On-line handwritten character string recognition

(オンライン手書き文字列認識技術に関する研究)

Masayoshi Okamoto

Jan. 1999

Gifu University

On-line handwritten character string recognition

オンライン手書き文字列認識技術に関する研究

要旨

本論文は、ペンで容易に文字入力するための基盤となるオンライン手書き文字認識技術と文字切り出し技術に関して述べたものである。人にやさしい操作手段の一つとしてペン入力が期待されている。１文字づつ丁寧に筆記された文字を認識する技術については、既に携帯型情報端末などへ応用されているが、更に筆記自由度の高い文字入力の実現が課題となっている。１文字毎の記入枠を無くし、文字列として連続して筆記入力できる環境を実現するために、筆記された文字列から文字を切り出す技術と、自由に筆記された低品質の文字を高精度に認識する技術が重要である。本論文では、切り出しに関する物理的特徴と文字認識結果や言語処理結果などの論理的特徴を適切に融合した文字切り出し技術の研究成果と、筆順、画数、字形変動に対応した文字認識手法として、ＯＣＲで代表的な方向性特徴（オフライン特徴）と独自の方向変化特徴（オンライン特徴）を用いたパターンマッチングによる新しい文字認識手法の研究成果について述べる。本論文は、６つの章から構成されており、以下にその概要を述べる。

第１章では、本研究を始めるに至った背景と研究目的および概要を述べている。

第２章では、文字ピッチなどの物理的特徴と文字認識結果や言語処理結果などの論理的特徴をネットワーク表現で融合させた高性能な文字切り出し手法を提案している。従来、文字切り出しにおいては、まず、物理的特徴によって切り出し位置の候補を抽出し、これらの切り出し位置候補間の筆記ストローク集合に対して文字認識して、文字認識類似度が高いかどうかの結果や、文字認識候補を組み合わせて、単語あるいは文章として成立するかどうか言語処理の結果に応じて切り出し位置を決定する方法がよく知られている。しかしながら、物理的特徴に基づく切り出しの確からしさの情報は使われず、文字認識結果と言語処理の結果だけによって最終的に切り出し位置が決定されている。このため、文字ピッチが小さくて本来文字間ではない個所でも、候補に含まれていれば誤って切り出されることもあり、文字認識と言語処理による悪影響

が少なくない。この手法は、必然的に文字認識や言語処理結果の論理的特徴に極めて重きが置かれた処理系になっていると言える。本提案手法は、物理的特徴による切り出しの確からしさと、文字認識、言語処理による確からしさとの総合的な判断によって切り出し位置を決定するものである。切り出し実験の結果、従来手法と比較して、86.10%から 90.72%に性能向上でき、有効性を確認している。また、切り出し候補を物理的特徴と論理的特徴によるネットワーク表現で表し、その最短経路探索により、効率的に切り出し処理を行えることを示している。より切り出し性能を高めるためには、特に文字認識性能を向上させることが重要である。

第３章では、オンライン文字認識技術における代表的な手法の概要と、本提案手法のアプローチについて述べている。オンライン文字認識では、今まで演算量とソフトウェアサイズの面から構造解析的な認識手法が主流であった。昨今のＣＰＵ能力向上、メモリの低価格化に伴い、今後は統計的な認識手法（パターンマッチング法）が有望であると考えている。何故なら、文字認識辞書の自動学習が容易であるからである。また、筆順自由に対応するがために単にＯＣＲで使われている特徴をオンライン認識に適用するのではなく、オンラインの有効な特徴（ストロークの情報）を積極的に用い、融合させることが肝要である。

第４章では、本提案の文字認識手法の基本的な理論とその有効性を示している。ＯＣＲで代表的な方向性特徴にオンライン特徴である独自の方向変化特徴を加えてパターンマッチングすることにより、字形の変動に強くなることを確認した。ここで、方向変化特徴とは、何処でどの方向に筆記方向が変化しているかを表す特徴であり、今まで統計的手法で用いて有効性を示した例はない。また、ペンがアップしている区間を仮想的な直線(仮想ストローク)で補ってから方向変化特徴を抽出することにより、続け字などの画数変動に強くなることを確認した。本来ペンアップの区間が短い個所ほど続けて筆記されるケースが多いが、自由に