60	86	38	37
A2	157	70	63	9	136
A3	93	85	39	46	169
A4	450	82	46	49	120

(c)

(H = 1;R = 0;Z = 4;N = 585)

	L	L_v	D_1	D_2	Ø
A1	251	58	84	38	36
A2	152	72	63	13	129
A3	90	82	38	45	168
A4	443	80	46	47	119

(d)

(H = 1;R = 0;Z = 4;N = 544)

	L	L_v	D_1	D_2	Ø
A1	180	129	84	49	152
A2	150	67	61	10	124
A3	90	83	38	45	171
A4	449	80	44	50	118

(e)

Table 2 Feature values of the model.
(H = 1; R = 0; Z = 4; N = 601)

	L	L_v	D_1	D_2	Ø
A1	237	74	85	39	60
A2	153	69	62	11	129
A3	92	84	39	46	169
A4	447	80	45	49	119

Table 3 Values of thresholds.

k_1 = 5.0 for 256 x 256 characters.

k_1 = 3.0 for 128 x 128 characters.

C_{CE} = 1.04

C_{DS} = 5.0

θ_A = 155°

θ_k = 120°

DIFN = 100 for 256 x 256 characters.

DIFN = 50 for 128 x 128 characters.

for k = 1~4,

ξ_{1k} = 5 ; ξ_{2k} = 10 for $MM_{j,k} \leq 50$

ξ_{1k} = 10 ; ξ_{2k} = 15 for $MM_{j,k} \leq 100$

ξ_{1k} = 15 ; ξ_{2k} = 20 for $MM_{j,k} > 100$

for k = 5, ξ_{1k} = 30° , ξ_{2k} = 35°

S_{ak} = 0.9

S_{ck} = 50 %

Table 4 Result of the detection of the most similar arcs.

$SA_{A1,A1}$ = 1.80 ; $SA_{A4,A4} + SA_{A1,A1} + SA_{A2,A2}$ = 5.80

$SA_{A1,A3}$ = 0.00 ;

$SA_{A2,A2}$ = 2.00 ; $SA_{A1,A1} + SA_{A2,A2} + SA_{A3,A3}$ = 5.80

$SA_{A2,A4}$ = 0.00 ;

$SA_{A3,A1}$ = 0.00 ;

$SA_{A3,A3}$ = 2.00 ; $SA_{A2,A2} + SA_{A3,A3} + SA_{A4,A4}$ = 6.00

where, $SA_{Ai,Aj}$ means the similarity of arcs between arc Ai of the input character and arc Aj of the model. In this example, arcs A3, A3 are the pair of the most similar arcs.

Table 5 Values of $SA_{i,j}$ and $SC_{II,MM}$. (for character shown in Fig. 11(a) and its model.)

$SA_{A1,A1}$ = 1.80 ; $SA_{A2,A2}$ = 2.00 ;

$SA_{A3,A3}$ = 2.00 ; $SA_{A4,A4}$ = 2.00 ;

$SC_{II,MM}$ = 97.5 % .

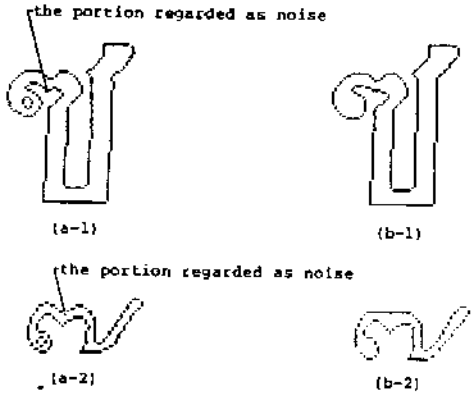


Fig. 12 Examples of rejected characters.
(a) Original characters.
(b) The characters after the processing.

rotations for a character are illustrated, and in Table 1 (a), (b), (c), (d) and (e), corresponding values of features of each data are given. From the data, the model is made and its calculated values of features are given in Table 2. Further, an experimental example of matching between the data shown in Fig. 11(a) and its model is shown, by the use of values of thresholds given in Table 3, as follows. Table 4 gives some calculated values of the similarity between arcs $SA_{i,j}$ and the result of the detection of the most similar arcs between the data and the model. After the detection of the most similar arcs, the similarity between every arc pair and that between characters $SC_{II,MM}$ are calculated; and their values are given in Table 5.

As for the experimental results, all characters except one rejected character were correctly identified in the case of fine characters, and except six rejected characters, other ones were correctly identified for coarse characters. Sometimes, the rotation of some characters generates arcs with low curvature, and it causes the rejection of the characters. Further, for coarse characters, as shown in Fig. 12, some small concavities and convexities are eliminated regarded as noises because of the low resolution of the characters.

7. Conclusions

In this paper, the recognition of Thai characters is described with experimental results. An effective feature extraction and a matching method of Thai characters which have many categories and complicated similar characters are proposed.

In the method, digital boundary tracing of characters is made and the traced boundary is coded in four directional differences to obtain a coded contour for each character. After the elimination of small digital errors, the concavities and convexities of each character are extracted by a simple method. The recognition is made by the use of similarities between arcs constituting characters and similarity between characters.

For decreasing the computation time of matching, the numbers of holes, disjoint regions and concavities and convexities, and the perimeters of characters are utilized as the parameters of matching. Further, models are made from input data. The features being used here are mainly geometric ones whose values vary with data sizes. However, the method is applicable to a variety of data sizes by selecting proper values of thresholds, i.e., ℓ_1, DIFN, $\epsilon_{1\ell}$ and $\epsilon_{2\ell}$. The proposed method was applied to a variety of 670 printed Thai characters obtained from five values of rotations and two kinds of sizes, and 99.7% of recognition rate for 256 × 256 characters, and 98.2% of that for 128 × 128 characters have been achieved.

References

(1) Agui, T., Nakajima, M., Kim, T.K. and Takahashi, E.T.: "A method of recognition and representation of Korean characters by tree grammars", IEEE Trans., PAMI-1, 3, pp. 246–251 (1979).

(2) Badie, K. and Shima, M.: "Machine recognition of arabic cursive scripts", Int. Con. of Pattern Recog. in Practice. Ams. (1980).

(3) Chinnuswamy, P. and Krishnamoorthy, S.G.: "Recognition of handprinted Tamil characters", Pattern Recognition, 12, 3, pp. 141–152 (1980).

(4) Sethi, I.K. and Chatterjee, B.: "Machine recognition of constrained handprinted Devanagari", Pattern Recognition, 9, 4, pp. 69–75 (1977).

(5) Suen, C.Y., Berthod, M. and Mori, S.: "Automatic recognition of handprinted characters — The state of the art", Proc. IEEE, 68, 4, pp. 469–487 (1980).

(6) Masuda, I.: "Machine recognition of handprinted Japanese characters KATAKANA using geometric features", Trans. IECE Japan, 55-D, 10, pp. 638–645 (Oct. 1972).

(7) Yamamoto, K., Mori, T. and Mori, S.: "Machine recognition of handprinted Japanese characters KATAKANA and numerals", Trans. IECE Japan, J59-D, 6, pp. 414–421 (June 1976).

(8) Yamamoto, K. and Mori, S.: "Recognition of handprinted characters by an outermost point method", Pattern Recognition, 12, 4, pp. 229—236 (1980).
(9) Pavlidis, T. and Ali, F.: "A hierarchical syntactic shape analyzer", IEEE Trans., PAMI-1, 1, pp. 2—9 (1979).
(10) Pavlidis, T.: "The use of a syntactic shape analyzer for contour matching", IEEE Trans., PAMI-1, 3, pp. 307—310 (1979).
(11) Pavlidis, T. and Horowitz, S.L.: "Segmentation of plane curves", IEEE Trans. Comput., C-23, pp. 860—870 (1974).
(12) Lee, C.C., Hwang J.J., Hall, E.L. and Thomason, M.G.: "New direction vector coding for shape description", 12-th IEEE Southeastern Symposium, SYSTEM THEORY, pp. 44—54 (1980).
(13) Nakajima, M. and Agui, T.: "A delta coding method of contour lines", Trans. IECE Japan, J64-D, 2, pp. 109—115 (Feb. 1981).
(14) Martin, W.N. and Aggarwal, J.K.: "Computer analysis of dynamic scenes containing curvilinear figures", Pattern Recognition, 11, 3, pp. 169—178 (1979).
(15) Davis, L.S.: "Shape matching using relaxation techniques", IEEE Trans., PAMI-1, 1, pp. 60—72 (1979).
(16) Agui, T., Hiranvanichakorn, P. and Nakajima, M.: "Recognition of printed Thai characters", Paper of Technical Group, TGPRLSI-30, IECE Japan (1981).
(17) Pavlidis, T. and Ali, F.: "Computer recognition of handwritten numerals by polygonal approximations", IEEE Trans. Syst., Man & Cynbern., SMC-5, 3, pp. 610—615 (1975).

Recognition of Thai Characters by Using Local Features

Pipat HIRANVANICHAKORN, Takeshi AGUI *and* Masayuki NAKAJIMA, *Regular Members*

UDC 803.337.4.055 : 159.937.52 : 681.32.05

SUMMARY A recognition method of printed Thai characters by local information of contours is described. In the method, Freeman chain code and directional differences of contour tracing of characters are utilized for eliminating contour noises and extracting concavities and convexities of characters. Each arc is then segmented by points at which the arc bends. Several local features of arcs are extracted, and are used to detect a pair of the most similar arcs between a model and an input character. Finally, the similarity between each arc pair and the similarity between characters are calculated. Further, a model making, in which a single model is generated for each category by making use of feature values of characters belonging to the same category, is described. By applying the method to 69 categories (345 data) of 50 × 50 dots of Thai characters rotated with five kinds of arbitrary angles, a very high recognition rate has been obtained.

1. Introduction

In the field of pattern recognition, the studies of character recognition have been made for more than two decades, because they provide possible commercial applications, such as postal sorting machines, credit card readers, etc. Up to now, most of character recognition techniques are proposed for alphanumerics, Japanese characters and Chinese ones. Recently, the recognition studies of Korean[1], Arabic[2], Hebrew[3] and various kinds of characters[4] have been made, because each character has different properties and there are many interesting and challenging problems to be dealt with. A few researches[5],[6] of Thai characters have been made, because the characters have many complicated and similar ones, and they are mainly composed of curves. In the purpose of possible commercial use, a system having capability of recognizing low resolution characters is needed. The recognition of low resolution characters is difficult. As for Thai characters, small intrusions and protrusions give important meanings to the characters and the extraction of features is somewhat difficult. In this paper, a recognition method of printed Thai characters with resolution of 50×50 dots shown in Fig. 1 is proposed.

Up to now, methods using global information are popularly used in pattern recognition, because the features are easily extracted and are slightly affected by local noises. However, when an important portion of the input character disappears, recognition results are seriously affected. On the other hand, local information methods can be classified into structural analysis of strokes and that of contour lines. The contour analysis approach, especially methods based on concavity and convexity information are considered to be effective for characters being composed of curves, such as alphanumerics[7], Hiragana[8] and Thai ones[5],[6]

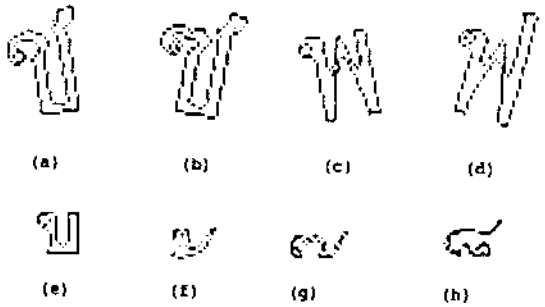


Fig. 1 Examples of 50×50 dots of Thai characters.

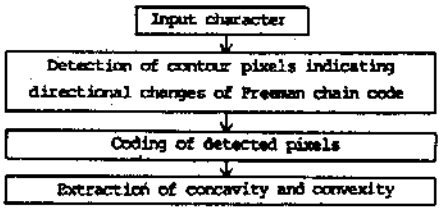


Fig. 2 A block diagram of concavity and convexity extraction.

In the paper[8], a successful recognition method of Hirakana has been proposed, in which concave and convex portions are extracted by detecting start and end points of the portions. Its main disadvantage is that local details of arcs are lost. Further, the method is difficult to apply to such similar characters shown in Fig. 1(a) and (b).

In the papers[5],[6], a successful recognition method of 128×128 dots of Thai characters has been proposed. The method was applied to 50×50 dots of Thai characters (345 data) and the recognition rate of 90% was obtained, because the concavity and convexity extraction did not work well for low resolution characters. Further, the features used in it were not sufficient for classifying similar characters.

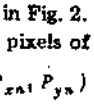
As a modification of the method[6], this paper reports a simple and effective extraction method of concavities, convexities and features for the recognition of low resolution Thai characters. Further, a model making, in which a single model is automatically generated for each category, is introduced. In this method, Freeman chain code and directional differences of contour tracing of characters are utilized for extracting concavities and convexities. Each arc is then

Manuscript received October 31, 1983.
Manuscript revised February 7, 1984.

The authors are with Imaging Science and Engineering Laboratory, Tokyo Institute of Technology, Yokohama, 227 Japan.

segmented by points at which the arc bends. Several local features of arcs are extracted, and are used to calculate similarities between arcs. Then, a pair of the most similar arcs between a model and an input character is detected, and is utilized as standard for determining the matching of the next arcs. Finally, for recognizing characters, the similarity between each arc pair of the character portion and the similarity between the characters are calculated. Further, when the total number of concavities and convexities between a data and a model is different, the recognition is made by detecting similar arc pairs from the characters.

2. Thai Characters

Thai characters are composed of forty four consonant characters, thirty two vowel ones, four tonal symbols and four special characters. Several vowel characters can be decomposed into fundamental ones, making a total of sixty nine fundamental ones in all[6]. In the view of character recognition, the characters are said to be composed of curves. Most of the characters have holes, and few characters, such as  are separated.

3. Feature Extraction Algorithm

3.1 Extraction of Concavity and Convexity

In shape analysis and character recognition, polygonal boundary approximation approach such as the split-and-merge one is used in various studies[9]-[11], because it preserves local details and allows extensive use of semantics. The major disadvantage of the method is that the program is complex and the process is time-consuming one. For avoiding the problem, methods based on chain codes[12]-[14] have been attempted on straight line detection. However, noises are not reduced by the methods, and therefore processing such as filtering is needed in the next step.

In the paper[6], a successful method of detecting + and − vertices which indicate convexities and concavities of characters has been proposed, by making use of directional differences of traced contours. Unfortunately, the method failed when it was applied to low resolution Thai characters. Here, both the chain code and the directional difference ideas are utilized for simple and effective extraction of concavities and convexities of 50×50 dots of Thai characters. A block diagram of concavity and convexity extraction is shown in Fig. 2.

Contour pixels of a character are represented as follows.

$$P_n=(P_{xn}, P_{yn}) : n=1\sim N \qquad (1)$$

where, P_n is the n-th element of the contour of a character in an x-y plane, N is the number of element of the contour.

A digital distance, $DD_{i,j}$, between i-th and j-th pixels is defined as follows.

$$DD_{i,j} = (\text{pixel number between } i\text{-th and } j\text{-th pixels}) + 1 \qquad (2)$$

Fig. 3 An example of elements indicating directional changes of Freeman chain code.
(a) Contour pixels indicating directional changes.
(b) Original character of (a).

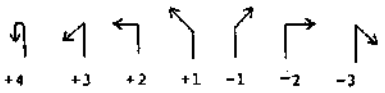


Fig. 4 Sign assignment according to directional differences.

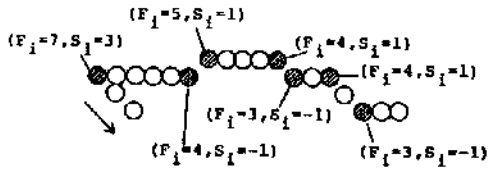


Fig. 5 Examples of elements having signs.

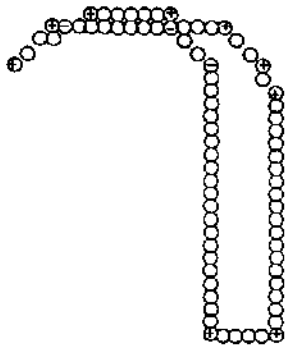


Fig. 6 An example of a Thai character with labels + and −.

Since, concave and convex portions of figures can be extracted by labeling a point as − or + at which counter-clockwise boundary tracing turns to the right or left, respectively. Further, a sequence of contour pixels having the same chain code value is considered to make a line whose direction is described by the chain code value. In this paper, contour pixels which indicate directional changes of Freeman chain code are detected. Each detected pixel is given a chain code value which describes direction of contour portion at the point, and is given a + (or −) value which describes convexity (or concavity) state. However, contours of characters contain such noises as protrusions and intrusions. The pixels which indicate directional changes of chain code, and are not regarded as noise pixels are detected by a processing given in Appendix. Fig. 3(a) shows detected contour pixels of a Thai character shown in Fig. 3(b).

Let $C_i=(C_{xi}, C_{yi})$ $: i=1\sim I$ be a sequence of contour pixels indicating directional changes of chain code, and let $F_i \in \{0\sim 7\}$ be chain code value of C_i. Then, a sign S_i is assigned to each element C_i according to directional differences of chain code illustrated in Fig. 4, i.e., $S_i \in \{+1,+2,+3,+4,-1,-2,-3\}$. In Fig. 5, examples of elements having signs are shown.

In the paper[9, 10], concavities and convexities are extracted by calculating angles of vertices of approximated polygons. Then, concave and convex vertices are labeled as — and +, respectively. Here, — or + is given to a contour point according to the following definitions.

[Definition 1] When a contour point C_i satisfies either condition (a) or (b), a label + is assigned to C_i.

(a) $(S_i>0)\&(S_{i-1}>0 \text{ or } S_{i+1}>0)$

(b) $(S_i>0)\&(S_{i-1}<0)\&(S_{i+1}<0)$

$$\&\,(F_i \neq FF)\&(DD_{i,i+1} \geq k_1)$$

where, $DD_{i,i+1}$ is the digital distance between points C_i and C_{i+1}, k_1 is a threshold. FF is a digital direction (i.e., chain code value) of the contour portion before coming to C_i. Further, when C_i is selected as a + vertex and $DD_{i,i+1}$ is larger than a threshold k_2, FF is newly set by the value of F_i. Value 4 which indicates horizontal direction, is initially given to FF.

[Definition 2] When a contour point C_i satisfies either condition (a) or (b), a label — is assigned to C_i.

(a) $(S_i<0)\&(S_{i-1}<0 \text{ or } S_{i+1}<0)$

(b) $(S_i<0)\&(S_{i-1}>0)\&(S_{i+1}>0)$

$$\&\,(F_i \neq FF)\&(DD_{i,i+1} \geq k_1)$$

Example of assignment of character shown in Fig. 3(b) with + and — according to the definitions is shown in Fig. 6.

Thus label vertices are used to extract concave and convex portions. For the methods based on concavities and convexities, the locations of the start and end vertices of the concave and convex portions have important effects for the extraction of features. To achieve insensitive locations of the start and end vertices, the following condition (A) is introduced.

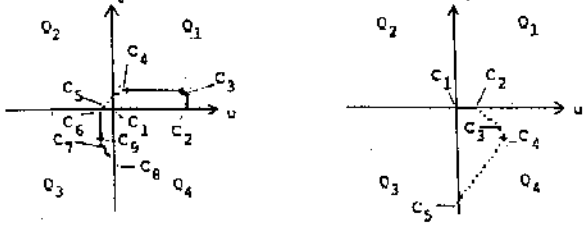


Fig. 7 Examples of arcs drawn in $u-v$ plane.
(a) Convex arc of character shown in Fig. 3(b).
(b) Concave arc of character shown in Fig. 3(b).

(A) When the digital distance $DD_{i,j}$ of adjacent + and — vertices (or — and + ones), C_i and C_j, is equal to one, the vertex or both vertices having the sign +1 (or —1) are deleted.

For avoiding a disadvantage which occurs when two concave vertices appear between convex ones, the portion which connects the two vertices are defined as an arc[9], a concave and a convex arc are defined as follows[6].

When there are successive + vertices between two — vertices, the successive + vertices including the two — vertices are called a convex arc, and a concave arc is the successive — vertices existing between two + vertices.

3.2 Features of Characters

Geometric features of arcs, such as length of an arc, distance between the start and end vertices of an arc, etc, are utilized in various studies[6]–[8], because they can be extracted easily. The main disadvantage of these features is that local details of arcs are lost. In the paper[6], several geometric features are adopted effectively for the recognition of 128×128 dots of Thai characters. However, the features are insufficient for classifying small and similar characters. For the recognition of low resolution Thai characters, more effective features are required.

A line approximation is applied to Thai characters by eliminating vertices whose vertex angles are nearby 180 degrees, and by merging adjacent vertices having very short distance. Then, features of characters are extracted as follows.

Since Thai characters have many categories, global features of contours, i.e., the number of holes H, that of disjoint regions R, the total number of concavities and convexities Z, and the number of contour pixels (perimeter of character) N are utilized for the classification.

Thai characters are composed of arcs of complex structures and local information of arcs has important meanings for the classification of similar arcs. A simple extraction of local features of arcs is introduced for the recognition of 50×50 dots of Thai characters as follows.

Let $C_j=(C_{xj}, C_{yj})$ $; j=1\sim J$ be a sequence of + and — vertices belonging to an arc in the $x-y$ plane. The vertices are transformed to a $u-v$ plane (as shown in Fig. 7), in the fashion that the start vertex of the arc C_1 is located at the original point of the plane, and the transformation is supposed to be made without changing of the original shape. Further, for avoiding the rotation problem, the first segment C_1-C_2 of the arc is located on the positive u-axis. Fig. 7 shows all arcs of the character shown in Fig. 3(b) drawn in the $u-v$ plane according to the above conditions.

Let $C_j=(C_{uj}, C_{vj})$ $; j=1\sim J$ be a sequence of + and — vertices belonging to an arc in the $u-v$ plane. Then, length of arc L is defined as follows.

$$L=\sum_{j=1}^{J-1} D_{j,j+1}$$

Table 1 Feature values of character shown in Fig. 3(b).

(H = 0; R = 0; Z = 2; N = 81)

	SS	E1	E2	E3	E4	L	NU	LU_1	LU_2	LU_3	LU_4	NV	LV_1	LV_2	LV_3	LV_4
A	+	1	1	1	1	74	4	26	36	7	5	3	45	24	5	0
B	-	0	0	1	1	38	2	17	21	0	0	1	38	0	0	0

where; A: the convex arc portion.

B: the concave arc portion.

where, $D_{i,j}=\sqrt{(C_{ui}-C_{uj})^2+(C_{vi}-C_{vj})^2}$

Define;

$$dU_{j,j+1}=C_{u,j+1}-C_{uj},\ dV_{j,j+1}=C_{v,j+1}-C_{vj}$$

The values of $dU_{j,j+1}$ are calculated at every vertex ($j=1\sim J-1$). The vertices at which sign of $dU_{j,j+1}$ changes whether from positive (+) to negative (−) or vice -versa are called segmenting points of arc along u-axis (SU). Further, arc portions segmented by SU are called segmented portions of arc along u-axis (PU). In order to obtain good locations of SU, it is noted that a vertex j is selected as SU when j indicates changing of sign of $dU_{j,j+1}$, and the segment C_j-C_{j+1} forms angle with v-axis larger than 10 degrees.

In the similar way, segmenting points of arc along v-axis (SV) and segmented portions of arc along v-axis (PV) are also extracted. In the example shown in Fig. 7(a), vertices C_3, C_7, C_9 are chosen as SU, C_4, C_8 as SV, portions C_1-C_3, C_3-C_7, C_7-C_8, C_8-C_9 as PU and C_1-C_4, C_4-C_8, C_8-C_9 as PV. As for Fig. 7(b), vertex C_4 is chosen as SU, portions C_1-C_4, C_4-C_5 as PU and C_1-C_5 as PV.

The following parameters are utilized to describe features of arc in $u-v$ plane. For facilitating the explanation, $u-v$ plane is divided into four regions, i.e., Q_1 (having region $u\geq 0$, $v>0$), Q_2 ($u<0$, $v>0$), Q_3 ($u<0$, $v\leq 0$) and Q_4 ($u\geq 0$, $v<0$).

(1) SS which takes sign + for convex arc and takes sign − for concave arc.

(2) En ; $n=1, 2, 3, 4$ which takes value 1 when arc portion exists in Q_n and takes value 0 when not exists.

(3) L length of arc.

(4) NU number of segmented portions along u-axis.

(5) LU_{ku} ; $ku=1\sim NU$ length of a semgnted portion along u-axis.

(6) NV number of segmented portions along v-axis.

(7) LV_{kv} ; $kv=1\sim NV$ length of a segmented portion along v-axis.

Note that the parameters SS, $E1$, $E2$, $E3$, $E4$ are logical features, where L, NU, LU_{ku}, NV, LV_{kv} are integer features. A character II is represented by its features as follows.

$$II=\{H_{II}, R_{II}, Z_{II}, N_{II}, (SS_i, E1_i, E2_i, E3_i, E4_i, L_i, NU_i, (LU_{ku_i})_{ku_i=1}^{NU_i}, NV_i, (LV_{kv_i})_{kv_i=1}^{NV_i})_{i=1}^{Z_{II}}\} \quad (3)$$

Hereafter, II is called a character.

For example, values of features of character shown in Fig. 3(b) are given in Table 1.

As for the characters being composed of only convex portions such as ○, ⌒ and ', the following feature is utilized for the classification.

The number of vertices of the approximated character whose vertex angles θ_i are smaller than a threshold θ_k is taken up as feature parameter.

4. Matching Algorithm

The matching method is based on that in the Ref. (6), and it is modified for efficient use of the extracted features. A model MM (model making is described later) is represented by its features as follows.

$$MM=\{H_{MM}, R_{MM}, Z_{MM}, N_{MM}, (SS_j, E1_j, E2_j, E3_j, E4_j, L_j, NU_j, (LU_{ku_j})_{ku_j=1}^{NU_j}, NV_j, (LV_{kv_j})_{kv_j=1}^{NV_j})_{j=1}^{Z_{MM}}\} \quad (4)$$

First, the matching of H, R and Z between a model and an input data is tested. If they match, and the percentage of the difference of the number of contour pixels N is smaller than a threshold (denoted by DIFN), the matching of arcs is tested as follows.

(1) The matching of SS, $E1$, $E2$, $E3$, $E4$, L, NU and NV between the input arc and the model arc is made. If they match, a similarity between the arcs is calculated as follows.

(2) The following scores T_{ku_i} and link levels R_{ku_i} are given to the i-th arc of II and the j-th arc of MM according to the following condition

(a) or (b).

(a) $T_{ku_i}=1$, $R_{ku_i}=2$

for $|LU_{ku_i}-LU_{ku_j}|\leq \varepsilon_1$

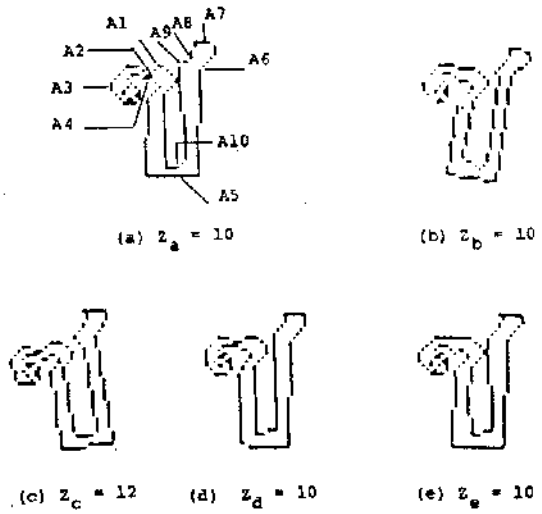


Fig. 8 An example of characters belonging to the same category. Z_a, Z_b, Z_c, Z_d and Z_e is the number of concavities and convexities of characters shown in Fig. 8 (a), (b), (c), (d) and (e) respectively. $A1, A2, \cdots, A10$ are arcs constituting Fig. 8(a).

(b) $T_{ku_i} = 1, \quad R_{ku_i} = 1$

for $\left| LU_{ku_i} - LU_{ku_j} \right| \leq \mathcal{C}_2$

where, ku_i and ku_j vary from 1 to NU_i

$\mathcal{C}_1$ and $\mathcal{C}_2$ are thresholds.

Similary, T_{kv_i} and R_{kv_i} are given to the arc pair according to the condition (c) or (d).

(c) $T_{kv_i} = 1, \quad R_{kv_i} = 2$

for $\left| LV_{kv_i} - LV_{kv_j} \right| \leq \mathcal{C}_1$

(d) $T_{kv_i} = 1, \quad R_{kv_i} = 1$

for $\left| LV_{kv_i} - LV_{kv_j} \right| \leq \mathcal{C}_1$

where, kv_i and kv_j vary from 1 to NV_i.

By using $T_{ku_i}, T_{kv_i}, R_{ku_i}$ and R_{kv_i}, a similarity between arcs $SA_{i,j}$ is determined as follows.

$$SA_{i,j} = \Big(\sum_{ku_i=1}^{NU_i} (T_{ku_i} + R_{ku_i} / 2) + \sum_{kv_i=1}^{NV_i} (T_{kv_i} + R_{kv_i} / 2) \Big) / NUV_i \qquad (5)$$

where, $NUV_i = NU_i + NV_i$. $SA_{i,j}$ takes a value in the range from 0.0 to 2.0, however, when $SA_{i,j}$ has value smaller than a threshold (denoted by S_{ak}), zero is given to it.

When a rotated character is inputted to a computer, the arcs constituting the character appear indifferent order from a model. For avoiding the problem, first, a pair of the most similar arcs is detected from II and MM[6]. Then, the pair is utilized as standard for determining the matching of the next arc pairs. By calculating the similarity between

Table 2 Feature values of the model for characters shown in Fig. 8.

(H = 1; R = 0; Z = 10; N = 169)

	SS	E1	E2	E3	E4	L	NU	LU_1	LU_2	LU_3	NV	LV_1	LV_2	LV_3
A1	+	1	0	0	0	13	2	9	4	0	1	13	0	0
A2	-	0	0	0	1	6	1	6	0	0	1	8	0	0
A3	+	1	1	0	0	28	3	12	11	5	2	19	10	0
A4	-	0	0	1	1	13	2	5	8	0	2	12	2	0
A5	+	1	1	0	0	62	2	23	39	0	2	35	26	0
A6	-	0	0	0	1	33	1	33	0	0	1	33	0	0
A7	+	1	0	0	0	16	2	9	7	0	2	14	2	0
A8	-	0	0	0	1	8	1	8	0	0	1	8	0	0
A9	+	1	0	0	0	27	1	27	0	0	1	27	0	0
A10	-	0	0	0	1	50	2	28	22	0	2	40	10	0

Table 3 Feature values of character shown in Fig. 8(a).

(H = 1; R = 0 ; Z = 10; N = 168)

	SS	E1	E2	E3	E4	L	NU	LU_1	LU_2	LU_3	NV	LV_1	LV_2	LV_3
A1	+	1	0	0	0	13	2	9	4	0	1	13	0	0
A2	-	0	0	0	1	7	1	7	0	0	1	7	0	0
A3	+	1	1	0	0	28	3	10	12	6	2	17	11	0
A4	-	0	0	1	1	18	2	6	12	0	2	14	4	0
A5	+	1	1	0	0	62	2	23	39	0	2	36	26	0
A6	-	0	0	0	1	32	1	32	0	0	1	32	0	0
A7	+	1	0	0	0	17	2	9	8	0	2	13	4	0
A8	-	0	0	0	1	8	1	8	0	0	1	8	0	0
A9	+	1	0	0	0	27	1	27	0	0	1	27	0	0
A10	-	0	0	0	1	51	2	29	22	0	2	47	4	0

every determined arc pair, the similarity between characters $SC_{II,MM}$ is calculated as follows.

$$SC_{II,MM} = \sum_{i,j} (SA_{i,j} \times 100) / (2 \times Z) \ (\%) \qquad (6)$$

The model MM whose $SC_{II,MM}$ is maximum value and larger than a threshold (denoted by S_{ck}) is selected as the identified character with the input one. Else, the input data is rejected. However, the matching is made again for the rejected character, by detecting every similar arc pair from the character and a model, and by calculating similarities between arcs and similarities between characters.

Every pair of arc i and arc j satisfying the following conditions is decided to be similar.

(a) $SA_{i,j} \geq S_{ak}$.

(b) $\{ i, j \mid \max_{j=1}^{Z_{MM}} (SA_{i,j} + SA_{i-1,j-1} + SA_{i+1,j+1}) \}$

where, Z_{MM} is the total number of concavities and convexities of the model.

5. Model Making

In the paper[6], a simple model making has been proposed, in which a single model is generated for each category by averaging feature values of characters belonging to the same category. As shown in Fig. 8, characters belonging to the same category have various numbers of arcs Z. Moreover, several characters having the same number of Z, such as characters shown in Fig.8(a) and Fig. 8(b) or Fig. 8(a) and Fig. 8(e), are constituted of some different arcs. To deal with such characters the following model making is introduced.

For each category, characters which have the same values of H, R, Z, and are the majority group in the category are chosen for making model. The most similar arc pairs among the chosen characters are detected. The arc pairs are used as standard for calculating similarities between arcs $SA_{i,j}$ of the characters. The features values of every arc pair having $SA_{i,j}$ value larger than a threshold are preserved. The preserved values are averaged to make the model.

Let II_m; $m=1\sim M$, be characters belonging to a category (where, M is the number of those characters).

(Step 1) Set $m=1$.

(Step 2) If $m>M$, go to step 10.
Preserve feature values of II_m and set $k=1$

where, k is a parameter for counting characters used for model making.

(Step 3) Set $m1=1$.

(Step 4) If $m1>M$, then go to step 9.

(Step 5) If $m1=m$, then increment $m1$ and go to step 4.

(Step 6) If II_m and II_{m1} have the same values of H, R, Z, and if a pair of the most similar arcs between II_m and II_{m1} can be found, go to step 7. Else increment $m1$ and go to step 4.

(Step 7) Set $k=k+1$. Suppose the p-th arc of II_m is the most similar arc to the q-th arc of II_{m1}. Let $i=p$, $j=q$

(Step 8) Calculate the similarity between arcs $SA_{i,j}$ of II_m and II_{m1}. If $SA_{i,j}$ is larger than a threshold (denoted by S_a), preserve feature values of the j-th arc of II_{m1}.

Repeat step 8 procedure by executing every pair of arcs constituting II_m and II_{m1}. Increment $m1$ and go to step 4.

(Step 9) If $k<$ He, increment m and go to step 2. Else go to step 11.

where, He is a threshold (in the experiment, He $=M/2+1=3$).

(Step 10) Set He = He -1, and to to step 1.

(Step 11) Repeat the following procedure (for $i=1\sim Z_{II_m}$).
Average the preserved feature values of arcs which correspond to the i-th arc of II_m. Set the average values to the feature values of the i-th arc of the model of the category.

The process has finished. In Table 2, feature values of the model for the characters shown in Fig. 8 are given. In this example, the character shown in Fig. 8(a) whose feature values are given in Table 3, is used as standard for making the model.

Table 4 Values of thresholds.

$k_1 = 2$ for $|S_i| = 1$
$k_1 = 1$ for $|S_i| > 1$
$k_2 = 3$
$\theta_k = 120°$
DIFN = 50%
$\xi_1 = 3$; $\xi_2 = 5$ for LU_{ku_j} (or LV_{kv_j}) ≤ 30
$\xi_1 = 5$; $\xi_2 = 7$ for LU_{ku_j} (or LV_{kv_j}) ≤ 50
$\xi_1 = 7$; $\xi_2 = 9$ for LU_{ku_j} (or LV_{kv_j}) > 50
$S_{ak} = S_a = 1.0$
$S_{ck} = 60\%$

Table 5 Result of the detection of the most similar arcs.

$SA_{A1,A1} = 2.0$; $SA_{A10,A10} + SA_{A1,A1} + SA_{A2,A2} = 5.4$
$SA_{A1,A7} = 2.0$; $SA_{A10,A6} + SA_{A1,A7} + SA_{A2,A8} = 4.0$
$SA_{A1,A3} = 0.0$; $SA_{A1,A5} = 0.0$; $SA_{A1,A9} = 0.0$
$SA_{A2,A2} = 2.0$; $SA_{A1,A1} + SA_{A2,A2} + SA_{A3,A3} = 6.0$
$SA_{A2,A8} = 2.0$; $SA_{A1,A7} + SA_{A2,A8} + SA_{A3,A9} = 4.0$
$SA_{A2,A4} = 0.0$; $SA_{A2,A6} = 0.0$; $SA_{A2,A10} = 0.0$

where, $SA_{Ai,Aj}$ means the similarity of arcs between arc Ai of the input data and arc Aj of the model. In this example, arcs A2,A2 are the pair of the most similar arcs.

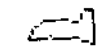
6. Experimental Results

Sixty nine Thai characters were inputted to a mini-computer (OKITAC-4300b, 32kW, 16 bits/W), as a binary data of 50×50 dots. In the process of taking the characters into the computer, they are rotated with five kinds of arbitary angles. In total, 345 data were used for the experiments. An experimental example of matching between the data shown in Fig. 8(a) and its model is shown, by the use of threshold values given in Table 4, as follows. After the matching of H, R, Z and N, the similarities between arcs are calculated and are used to detect a pair of the most similar arcs between characters. Several values of the similarity between arcs $SA_{i,j}$ and the result of the detection of the most similar arc pair between the data and the model are given in Table 5. As shown in Table 5, the matching is

limited to only the matching of similar arcs by using the extracted features, and the most similar arc pair is properly detected though the characters are composed of several similar arcs. After the detection of the most similar arc pair, the similarity between every arc pair and that between the characters $SC_{II,MM}$ are calculated, and their values are given in Table 6. As for the experimental results, all characters except two errors were correctly identified by the algorithm described above. In Fig. 9, the ill-identified characters are shown. The locations of the start and end vertices of arcs of the characters are mis-extracted by the algorithm.

Table 6 Values of $SA_{i,j}$ and $SC_{II,MM}$ (for character shown in Fig. 8(a) and its model.)

$SA_{A1,A1} = 2.0$; $SA_{A2,A2} = 2.0$; $SA_{A3,A3} = 2.0$;
$SA_{A4,A4} = 1.9$; $SA_{A5,A5} = 2.0$; $SA_{A6,A6} = 2.0$;
$SA_{A7,A7} = 2.0$; $SA_{A8,A8} = 2.0$; $SA_{A9,A9} = 2.0$;
$SA_{A10,A10} = 1.4$;
$SC_{11,MM} = 96.5\ \%$.



the data was identified as

(a)

the data was identified as

(b)

Fig. 9 Ill-identified characters.

7. Conclusions

In this paper, a recognition method of Thai characters based on local information of contours is described. In the method, noise pixels of character contours are eliminated by making use of Freeman chain code. Contour pixels which indicate directional change of chain code are given the values of the chain code and the values of + (or −) signs. The concavities and convexities of contours are then extracted by simple method. Simple local features of arc, both logical and integer features are adopted for the matching. Logical features are used to decrease the computation time of matching, and the integer features are utilized to calculate similarities between arcs constituting characters and the similarity between characters.

The proposed method was applied efficiently to characters rotated with arbitrary angles. Further, when rotated characters are inputted to a computer, sizes as well as feature values of characters belonging to the same category vary with the rotations. However, the method is applicable to these different size characters, because a model is made by averaging feature values of characters belonging to the same category, and because thresholds $\mathcal{E}_1$ and $\mathcal{E}_2$ are used for flexibility in matching.

The proposed matching process has the capability of matching the input character with a model having different dimensions of features. Further, models are made from input data, and only single model is made for each category. The recognition method was applied to a variety of 345 printed Thai characters rotated with five kinds of arbitrary angles, and 99.4% of recognition rate has been achieved.

References

(1) Agui, T., Nakajima, M., Kim, T.K. and Takahashi, E.T.: "A method of recognition and representation of Korean characters by tree grammars", IEEE Trans. Pattern Anal. & Mach. Intell., PAMI-1, 3, pp. 245—251 (1976).

(2) Badie, K. and Shimura, M.: "Machine recognition of Arabic handprinted scripts", Trans. IECE Japan, E65, 2, pp. 107—114 (Feb. 1982).

(3) Kushnir, M., Guo, B.L. and Matsumoto, K.: "The recognition of handwritten Hebrew characters by the double inclusive matching method", Trans. IECE Japan, J65-D, 8, pp. 1011—1017 (Aug. 1982).

(4) Matsuse, T., Minuo, M. and Sakai, T.: "Classification of 9 kinds of characters based on the statistics of legal patterns", Paper of Technical Group, TG PRL 83-9, IECE Japan (1983)

(5) Agui, T., Hiranvanichakorn, P. and Nakajima, M.: "Recognition of printed Thai characters", Paper of Technical Group, TG PRL 81-30, IECE Japan (1981).

(6) Hiranvanichakorn, P., Agui, T. and Nakajima, M.: "A recognition method of Thai characters", Trans. IECE Japan, E65, 12, pp. 737—744 (Dec. 1982).

(7) Yamamoto, K. and Mori, S.: "Recognition of handprinted characters by an outmost point method", Pattern Recognition, 12, 4, pp. 229—236 (1980).

(8) Yamamoto, K.: "Recognition of handprinted Hiragana characters by concave and convex features and automatically merging the dictionary", Trans. IECE Japan, J65-D, 6, pp. 774—781 (June 1982).

(9) Pavlidis, T. and Ali, F.: "Computer recognition of handwritten numerals by polygonal approximations", IEEE Trans. Syst., Man & Cybern., SMC-5, 8, pp. 610—615 (1975).

(10) Pavlidis, T. and Ali, F.: "A hierarchical syntactic shape analyzer", IEEE Trans. Pattern Anal. & Mach. Intell., PAMI-1, 1, pp. 2—9 (1979).

(11) Pavlidis, T.: "The use of a syntactic shape analyzer for contour matching", IEEE Trans. Pattern Anal. & Mach. Intell., PAMI-1, 3, pp. 307—310 (1979).

(12) Freeman H.: "Computer processing of line drawing images", Comput. Surv., 6, pp. 57—97 (March 1974).

(13) Freeman, H. and Davis, L.S.: "A corner-finding algorithm for chain-coded curves", IEEE Trans. Comput., C-26, pp. 297-303 (March 1977).

(14) Wallace, T.P., Mitchell, O.R. and Fukunaga, K.: "Three dimensional shape analysis using local shape descriptors", IEEE Trans. Pattern Anal. & Mach. Intell., PAMI-3, 3, pp. 310—323 (May 1981).

Appendix

The contour pixels which indicate directional changes of Freeman chain code, and are not regarded as noise pixels, are detected by this processing (see section 3.1). The idea

is that after the elimination of noise pixels, the remaining pixels on a line have the same chain code value.

Let $F_n \in \{0 \sim 7\}$ be Freeman chain code value of an element P_n ($n = 1 \sim N$; where, N is the number of contour pixels of a character). Assume $P_{N+1} = P_1$, $P_{N+2} = P_2$.

(Step 1) Set $n = 1$ and calculate F_n of P_n. Set chain code direction $F_D = F_n = F_1$

(Step 2) Increment n and calculate F_n. If $n = N + 2$, go to step 9. If $F_n = F_D$, repeat this step.

(Step 3) Set $k = -1$.

(Step 4) Increment k. If $k > T1$, go to step 8.

(Step 5) Set $m = 1$.

(Step 6) Increment m. If $m > T2$, go to step 4.

If ($P_{x,n+m} \neq P_{xn} + k \times t_x$)

or ($P_{y,n+m} \neq P_{yn} + k \times t_y$),

repeat this step.

(Step 7) Set $n = n + m - 1$ and go to step 2.

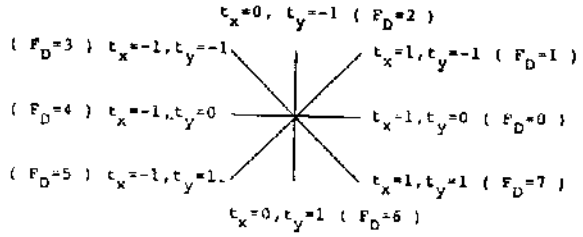


Fig. A-1 Values of chain code parameters t_x and t_y

(Step 8) Preserve P_n and F_n. Set $F_D = F_n$, and go to step 2.

(Step 9) If P_{N+1} is already preserved, then preserve P_1 and F_1.

where, $T1$ and $T2$ are thresholds (in the experiment, $T1 = 2$, $T2 = 3$), and t_x and t_y are chain code parameters along x and y-axises and their values are determined by the value of F_D shown in Fig. A-1.

A Recognition Method of Handprinted Thai Characters by Local Features

Pipat HIRANVANICHAKORN, Takeshi AGUI *and* Masayuki NAKAJIMA, *Members*

UDC 003.337.42.052 : 159.937.52 : 681.32.05

SUMMARY A recognition method of handprinted Thai characters by using local features is described. In the method, Freeman chain code and directional differences of contour tracing are utilized for extracting concavities and convexities of characters. Several local features are used to calculate similarities between arc portions and similarity between characters. Similar arcs are detected from characters of different categories to make a dictionary of arcs. Then, a dictionary of characters containing lists of names of character arcs is made to obtain a compact dictionary of models. By applying the method to 69 categories (828 data) of Thai characters, a recognition rate of 99.3% for learning data, and a recognition rate of 88.9% for test data have been obtained.

1. Introduction

Up to now, most character recognition techniques are proposed for alphanumerics, Japanese characters (Katakana and Hiragana) and Chinese ones. Recently, the recognition studies of Korean[1], Arabic[2], Tamil characters[3] and others[4] have been made, because each character has different properties and features, and there are many interesting and challenging problems to be dealt with. A few works have been done for recognizing machine-printed Thai characters[6],[7], because the characters have many complicated and similar shapes and they are mainly composed of curves. The study of handprinted Thai characters is not almost made. Owing to the deformation of image of a character caused by distortion, style variation and rotation[5], the recognition of handprinted characters is much difficult. In this paper, a recognition method of handprinted Thai characters is described.

In a paper[11], a successful recognition method of Hiragana has been proposed, in which concave and convex portions of characters are extracted by detecting start and end points of the portions, and features of concavities and convexities are utilized to calculate distance among characters. A main disadvantage of the method is that local details of arcs are lost. As Thai characters have many similar characters, and arcs constituting characters have complex structures, the local details of arcs hold important meanings for classifying similar arcs. Further, though a model is made by merging matched arcs of the characters of the same category, several models are generated for a category to solve the problem of rotation.

Manuscript received May 7, 1984.
Manuscript revised September 6, 1984.

The authors are with Imaging Science and Engineering Laboratory, Tokyo Institute of Technology, Yokohama, 227 Japan.

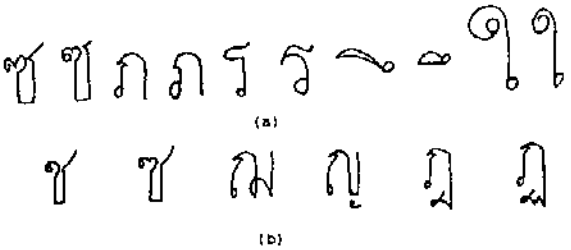


Fig. 1 (a) Examples of handprinted Thai characters. (b) Examples of similar characters.

In a paper[7], a successful recognition method based on concavities and convexities of contours has been proposed for printed Thai characters rotated with arbitrary angles. The method was applied to handprinted Thai characters and the recognition rate of 70% was obtained. The feature extraction, in which the first segment of arc is used as standard for solving the rotation problem, is affected by the distortion of handprinted characters shown in Fig. 1(a). Moreover, the model making, in which a single model is generated for each category, does not work well for handprinted Thai characters having curve-like structure. In the methods[7],[11], each model has individual feature values.

As a modification of the methods[6],[7], this paper proposes a simple and effective recognition method of handprinted Thai characters, in which an effective extraction of features, a model making and a matching process are dealt with. In the method, Freeman chain code and directional differences of contour tracing are utilized for extracting concavities and convexities of characters. Several local features of arc are extracted by making use of the average direction of all segments of the arc. A matching method which can tolerate size variation and rotation is described. Further, as shown in Fig. 1(b), several Thai characters have similar arc portions. Therefore, the idea of using common arc portions among different models is considerable. In this paper, a simple model making is introduced, in which similar arcs are detected from characters of different categories to make a dictionary of arcs. Further, a dictionary of characters which has elements being names of characters and lists of names of arcs of characters, is made to obtain a compact and effective dictionary of models. By applying the method to 828 data of handprinted Thai characters, a recognition rate of 99.3% for learning data, and a recognition rate of 88.9% for test data have been obtained.

2. Thai Characters

Thai characters are composed of forty four consonant characters, thirty two vowel ones, four tonal symbols and four special characters. Several vowel characters can be decomposed into fundamental ones. As the result, sixty nine fundamental characters are obtained. In Fig. 2, a set of fundamental Thai characters is shown.

Fig. 2 A set of Thai characters.

3. Feature Extraction Algorithm

3.1 Extraction of Concavity and Convexity

Concave and convex portions of figures can be extracted by labeling a point as — or + at which counterclockwise boundary tracing turns to the right or left, respectively. In the paper[7], contour pixels which indicate directional changes of Freeman chain code are detected. The pixels are then coded according to directional differences. The coded pixels are utilized to detect + and — vertices which indicate convexities and concavities of contours. The method is applied to handprinted Thai characters to extract concavities and convexities of them. An example of extracting + and — vertices of contours of a Thai character shown in Fig. 3(a) is depicted in Fig. 3(b). Then, + and — vertices are used to extract convex and concave arcs of a character. In a paper[12], a concave arc is defined as an uninterrupted sequence of concave vertices. Accordingly, when only two concave vertices appear between convex ones, the portion is defined to be a concave arc in spite of being a straight line. For avoiding the problem, a convex arc and a concave one are defined as follows.

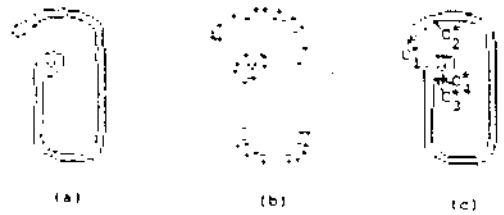


Fig. 3 An example of extracting concavities and convexities.
(a) Original character.
(b) Character with labels + and —.
(c) Extraction of concave and convex arcs of character.

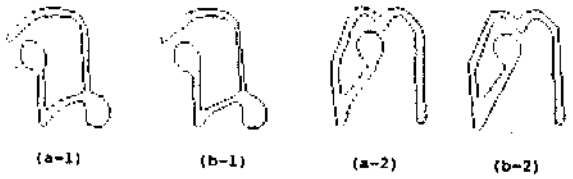


Fig. 4 Examples of line approximation.
(a) Original characters.
(b) Approximated characters.

When there are successive + vertices between two — vertices, the successive + vertices including the two — vertices are called a convex arc, and a concave arc is the successive — vertices existing between two + vertices.

Accordingly, the start and end vertices of a convex (or concave) arc are — (or +) vertices. Further, adjacent concave and convex arcs have a common portion. As shown in Fig. 3(c), the segment passing points C_3^*, C_4^*, C_1^* and C_2^* is regarded as convex arc, and the segment passing points C_1^*, C_2^*, C_3^* and C_4^* is concave arc.

3.2 Line Approximation

Some Thai characters have low curvature portions which give different numbers of arcs to the characters of the same category. For avoiding the problem, a low curvature portion is approximated by straight line segment. Further, a curve is also approximated by eliminating (+ or —) vertices whose vertex angles are nearby 180 degrees, and by merging adjacent vertices having very short distance. In Fig. 4, examples of characters after line approximation are shown with their original ones.

3.3 Features of Characters

In the recognition of handprinted characters, the selection of effective features is somewhat difficult, especially, for Thai characters having many complicated and similar shapes.

In the papers [9], [13], features such as kinds of arcs (sharp protrusion, convex arc, etc.), sizes of arcs, directions of arcs are adopted for the recognition of alphanumerics and simple shapes. However, the feature extraction algorithm is complex. Further, it is difficult to apply the method to Thai characters, because the characters are mainly composed of curves and several arcs constituting characters have complex structures such as spiral ones.

The number of holes H, that of disjoint regions R, perimeters of holes $LH_h : h = 1 \sim H$, the total number of concavities and convexities Z, and the number of contour pixels (perimeter of character) N are utilized for the classification, because Thai characters have many categories.

In the paper [7], several local features of arcs of characters are adopted effectively for the recognition of printed Thai characters. In the method, direction of the first segment of arc is utilized as standard for solving the rotation problem. Unfortunately, the features extracted by the method are affected by the distortion of handprinted characters. The extraction method is modified for the recognition of handprinted Thai characters as follows.

Let $P_j^* = (P_{xj}^*, P_{yj}^*) ; j = 1 \sim J$ be a sequence of + and − vertices belonging to an arc in the x-y plane, and let AN_j be direction of segment $P_j^* - P_{j+1}^*$ (i.e. angle with x axis). An average direction A_{ave} of all segments of the arc is defined as follows.

$$A_{ave} = \left(\sum_{j=1}^{J-1} AN_j \right) \Big/ (J-1) \qquad (1)$$

For facilitating the above calculation, the value of AN_j is given in counterclockwise manner (i.e., $AN_j >= 0$ and $AN_{j+1} > AN_j$) for a convex arc, and is given in clockwise manner (i.e., $AN_j <= 0$ and $AN_{j+1} < AN_j$) for a concave one. As examples, the calculated value of A_{ave} of the convex arc of character shown in Fig. 3(a) is 27 degrees (i.e., the value 0, 45, 90, 135, 180, 225, 252, 270, 315, 337, 360, 374, 405, 495, 513, 534, 557, 578, 630, 675, 720, 757 are substituted for AN_j in (1)) and that of the concave arc is −156 degrees (i.e., the value −323, −360, −374, −432, −452, −523, −576, −603, −632, −675 and −720 are substituted for AN_j). Then, the vertices of the arc are transformed to a u-v plane (as shown in Fig. 5), in the fashion that the start vertex C_1 (when the arc is concave, the first + vertex is called the start vertex and the other + vertex is called the end vertex, and when the arc is convex, the first − vertex is called the start vertex and the other − vertex is called the end vertex) of arc is located at the original point of the plane, and the average direction A_{ave} lies along the positive u-axis. Fig. 5 shows all arcs of the character shown in Fig. 3(a) drawn in the u-v plane according to the above conditions.

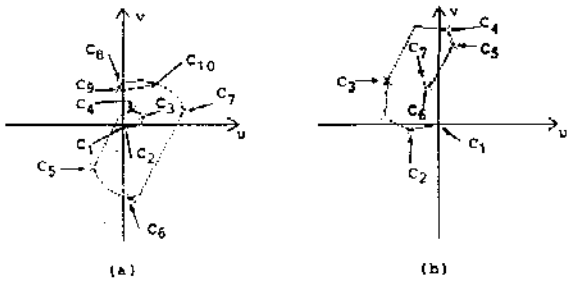


Fig. 5 Example of arcs drawn on the u-v plane.
(a) Convex arc of character shown in Fig. 3(a).
(b) Concave arc of character shown in Fig. 3(a).

Table 1 Feature values of character shown in Fig. 3(a).

(H = 1; R = 0; Z = 2; N = 328; LH_1 = 9)

	SS	L	NU	LU_1	LU_2	LU_3	LU_4	LU_5	NV	LV_1	LV_2	LV_3	LV_4	LV_5
A	+	216	5	14	49	88	48	17	5	2	19	70	104	20
B	−	144	3	51	63	30	0	0	4	15	88	39	2	0

where; A: the convex arc portion.
B: the concave arc portion.

Let $P_j = (P_{uj}, P_{vj}) ; j = 1 \sim J$ (where, $P_{uj} = ((P_{xj}^* - P_{x1}^*)\cos(A_{ave}) + (P_{yj}^* - P_{y1}^*)\sin(A_{ave}))$ and $P_{vj} = (-(P_{xj}^* - P_{x1}^*)\sin(A_{ave}) + (P_{yj}^* - P_{y1}^*)\cos(A_{ave}))$, and (P_{x1}^*, P_{y1}^*) is the co-ordinate of the start vertex of arc in the x-y plane) be a sequence of + and − vertices belonging to an arc in the u-v plane.

Define;

$$dU_{j,j+1} = P_{u,j+1} - P_{uj}, dV_{j,j+1} = P_{v,j+1} - P_{vj}. \qquad (2)$$

The value of $dU_{j,j+1}$ is calculated at every vertex ($j = 1 \sim J-1$). The vertices at which sign of $dU_{j,j+1}$ changes whether from positive to negative or vice-versa are called segmenting points of arc along the u-axis (SU). Further, arc portions segmented by SU are called segmented portions of arc along the u-axis (PU). In the similar way, segmenting points (SV) and segmented portions (PV) of arc along the v-axis are extracted. In the method, the affection of the detection of SU and SV which is caused by rotation is reduced by making use of A_{ave} as standard for transforming arc vertices from x-y plane to u-v plane. In the example shown in Fig. 5(a), vertices C_3, C_5, C_7 and C_9 are chosen as SU, C_2, C_4, C_6 and C_8 as SV, portions $C_1 - C_3$, $C_3 - C_5$, $C_5 - C_7$, $C_7 - C_9$ and $C_9 - C_{10}$ as PU, and $C_1 - C_2$, $C_2 - C_4$, $C_4 - C_6$, $C_6 - C_8$ and $C_8 - C_{10}$ as PV (where, C_1 and C_{10} indicate the positions of the start vertex and the end vertex of the arc, respectively). As for Fig. 5(b), vertices C_3 and C_5 are chosen as SU, C_2, C_4 and C_6 as SV, portions $C_1 - C_3$, $C_3 - C_5$ and $C_5 - C_7$ as PU, and $C_1 - C_2$, $C_2 - C_4$, $C_4 - C_6$ and $C_6 - C_7$ as PV (where, C_1 and C_7 indicate the positions of the start vertex and the end vertex of the arc, respectively).

The following parameters are utilized to describe features of arc in the $u-v$ plane.

(1) SS which takes sign + for convex arc and takes sign − for concave arc.
(2) L length of arc.
(3) NU number of segmented portions along the u-axis.
(4) LU_{ku} ; $ku=1\sim NU$ length of a segmented portion along the u-axis.
(5) NV number of segmented portions along the v-axis.
(6) LV_{kv} ; $kv=1\sim NV$ length of a segmented portion along the v-axis.

As an example, values of features of character shown in Fig. 3(a) are given in Table 1.

It is noted that the extracted feature values are not affected by rotation. However, they vary with the data sizes. In this paper, the size variation problem is solved in the process of calculating the similarity of arcs described in Section 4.1 and Appendix.

As for characters shown in Fig. 6 which are composed of only convex portions, the following feature is utilized for the classification.

The number of vertices of the approximated character whose vertex angles are smaller than a threshold is taken up as the feature parameter.

4. Model Making

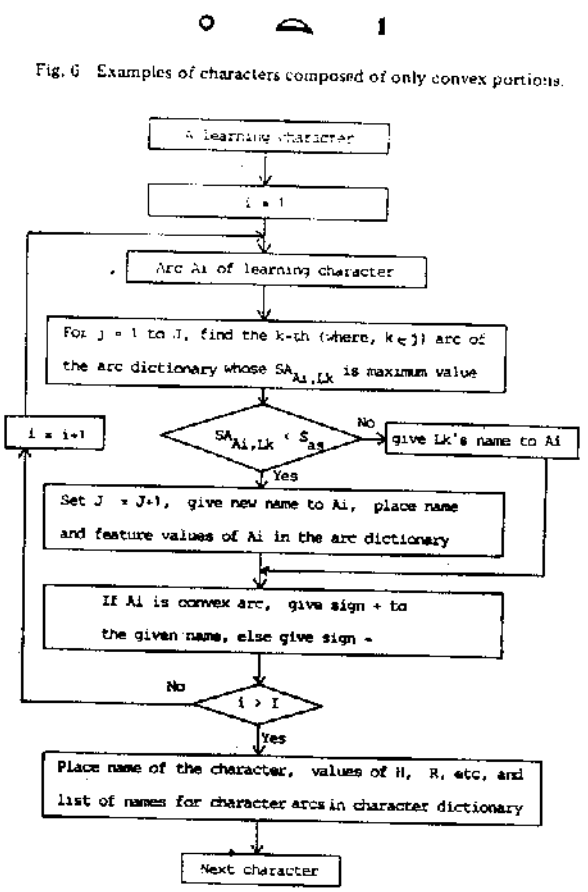
For using practically, an effective dictionary of models having compact size is required. In the paper[11], a recognition method of Hiragana has been proposed, in which a dictionary is made by merging similar arcs of characters of the same category. However, several models are generated for a category to solve the problem of rotation.

In the papers[6],[7], recognition methods of printed Thai characters have been proposed, in which a single model is automatically generated for each category. The methods have been applied effectively to characters rotated with arbitrary angles. However, it is difficult to apply the methods to handprinted Thai characters being mainly composed of curves, and having many similar characters.

In the methods[6],[7],[11], each model has individual feature values. In the paper[8], a compact library of models is made by constructing a library of features and a library of known objects. In this manner, it is possible to use common feature values for models of different categories. However, for the recognition of handprinted characters, the method needs a modification. As several Thai characters have similar arc portions, the idea of using common arc portions among different models is considerable.

4.1 Similarity of Arcs

The matching of two arcs i and j is made as follows. When the differences of numbers of segmented portions of arcs (NU, NV) between the two arcs are smaller than thresholds, a similarity between arcs $SA_{i,j}$ which has a value from 0.0 to 2.0 is calculated. A detail explanation of calculating $SA_{i,j}$ is given in Appendix. Further, when the value of $SA_{i,j}$ of the two arcs i, j is larger than a threshold (denoted by S_{as}), the two arcs i, j are decided as a similar arc pair.



J:number of arcs of arc dictionary, I:number of character arcs

Fig. 7 A flowchart of model making.

4.2 Dictionary of Arcs and Dictionary of Characters

In this method, the dictionary of models is divided into a dictionary of arcs and a dictionary of characters. The dictionary of arcs has elements being numbers of arcs, and each arc is described by a name of arc (alphanumeric is adopted in this experiment) and feature values of arc (except for the sign of arc, SS) The dictionary of characters has elements being numbers of models and each model is described by a name of character, values of H, R, N, LH_k and Z, lengths of all arcs of the character and a list of names of all arcs. A model is generated from a learning character as follows.

Each arc of the learning character is compared to all arcs of the dictionary of arcs in order to find out a similar

arc for the character arc. If a similar arc is found, the name of the arc is given to the character arc. Otherwise, a new name is given to the character arc. The given name and feature values of the arc are placed in the dictionary of arcs. After all arcs of the character have been named, a description of the character is formed and placed in the dictionary of characters.

Let $Ai\ ; i=1\sim I$ (where, I is the number of arcs) be a sequence of arcs constituting a learning character and $Lj\ ; j=1\sim J$ (where, J is the number of arcs) be an order of arcs contained in the dictionary of arcs. In Fig. 7, a flowchart of model making is shown.

(Step 1) Let $i=1$.

(Step 2) If $i>I$ then go to step 7.

(Step 3) For $j=1$ to J, calculate the similarity of arcs $SA_{Ai,Lj}$, and find out the k-th arc from the arc dictionary (where, k is a member of j) whose $SA_{Ai,Lj}$ is the maximum value. If $SA_{Ai,Lk}$ is smaller than a threshold (denoted by S_{ar}), then go to step 5.

(Step 4) Give the name of the k-th arc to the character arc. Go to step 6.

(Step 5) Set $J=J+1$. Give a new name to the character arc. Place the name and feature values (except for sign of arc, SS) of the character arc in the dictionary of arcs.

(Step 6) Give a sign + to the given name of the character arc in case that the arc is convex, else give a sign −. Set $i=i+1$, go to step 2.

(Step 7) Place the name of the character, feature values of H, R, N, LH_1 and Z, lengths of all character arcs and the list of names for character arcs in the dictionary of characters.

Fig. 8(a) and (b) show arcs of two similar characters which are named by the above process. Table 2(a) and (b) give feature values of the characters, respectively. Table 3 gives feature values of the dictionary arcs named to the character arcs.

4.3 Grouping of Characters

According to the process described above, a compact dictionary of models is obtained. However, when numerous learning characters are adopted for each category, the size of the dictionary of characters becomes large. For avoiding the problem, for each category, characters which have the same values of R and Z, and have high similarities among characters are grouped. The calculation of a similarity between characters is described in Section 5. A single model is generated for a group by averaging feature values of characters of the group. The model is then utilized to make the dictionary of models. As for example shown in Fig. 9, characters of the category are collected into three groups, i.e., characters shown in Figs. 9(a), 9(b) and 9(e) as one group, those shown in Figs. 9(c) and 9(f) as one and that shown in Fig. 9(d) as one.

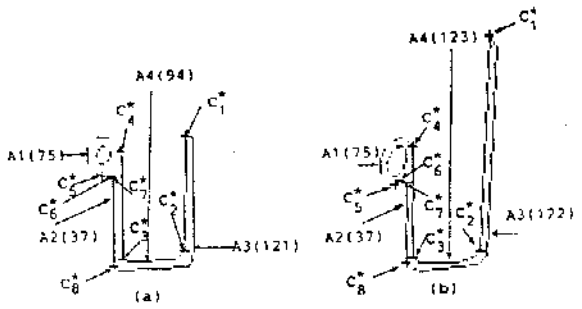


Fig. 8 An example of naming of arcs of similar characters.
(arc $A1$: The segment passing points C_3^*, C_4^*, C_5^* and C_6^*,
arc $A2$: The segment passing points C_3^*, C_4^*, C_7^* and C_8^*,
arc $A3$: The segment passing points C_1^*, C_2^*, C_1^* and C_2^*,
arc $A4$: The segment passing points C_1^*, C_2^*, C_3^* and C_4^*,
and the number in ( ) is the name of the arc of arc dictionary given to the character arc.)

Table 2 Feature values of characters shown in Fig. 8.

(H = 1; R = 0; Z = 4; N = 298; LH_1 = 21)

	SS	L	NU	LU_1	LU_2	LU_3	NV	LV_1	LV_2	LV_3
A1	+	87	3	56	25	6	2	66	21	0
A2	−	44	1	44	0	0	2	5	39	0
A3	+	185	3	42	89	54	3	80	54	51
A4	−	128	1	128	0	0	2	77	51	0

(a)

(H = 1; R = 0; Z = 4; N = 379; LH_1 = 31)

	SS	L	NU	LU_1	LU_2	LU_3	NV	LV_1	LV_2	LV_3
A1	+	87	3	55	29	3	2	70	17	0
A2	−	41	1	41	0	0	2	5	36	0
A3	+	267	3	39	133	95	3	73	99	95
A4	−	176	1	176	0	0	2	124	52	0

(b)

Table 3 Feature values of the arcs of arc dictionary named to the arcs of characters shown in Fig. 8.

ARC NO.	SS	L	NU	LU_1	LU_2	LU_3	NV	LV_1	LV_2	LV_3
37		31	1	31	0	0	2	7	24	0
75		39	3	24	12	3	2	31	8	0
94		104	1	104	0	0	2	62	42	0
121		185	3	42	89	54	3	80	54	51
122		267	3	39	133	95	3	73	99	95
123		176	1	176	0	0	2	124	52	0

5. Matching Process

In the papers[6],[7], matching methods by calculating similarities between arc portions, and similarity between a model and an input character, have been proposed successfully for printed Thai character. The methods are modified and applied to handprinted Thai characters as follows.

It is noted that models are placed in the dictionary of characters, and each model is described by a name of character, a list of names of arcs, etc. According to the list of names of arcs, feature values of all arcs of a model are obtained from the dictionary of arcs. Further, sign of the parameter SS is obtained from the sign + or − already given to the name of the arc.

The matching is made between an input character and all models of the dictionary. For decreasing the computation time, first, the matching of B, R and Z between a model MM and the input data II is tested. If they match, and the differences of the perimeter of character N, perimeters of holes LH_k are smaller than thresholds, the matching of arcs is tested as follows.

The matching of SS, NU and NV between the input arc i and the model arc j is tested. If they match, the difference between the ratio of the length of the input arc to that of the model arc, and the ratio of the average length of all input arcs to that of all model arcs is calculated. If the difference is smaller than a threshold, the similarity between the arcs $SA_{i,j}$, as described in Appendix, is calculated.

When a rotated character is input to a computer, the arcs constituting the character appear in different order from those of a model. By the use of $SA_{i,j}$, a pair of the most similar arcs is detected from the model and the character.

The most similar arcs are such i and j as satisfy either conditions (a) and (b) in the case that the number of concavities and convexities is smaller than ten, or conditions (c) and (d) else.

(a) $(SA_{i-1,j-1} > S_b) \,\&\, (SA_{i,j} > S_c) \,\&\, (SA_{i+1,j+1} > S_b)$

(b) $\left\{ i,j \;\middle|\; \max_{i=1}^{Z_{II}} \left(\max_{j=1}^{Z_{MM}} (SA_{i-1,j-1} + SA_{i,j} + SA_{i+1,j+1}) \right) \right\}$

(c) $(SA_{i-2,j-2} > S_b) \,\&\, (SA_{i-1,j-1} > S_b) \,\&\, (SA_{i,j} > S_c) \,\&\, (SA_{i+1,j+1} > S_b) \,\&\, (SA_{i+2,j+2} > S_b)$

(d) $\left\{ i,j \;\middle|\; \max_{i=1}^{Z_{II}} \left(\max_{j=1}^{Z_{MM}} (SA_{i-2,j-2} + SA_{i-1,j-1} + SA_{i,j} + SA_{i+1,j+1} + SA_{i+2,j+2}) \right) \right\}$

where, S_b and S_c are thresholds. Z_{II}, Z_{MM} are the total number of concavities and convexities of the input data and the model, respectively. Conditions (c) and (d) are introduced because many complicated Thai characters are composed of several similar arcs.

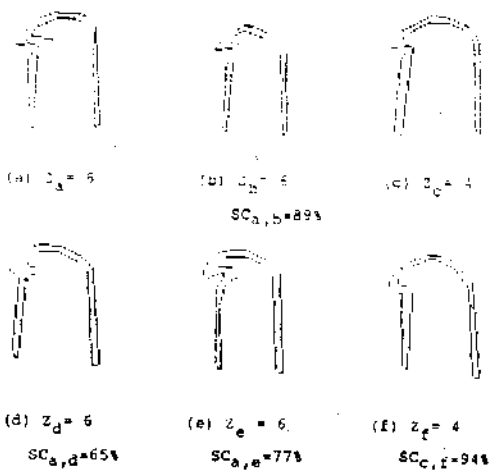


Fig. 9 An example of grouping of characters. Z_a, Z_b, Z_c, Z_d, Z_e and Z_f is the number of concavities and convexities of characters shown in Fig. (a), (b), (c), (d), (e) and (f), respectively. (SC: the similarity between characters.)

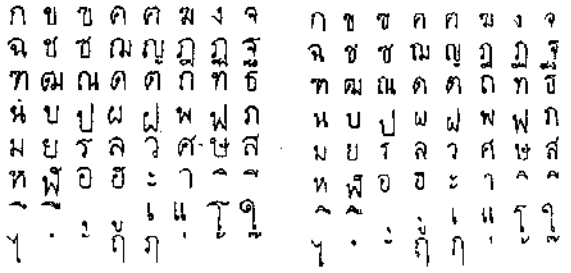


Fig. 10 Examples of data.

The most similar arc pair is utilized as standard for determining the matching of the next arc pairs. Finally, the similarity between characters $SC_{II,MM}$ is calculated as follows.

$$SC_{II,MM} = \sum_{i,j} (SA_{i,j}) \times 100 / (2 \times Z_{II}) \ \% \qquad (3)$$

The model MM whose $SC_{II,MM}$ is maximum value and larger than a threshold is selected as the identified character with the input one. Else, the input data is rejected.

6. Experimental Results

According to the algorithm described above, the recognition experiments by computer were made. Sixty nine of Thai characters, two of each from 6 authors, in total 828 data were used for the experiments. Examples of data are shown in Fig. 10. The data were input to a mini computer (OKITAC-4300, 40 kW, 16 bits/W), as a binary data of 128 × 128 dots. The following experiments were made. (1) A data set obtained from the first writing of each author was utilized as learning data (414 data) and the rest as test data (414 data). From the learning data, a dictionary of

models was made according to the algorithm described in Section 4.2. A dictionary of characters having 398 elements (characters composed of only convexities were not included) and a dictionary of arcs having 228 elements were obtained. As for recognition results, a recognition rate of 99.5% and an error rate of 0.5% for the same learning set, and the recognition rate of 87.2% and an error rate of 8.5% for the test set were observed.

(2.1) The learning set and the test set were the same as those of the experiment (1). A dictionary of models was made according to the algorithm described in Sections 4.2 and 4.3 using the learning set. A dictionary of characters having 224 elements and a dictionary of arcs having 220 elements were obtained. The method was tested on the learning set, and a recognition rate of 95.2% was observed.

(2.2) Only the ill-identified and rejected learning characters in experiment (2.1) were fedback for making models. After a dictionary of characters having 241 elements and a dictionary of arcs having 245 elements were obtained and a recognition rate of 99.3% and an error rate of 0.7% were observed for the learning set, an experiment was made for the test set used in experiment (1). A recognition rate of 88.9%, an error rate of 7.0% and a rejection rate of 4.1% were observed. In Fig. 11, examples of rejected and ill-identified characters are shown. The reason of the rejection and error is mainly due to the distortion of some complicated characters, and the difference in numbers of concavities and convexities between data and models.

According to the experimental results, the size of the dictionary of characters of the experiment (2.2) is reduced to 60.6% of that of the experiment (1). However, the size of the dictionary of arcs is increased from 228 to 245 elements. In the experiment (1), the classification for several similar characters was not good. Therefore, the detection of similar arc-pairs in model making was done strictly for the learning characters being different in lengths as well as the rejected and ill-identified characters in experiment (2.1), and that made the increment of the size of the dictionary of arcs in the experiment (2.2).

7. Conclusions

In this paper, the recognition method of handprinted Thai characters based on concavities and convexities of contours is described. In the method, Freeman chain code and directional differences of contour tracing are utilized to extract concavities and convexities of character contours. Several local features of arc are extracted and used to calculate similarities between arc portions and similarity between characters.

A dictionary of characters and that of arcs are made to obtain a compact and effective dictionary of models. In model making, it is possible for models of different categories to use common feature values of the arc dictionary.

The proposed method has tolerance to size variation and rotation without any implement of size and skew normalization. The effective combination of feature extraction, matching process and model making is made for the recognition of handprinted Thai characters being composed of many complicated and similar characters. Style variation caused by writers is absorbed by increasing number of arcs of the arc dictionary. Rotation problem is reduced by making use of A_{xyz} to transform arc vertices from x-y plane to u-v plane. Further, size variation problem is solved in the process of calculating the similarity between arcs. The method was applied to a variety of handprinted Thai characters, and 99.3% of recognition rate for learning data, and 88.9% of recognition rate for test data have been achieved.

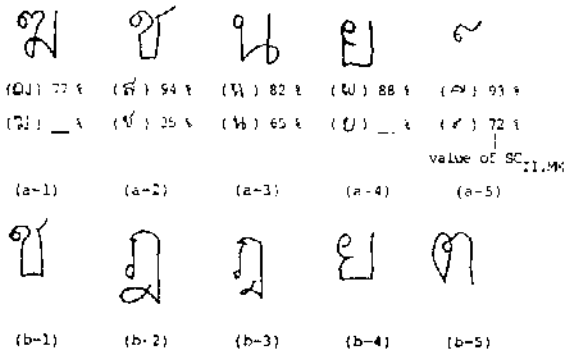


Fig. 11 (a) Examples of ill-identificaed characters.
(b) Examples of rejected characters.

References

(1) Agui, T., Nakajima, M., Kim, T.K. and Takahashi, E.T.: "A method of recognition and representation of Korean characters by tree grammars", IEEE Trans. PAMI-1, 3, pp. 245–251 (1976).

(2) Badie, K. and Shimura, M.: "Machine recognition of Arabic handprinted scripts", Trans. IECE Japan (Section E), E65, 2, pp. 107–114 (Feb. 1982).

(3) Chinnuswamy, P. and Krishnamoorthy, S.G.: "Recognition of handprinted Tamil characters", Pattern Recognition, 12, 3, pp. 141–152 (1980).

(4) Sethi, I.K. and Chatterjee, B.: "Machine recognition of constrained handprinted Devanagari", Pattern Recognition, 9, 4, pp. 69–75 (1977).

(5) Suen, C.Y., Berthod, M. and Mori, S.: "Automatic recognition of handprinted characters – The state of art", Proc. IEEE, 68, 4, pp. 469–487 (1980).

(6) Hiranvanichakorn, P., Agui, T. and Nakajima, M.: "A recognition method of Thai characters", Trans. IECE Japan (Section E), E65, 12, pp. 737–744 (Dec. 1982).

(7) Hiranvanichakorn, P., Agui, T. and Nakajima, M.: "Recognition of printed Thai characters by local information", Paper of Technical Group, PGIE83-56, IECE Japan (1983).

(8) Mckee, J.W. and Aggarwal, J.K.: "Computer recognition of partial views of curved objects", IEEE Trans. on Comput., C-26, 8, pp. 790–800 (Aug. 1977).

(9) Pavlidis, T. and Ali, F.: "A hierarchical syntactic shape analyzer", IEEE Trans. PAMI-1, 1, pp. 2–9 (1979).

(10) Pavlidis, T.: "The use of a syntactic shape analyzer for contour matching", IEEE Trans. PAMI-1, 3, pp. 307–310 (1979).

(11) Yamamoto, K.: "Recognition of handprinted Hiragana characters by concave and convex features and automatically merging the dictionary", Trans. IECE Japan (Section J), J65-D, 6, pp. 774–

781 (June 1982).
(12) Pavlidis, T. and Ali, F.: "Computer recognition of handwritten numerals by polygonal approximations", IEEE Trans. Syst., Man & Cybern., SMC-5, 3, pp. 610-615 (1975).

Appendix

A similarity between arcs i and j is calculated as follows.

Let LU_{kui} and LV_{kvi} ($kui=1\sim NU_i$; $kvi=1\sim NV_i$, where, NU_i and NV_i are numbers of segmented portions) be lengths of segmented portions of arc i along the u and v axis, respectively. And let LU_{kuj} and LV_{kvj} ($kuj=1\sim NU_j$; $kvj=1\sim NV_j$; where, NU_j and NV_j are numbers of segmented portions) be lengths of segmented portions of arc j along the u and v axis, respectively.

The following scores T_{ku} and link levels R_{ku} are given to the i,j arcs according to the following condition (a) or (b).

(a) $T_{ku}=1$, $R_{ku}=2$

for $|\ LU_{kui}\times B_{ji}-LU_{kuj}\times B_{ij}\ |<=\varepsilon_1$

(b) $T_{ku}=1$, $R_{ku}=1$

for $\varepsilon_1<|\ LU_{kui}\times B_{ji}-LU_{kuj}\times B_{ij}|<=\varepsilon_2$

where, kui and kuj vary from 1 to NU_{ij} (NU_{ij} is the value of the larger of NU_i and NU_j). If the length of arc i is larger than that of arc j, B_{ji} is equal to 1 and B_{ij} is equal to the ratio of length of arc i to that of arc j, else, B_{ji} is equal to the ratio of length of arc j to that of arc i, B_{ij} is equal to 1. ε_1 and ε_2 are thresholds.

Similarly, T_{kv} and R_{kv} are given to the i,j arcs according to the condition (c) or (d).

(c) $T_{kv}=1$, $R_{kv}=2$

for $|\ LV_{kvi}\times B_{ji}-LV_{kvj}\times B_{ij}\ |<=\varepsilon_1$

(d) $T_{kv}=1$, $R_{kv}=1$

for $\varepsilon_1<|\ LV_{kvi}\times B_{ji}-LV_{kvj}\times B_{ij}|<=\varepsilon_2$

where, kvi and kvj vary from 1 to NV_{ij} (NV_{ij} is the value of the larger of NV_i and NV_j).

Then, a similarity between arcs $SA_{i,j}$ is determined as follows.

$$SA_{i,j}=\Big(\sum_{ku=1}^{NU_{ij}}(T_{ku}+R_{ku}/2)+\sum_{kv=1}^{NV_{ij}}(T_{kv}+R_{kv}/2)\Big)/(NU_{ij}+NV_{ij})$$

$SA_{i,j}$ takes a value in the range from 0.0 to 2.0, however, when $SA_{i,j}$ has value smaller than a threshold, zero is given to it.

PAPER

An On-line Recognition Method of Thai Characters

Pipat HIRANVANICHAKORN†, Takeshi AGUI† *and* Masayuki NAKAJIMA†, *Members*

SUMMARY This paper reports an on-line recognition method of Thai characters being composed of curves, and having many complicated and similar shapes. A character stroke is segmented into clockwise and counter clockwise arcs according as the stroke tracing is clockwise or counter clockwise, by making use of eight directional codes and directional differences of stroke tracing. Intuitively described features such as the sequence of stroke arcs, types of arc and relative positions of arcs are utilized for classifying characters. A multi-step classification method is introduced to achieve a good recognition rate. By applying the method to 69 categories (414 data) of Thai characters, a recognition rate of 100% for learning data, and a recognition rate of 96.4% for test data have been obtained.

1. Introduction

Studies of on-line character recognition have been made for two decades, because they provide possible commercial applications such as computer aided design, on-line data input, learning by computers, etc [1]. Recently, due to the low cost and high performance of the personal computer as well as the digitizing tablet, the adoption of online character recognition in business use such as in OA (office automation) can be realized. Many researches on this subject have been reported [2]–[9]. Thai characters have many complicated and similar shapes and they are mainly composed of curves. Therefore, a few works based on concavities and convexities of character contours have been done on the recognition of Thai characters [10]–[13] in optical character recognition (OCR). The study of Thai characters in the on-line system is not almost made.

In the on-line system, characters are recognized as they are drawn. Character features such as the number of character strokes and the stroke sequence which are not available in OCR, are easily implemented and are usable to reduce the complexity of the recognition techniques [8],[9].

Thai characters have many similar shapes, and most characters are single-stroke characters having complex and curve-like structures. Therefore, the shapes of character strokes are considered holding significant information for the recognition. In this paper, a structural analysis method of character strokes is proposed for the on-line recognition of Thai characters.

As for on-line character recognition, many methods based on the structural information of character strokes have been proposed [4]–[9]. Especially, methods [4],[5] are considered to be effective for such characters as alphanumerics being composed of curves. In the methods, a character stroke is segmented, and the number, types of segments such as straight lines and curves are utilized as features. As Thai characters have complicated and similar shapes, and small intrusions and protrusions give important meanings to the characters, an effective method of segmenting character strokes and extracting features is required.

This paper proposes a simple and effective online recognition method of Thai characters. In this method, a character is input to a computer through a digitizing tablet in the form of sequences of point coordinates representing character strokes. Stroke points of each character stroke are coded by eight directional codes. The points which indicate directional changes of the codes are detected. The points are then utilized to detect + and − vertices which indicate if the stroke tracing is counter clockwise or clockwise. The + and − vertices are used to segment a stroke into counter clockwise and clockwise arcs. A counter clockwise arc is the stroke portion passing successive + vertices, and the stroke portion passing successive − vertices is called a clockwise arc. Further, for possible business use (such as in OA) which requires implementation of small computers, a simple and fast recognition algorithm is needed. Significant character information which is not available in OCR, is easily implemented in the on-line system. Further, the information is usable for reducing the complexity of the algorithm as well as the recognition time. In this paper, intuitively described features such as the sequence of stroke arcs, types of arcs, relative positions of arcs, etc., are extracted and utilized effectively for classifying characters. A multi-step classification method is introduced to achieve a good recognition rate.

2. Thai Characters

Thai characters are composed of forty four consonant characters, thirty two vowel ones, four tonal symbols and four special characters. Several vowel characters can be decomposed into fundamental ones [1]. As the result, sixty nine fundamental Thai characters shown in Fig. 1 are obtained. On the view point of on-line character recognition, most Thai characters are single-stroke characters having complex structures. Few characters shown in Fig. 1(b) are separated into several strokes. Many characters have small holes, and the drawing of the characters begins by tracing the portions having holes. Further, several characters such a

Manuscript received November 2, 1984.
Manuscript revised April 5, 1985.

†The authors are with the Imaging Science and Engineering Laboratory, Tokyo Institute of Technology, Yokohama-shi, 227 Japan.

Fig. 1 (a) A set of typical Thai characters.
(b) Characters composed of several strokes.
(c) Similar characters.

ones shown in Fig. 1(c) have similar shapes.

3. Feature Extraction

3.1 Segmentation of Character Stroke

Most Thai characters are single-stroke characters having similar and complex shapes. Therefore, the idea of segmenting a character stroke into several arcs and using features of arcs for the classification is considered to be effective for the recognition of the characters.

As for character segmentation, the polygonal approximation approach such as the split- and- merge one has been proposed [14-16]. The major disadvantage of the method is that the program is complex and the process is time-consuming. For avoiding the problem, methods in which concavities and convexities of characters are extracted by using directional differences of contour tracing of characters, have been proposed [12-13]. Especially, methods [11-13] have been applied effectively to the recognition of Thai characters in OCR. In on-line character recognition, methods in which directional codes are utilized to segment a character stroke into straight line segments, have been proposed [6]-[8]. As Thai characters have complex and curve-like structures, an effective segmentation method is needed.

In this paper, the extraction method of convexity and concavity in Ref. (13) is modified and utilized to segment character strokes according to clockwise or counter clockwise tracing.

In the on-line system, the segmentation of character strokes is somewhat difficult owing to change in speed of writing of a user and noises introduced by pen-up, pen-down operations and the digitization. As small intrusions and protrusions give important meanings to Thai characters, the segmentation of character strokes is difficult.

In this paper, stroke points of each character stroke are coded by eight directional codes (as shown in Fig. 2). The points which indicate directional changes of the codes are detected. The detected points are then coded according to directional differences. The coded points are utilized to detect + and − vertices which indicate if the stroke tracing is counter clockwise or clockwise.

A character is input to a computer through a digitizing tablet in the form of sequences of x-y point coordinates representing character strokes. Each sequence of point coordinates obtained while the pen is pressing the tablet surface, represents each character stroke as follows.

$$P_n = (P_{x_n},\ P_{y_n}) : n = 1 \sim N \qquad (1)$$

where, P_n is the n-th point of a stroke in an $x-y$ plane having the original coordinate at the upper-and-left most of the plane. N is the number of the points.

Each point is coded by eight directional codes shown in Fig. 2, and let $F_n \in \{ 0 \sim 7 \}$ be an element of a set of codes for P_n. Then, the points which indicate directional changes of the codes are detected. However, a stroke contains noisy data, especially, at the end portions of the stroke. The two end-points of a stroke and points indicating directional changes of the codes are detected by a procedure given in Appendix.

Let $C_i = (C_{x_i},\ C_{y_i}) : i = 1 \sim I$ be a sequence of a stroke's points indicating directional changes of eight directional codes and including the end points of the stroke. Further, let $F_i \in \{ 0 \sim 7 \}$ be the code value of C_i.

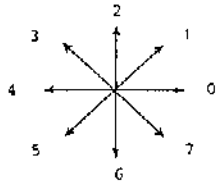


Fig. 2 Eight directional codes.

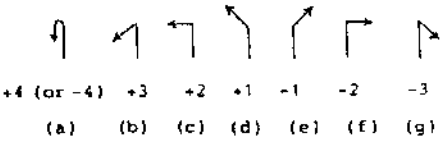


Fig. 3 Sign assignment according to directional differences.

Generally, clockwise and counter clockwise portions of a stroke can be extracted by labeling a point as − or + at which the stroke tracing turns to the right or left, respectively. Therefore, a sign S_i is assigned to each point C_i according to directional differences of the codes between C_j and C_{j-1} as illustrated in Fig. 3. In the case of Fig. 3(a), S_i is given a sign +4 when the directional differences of the codes between C_{j+1} and C_{j-1} has a sign +, else S_i is given a sign −4. Further, S_i is represented as follows.

$$S_i = 0 \qquad \text{for } i = 1 \text{ and } i = I$$

$$S_i \in \{+1, +2, +3, +4, -1, -2, -3, -4\} \text{ for } i = 2 \sim I-1.$$

Accordingly, each C_i is given a value of the eight directional codes F_i and a sign S_i. Examples of points having signs are shown in Fig. 4(a). The points are then utilized to detect + and − vertices of a stroke as follows.

[Definition 1] When a point C_i satisfies either condition (a) or (b), a label + is assigned to C_i

(a) $(S_i > 0)$ & $(S_{i-1} > 0$ or $S_{i+1} > 0)$

(b) $(S_i > 0)$ & $(S_{i-1} \leqq 0)$ & $(S_{i+1} \leqq 0)$ & $(F_i \neq FF)$ & $(D_{i,i+1} \geqq k_1)$ & $(D_{i-1,i} \geqq k_2)$

where, $D_{i,j} = \sqrt{(C_{x_i} - C_{x_j})^2 + (C_{y_i} - C_{y_j})^2}$

and k_1 and k_2 are thresholds. FF is a digital direction (i.e., the value of the eight directional codes) of the stroke portion before coming to C_i. Further, when C_i is selected as a + vertex and $D_{i,i+1}$ is larger than a threshold, FF is newly set by the value of F_i. The code value of the start point of a stroke is initially given to FF.

[Definition 2] When a point C_i satisfies either condition (a) or (b), a label − is assigned to C_i

(a) $(S_i < 0)$ & $(S_{i-1} < 0$ or $S_{i+1} < 0)$

(b) $(S_i < 0)$ & $(S_{i-1} \geqq 0)$ & $(S_{i+1} \geqq 0)$ & $(F_i \neq FF)$ & $(D_{i,i+1} \geqq k_1)$ & $(D_{i-1,i} \geqq k_2)$.

An example of assignment for a Thai character with + and − according to the definitions is shown in Fig. 4(b).

Thus labeled vertices are used to extract clockwise and counter clockwise arcs. Further, in order to achieve insensitive locations of the start and end vertices of arcs, the following condition is introduced. When the distance between adjoining + and − vertices is short, the vertex or both vertices which have the sign S_i being equal to +1 (or −1) are deleted.

For avoiding a disadvantage which occurs when only two − (or +) vertices appear between + (or −) vertices, the portion connecting the two vertices is regarded as an arc, a clockwise arc and a counter clockwise arc are defined as follows.

[Definition 3] When there are successive + vertices between two points PS and PE, the successive + vertices including PS and PE are called a counter clockwise arc; where, PS and PE have following four cases, i.e.,

1) PS and PE are both − vertices
2) PS is a − vertex and PE is the end point of stroke
3) PS is the start point of stroke and PE is a − vertex
4) PS and PE are both the end points of stroke.

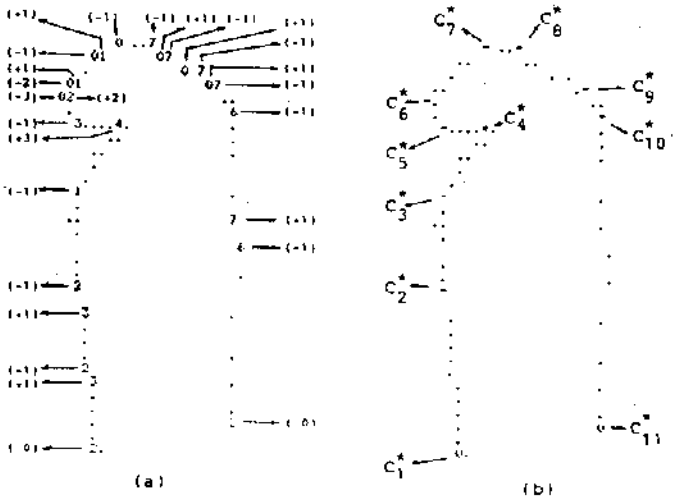


Fig. 4 (a) Examples of assignment of signs S_i (sign in ( )) to stroke points indicating directional changes. The numeral 0, 1, 2, 3, 4, 5, 6 or 7 is the value of eight directional codes F_i given to each point.
(b) An example of assignment a character with + and −. C_1^* is the stroke start-point and C_{11}^* is the stroke end-point.

Further, PS and PE are points being adjacent to the successive + vertices, PS is the point found before the successive + vertices and PE is the one found after.

[Definition 4]' When there are successive − vertices between two points MS and ME, the successive − vertices including MS and ME are called a clockwise arc;
where, MS and ME have following four cases, i.e.,
1) MS and ME are both + vertices
2) MS is a + vertex and ME is the end point of stroke
3) MS is the start point of stroke and ME is a + vertex
4) MS and ME are both the end points of stroke.
Further, MS and ME are points being adjacent to the successive − vertices, MS is the point found before the successive − vertices and ME is the one found after.

Accordingly, adjoining counter clockwise and clockwise arcs have a common portion. Further, the start and end vertices of a counter clockwise arc are PS and PE, respectively, and those of a clockwise arc are MS and ME, respectively. As shown in Fig. 4(b), the segment passing points C_3^*, C_4^* and C_5^* is regarded as a counter clockwise arc having C_3^* as PS and C_5^* as PE. Further, the segment passing points C_1^*, C_2^*, C_3^* and C_4^* is a clockwise arc having C_1^* as MS and C_4^* as ME, and that passing points C_6^*, C_7^*, C_8^*, C_7^*, C_8^*, C_9^*, C_{10}^* and C_{11}^* is a clockwise arc having C_6^* as MS and C_{11}^* as ME.

Further, a line stroke is defined as a stroke which has neither + nor − vertices.

3.2 Line Approximation

Some Thai characters have low curvature portions which give different numbers of arcs to the characters of the same category. For avoiding the problem, a line approximation is introduced as follows. When the ratio of the length of an arc to the distance between the start and end points of the arc is smaller than a threshold, the arc is approximated by a straight line segment.

3.3 Features of Characters

In on-line character recognition, the number of character strokes, the stroke sequence and relative positions of strokes are effectively used as features[8],[9]. As Thai characters are written in one or few strokes, the problem of the instability of the features is not so considerable as in the recognition of Chinese characters.

In the papers[01−03], several features of concave and convex arcs are utilized for calculating similarities between arcs and between characters. The methods are applied effectively to the recognition of Thai characters in OCR. However, the methods require great computation time for recognizing numerous characters having complex structures.

In the paper[03], distinctive character features such as relative positions and types of concave arcs are utilized effectively for the recognition of numerals in OCR. In the paper[10], an on-line recognition method of alphabet has been proposed, in which character height is divided into zones, and the zones are utilized to extract character features. Further, in the on-line system, information such as the sequence of segmented portions of stroke, locations of the start and end points of stroke is easily implemented[8],[9]. Further, such information is utilizable for reducing the recognition time. In this paper, intuitively described features such as the sequence of stroke arcs, relative positions of arcs, loops, etc., are extracted and utilized for the effective recognition of Thai characters.

Let $V_j = (V_{xj}, V_{yj})$; $j = 1 \sim J$ be a sequence of + and − vertices belonging to an arc in the x-y plane having the original coordinate at the upper-and-left most of the plane. Define;

$$dY_{j,j+1} = V_{y,j+1} - V_{yj} . \tag{2}$$

The value of $dY_{y,j+1}$ is calculated at every vertex ($j = 1 \sim J-1$). The vertices at which the sign of $dY_{j,j+1}$ changes from negative to positive are called top points (TP) of arc, and bottom points (BP) of arc are vertices at which the sign changes from positive to negative. Further, both TP and BP are called bending points of arc.

As shown in Fig. 5, the distance between the uppermost and lowermost points of each character stroke is divided along y-direction into three zones, i.e., zone A, zone B and zone C. One of the reasons is that most Thai characters are written from the portions existing at the uppermost or lowermost parts of the characters, and few characters are written from the middle parts. Further, the distance along y-direction from the bottom of zone C to two third of the zone is called zone D.

The following list summarizes the features utilized to describe Thai characters in the on-line system.

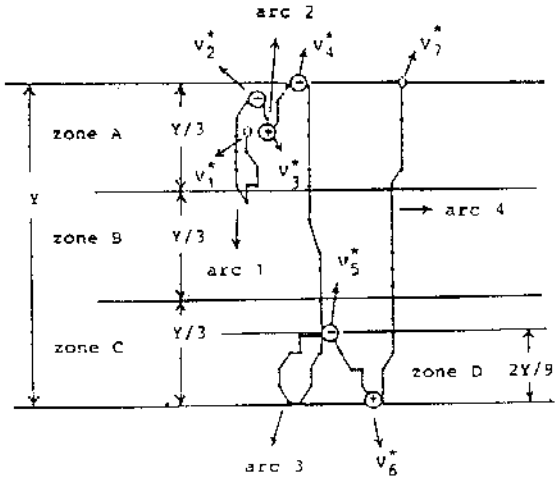


Fig. 5 An example of dividing distance along y-direction of a character stroke into zones.
arc 1: the portion passing points V_1^*, V_2^* and V_3^*.
arc 2: the portion passing points V_3^*, V_4^* and V_5^*.
arc 3: the portion passing points V_5^*, V_6^*, V_1^* and V_2^*.
arc 4: the portion passing points V_5^*, V_6^* and V_7^*.
V_1^* and V_7^* are the end points of the stroke.

(1) NS number of strokes.

(2) $q_{ns,1}$ number of stroke arcs and ns varies from 1 to NS.

(3) $q_{ns,2}$ which takes sign + when the first arc of a stroke is counter clockwise and takes sign − for a clockwise arc.

(4) $q_{ns,3}$ Which takes value 1 when a loop exists at the start portion of a stroke and takes value 0 when not exist. The portion from the start point of a stroke to the k-th stroke point is called the start portion of the stroke, and k is a threshold.

(5) $q_{ns,4}$ location of the start point of a stroke. It is noted that when a loop exists at the start portion of the stroke (i.e., $q_{ns,3}=1$), $q_{ns,4}$ indicates the location of the loop. The location of a loop is represented by the average of its uppermost and lowermost y coordinates.
$q_{ns,4}=1$, if the location is in zone A.
$q_{ns,4}=2$, if the location is in zone B.
$q_{ns,4}=3$, if the location is in zone C.

(6) $q_{ns,5}$ which takes value 1 when another loop which is not the one described in (4) exists in a stroke and takes value 0 when not exist. Value 1 is given to $q_{ns,5}$ either when $q_{ns,3}$ has value 1 and another loop is found or when $q_{ns,3}$ has value 0 and a loop is found.

(7) $q_{ns,6}$ location of the loop described in (6) and $q_{ns,6}$ has value 1, 2 or 3 according to its location in zone A, B or C.

(8) $q_{ns,7}$ the arc number at which the loop described in (6) exists.

(9) $q_{ns,8}$ which takes value 1 if the first arc has a portion existing in zone D and takes value 0 else.

(10) $q_{ns,9}$ which takes value 1 if the last bending point of the last arc is TP (i.e., top point), and takes value 0 else.

(11) $q_{ns,10}$ which takes value 1 if the last bending point of the first arc is TP and takes value 0 else.

(12) $q_{ns,11}$ which takes value 1 if the difference of x coordinates between the start vertex and the end vertex of the last arc has sign +, and takes value 0 else.

(13) $q_{ns,12}$ which takes value 1 if $q_{ns,9}$ has value 1 and the location of the end vertex of the last arc is in zone C, and takes value 0 else.

(14) $q_{ns,13}$ which takes value 1 if $q_{ns,10}$ has value 1 and the location of the end vertex of the first arc is in zone C, and takes value 0 else.

(15) $q_{ns,14}$ length of stroke.

(16) $q_{ns,15}$ ratio of the stroke length from the last bending point of the first arc to the stroke end-point to the distance of the points. It is noted that $q_{ns,15}$ is calculated only when $q_{ns,1}$ has value more than one, $q_{ns,9}$ has value 0 and $q_{ns,10}$ has value 1.

Table 1 The use of intuitively described features for classifying similar characters.

	NS	$q_{1,1}$	$q_{1,2}$	$q_{1,3}$	$q_{1,4}$	$q_{1,5}$	$q_{1,6}$	$q_{1,7}$	$q_{1,8}$	$q_{1,9}$
ก	2	2	-	1	3	0	0	0	1	1
ถ	1	2	-	1	3	0	0	0	1	1
ภ	1	2	-	0	3	0	0	0	1	1
ฦ	1	3	-	1	3	0	0	0	1	1
ฝ	1	3	+	1	2	0	0	0	1	0
ฬ	1	3	+	1	1	0	0	0	1	0
ฮ	1	3	+	1	1	0	0	0	0	0
ป	1	2	-	1	1	1	3	1	1	0
ฦ	1	4	-	1	1	1	3	3	0	0
ฦ	1	4	-	1	2	1	3	3	1	0
ฬ	1	4	-	1	2	0	0	0	1	0
ฬ	1	4	-	1	2	1	1	4	1	0

For the character shown in Fig. 5, the values NS = 1, $q_{1,1}=4$, $q_{1,2}=-$, $q_{1,3}=1$, $q_{1,4}=1$, $q_{1,5}=1$, $q_{1,6}=3$, $q_{1,7}=3$, $q_{1,8}=0$, $q_{1,9}=0$, $q_{1,10}=1$, $q_{1,11}=1$, $q_{1,12}=0$, $q_{1,13}=0$, $q_{1,14}=74$, $q_{1,15}=4.7$, are obtained. Further, Table 1 demonstrates the use of the extracted features for classifying several similar characters. As shown in Table 1, zone D is used effectively for classifying such similar characters as ปี and ฝ, ฦ and ฦ. Further, the feature $q_{1,10}$ is used for classifying the characters ก and ภ, $q_{1,11}$ for classifying ฦ and ฮ, $q_{1,13}$ for classifying ฦ and ฬ, and $q_{1,15}$ for classifying ป and ฝ.

4. Classification and Dictionary Generation

4.1 Classification Algorithm

For the efficient use of the extracted features as well as gain of the recognition results, the classification method shown in Fig. 6 is introduced, in which the matching of data and models is made at several steps. In each step, the matching of the determined features is made between the input character and models. When only one model matches with the input character, the matching of the all features is made between the two characters. If the number of the model features which match those of the data is larger than a threshold, the model is selected as the identified character with the input one. Otherwise, the input character is rejected.

4.2 Dictionary

A dictionary is made by making use of learning characters. Each character of a set of typical Thai characters is written, and its feature values are put in the dictionary as feature values of each of character models. Further, as

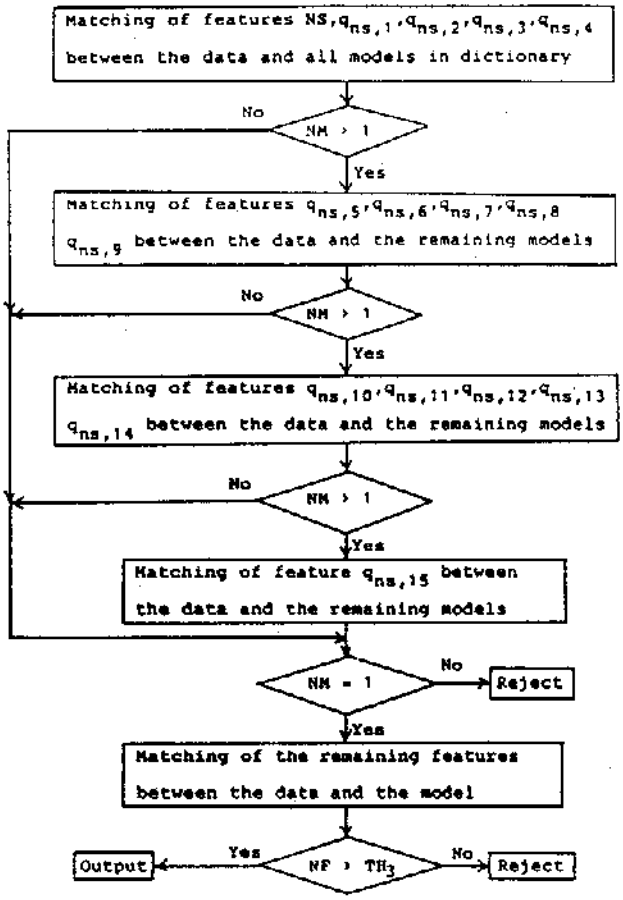


Fig. 6 A flowchart of classification.

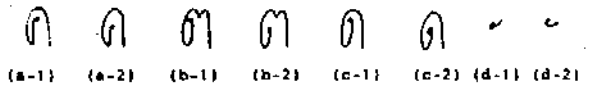


Fig. 7 Examples of characters whose locations of stroke start-points are not stable.

shown in Fig. 7, the locations of the start points of several Thai character strokes are not stable. Therefore, prior information of the start-point locations of several character strokes is given to the models of the characters. Then, feature values of each learning data are compared with those of the models according to the algorithm described in Section 4.1. If a data is rejected, its feature values are placed in the dictionary as feature values of another new model.

5. Experimental Results

According to the algorithm described above, the recognition experiments by a computer were made. Sixty-nine of Thai characters, three of each from 2 authors, in total

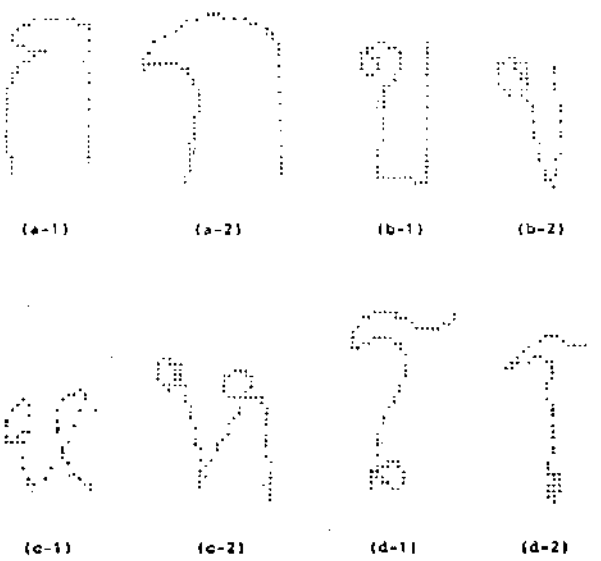


Fig. 8 Examples of characters belonging to the same categories but having different shapes.

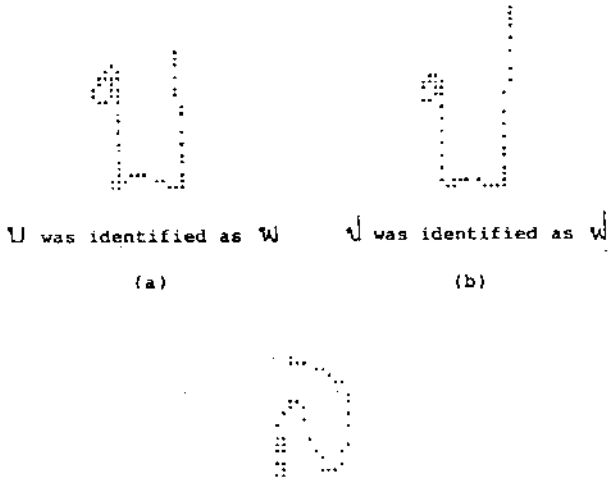


Fig. 9 The ill-identified characters.

414 data were used for the experiments. The data were input to a mini computer (OKITAC-4300b, 40 kW, 16 bits/W) through a digitizing tablet, as data sizes of 64×64 dots.

A data set obtained from the first writing of each author was utilized as learning data (138 data) and the rest as test data (276 data). Feature values of each of the learning characters are compared with those of models in the dictionary. Only the rejected characters are fedback for generating models. Fig. 8 shows examples of characters which belongs to the same categories but have different shapes and are utilized for making models. After a recognition rate of 100% was observed for the learning set, an experiment

was made for the test set. A recognition rate of 96.4%, and error rate of 1.1% and a rejection rate of 2.5% were observed. In Fig. 9, the ill-identified characters are shown. The characters shown in Fig. 9(a) and (b) were written in such way that the values of $g_{1,1}$ of them (i.e., $g_{1,1}=4$) differ from those of their models (i.e., $g_{1,1}=2$). However, the problem of this writing style can be solved by introducing complementary information such as the position of the third arc. As for the character shown in Fig. 9(c), the value of the feature $g_{1,9}$ (i.e., $g_{1,9}=0$) differs from that of its model (i.e., $g_{1,9}=1$). However, the characters  and  can be classified by the feature $g_{1,10}$ (i.e., $g_{1,10}=1$ for the character  and $g_{1,10}=0$ for ). Therefore, the ill-identified character may be fedback to make a model for the characters having this writing style. The reason of the rejection is mainly due to the differences of the locations of the start points of strokes between the characters and their models. Further, failure in the detection of loops of several characters causes the rejection of the characters.

7. Conclusions

In this paper, an on-line recognition method of handprinted Thai characters based on the structural analysis of character strokes is described. An effective combination system of feature extraction, classification process and model making is introduced for the recognition of Thai characters being composed of many complicated and similar characters. In the method, stroke points which indicate directional changes of eight directional codes are given signs + and − according to directional differences. A character stroke is then segmented into clockwise and counter clockwise arcs by making use of the points given signs. Intuitively described features of arcs such as the sequence of arcs, types of arcs, locations of arcs, etc., are extracted and used effectively for classifying characters. It is noted that the characteristics of on-line character recognition allow us to use such intuitively described features effectively. In order to achieve a good recognition rate, the classification process is divided into several steps. The recognition method is applied to a variety of Thai characters, and 100% of recognition rate for learning data, and 96.4% of recognition rate for test data have been achieved.

References

(1) C.Y. Suen, M. Berthod and S. Mori: "Automatic recognition of handprinted characters — The state of art", Proc. IEEE, 68, 4, pp. 469—487 (1980).

(2) T.K. Kim, T. Agui and M. Nakajima: "On-line recognition of Korean characters", Paper of Technical Group, TGPRL84-18, IECE Japan (1984).

(3) M. Yuruğ, S. Nagata, K. Onuma and K. Kubota: "On-line character recognition by Hierarchical analysis method", Paper of Technical Group, TGPRL84-17, IECE Japan (1984).

(4) W.W. Loy and I.D. Landau: "An on-line procedure for recognition of handprinted alphanumeric characters", IEEE Trans. Pattern Anal. & Mach. Intell., PAMI-4, 4, pp. 422—427 (1982).

(5) A. Belaid and J.P. Haton: "A syntactic approach for handwritten mathematical formular recognition", IEEE Trans. Pattern Anal. & Mach. Intell., PAMI-6, 1, pp. 105—111 (1984).

(6) T. Tomimoto, K. Shima and H. Ota: "On-line recognition of handprinted Japanese characters", Paper of Technical Group, TGPRL83-37, IECE Japan (1982).

(7) K. Yoshida and H. Sakoe: "Online character recognition by stack DP matching method", Paper of Technical Group, TGPRL83-29, IECE Japan (1983).

(8) H. Terai and K. Nakata: "On-line, real-time recognition of handwriting Chinese characters and Japanese Katakana sylabary", Trans IECE Japan (Section J), J56-D, 5, pp. 312—319 (May 1973).

(9) K. Odaka, H. Arakawa and I. Masuda: "Online recognition of handwritten characters by approximating each stroke with several points", Trans. IECE Japan (Section J), J63-D, 2, pp. 153—160 (Feb. 1980).

(10) K. Yamamoto: "Recognition of handprinted Hiragana characters by concave and convex features and automatically merging the dictionary", Trans. IECE Japan (Section J), J65-D, 6, pp. 774—781 (June 1982).

(11) P. Hiranvanichakorn, T. Agui and M. Nakajima: "A recognition method of Thai characters", Trans. IECE Japan, E65, 12, pp. 737—744 (Dec. 1982).

(12) P. Hiranvanichakorn, T. Agui and M. Nakajima: "Recognition of printed Thai characters by local information", Paper of Technical Group, TGIE83-56, IECE Japan (1981).

(13) P. Hiranvanichakorn, T. Agui and M. Nakajima: "Recognition method of Thai characters by local features", Trans. IECE Japan (Section E), E67, 6, pp. 425—432 (Aug. 1984).

(14) T. Pavlidis and F. Ali: "Computer recognition of handwritten numerals by polygonal approximations", IEEE Trans. Syst., Man & Cybern., SMC-5, 3, pp. 610—615 (1975).

(15) T. Pavlidis and F. Ali: "A hierarchical syntactic shape analyzer", IEEE Trans. Pattern Anal. & Mach. Intell., PAMI-1, 1, pp. 2—9 (1979).

(16) T. Pavlidis: "The use of a syntactic shape analyzer for contour matching", IEEE Trans. Pattern Anal. & Mach. Intell., PAMI-1, 3, pp. 307—310 (1979).

(17) T. Sakai and M. Nagao: "Recognition machines of characters and shapes", pp. 63—68, KYOURITSUSHUPPAN (1967).

Appendix

The two end points of a stroke and stroke points which indicate directional changes of the eight directional codes are detected by this procedure (see Section 3.1).

Let $F_n \in \{0 \sim 7\}$ be an element of a set of the eight directional codes of a stroke point P_n, $n = 1 \sim N$ (where, N is the number of points constituting the stroke).

(a) Find the first P_m (where, $m \in n$) whose the code value F_m differs from that of P_{m+1} less than two and regard the point P_m as the start point of the stroke.

(b) The following procedure is made from the start point P_m of the stroke to P_N. When there is an indication of changing of the eight directional codes at a point P_{m_1}, $m < m_1 < N$ (i.e., $F_{m_1} \neq FD$, where FD is the direction of the stroke portion before coming to P_{m_1}), the changing is tested whether it is caused by a noise or not. If a very small intrusion or protrusion is found, the point P_{m_1} is ignored. Else, P_{m_1} is detected as a point indicating directional changes of the eight directional codes and the value of FD is newly set by F_{m_1}. Note that value of F_m is initially given to FD.

(c) Find the first P_{m_2}, $m < m_2 < N$, which is a point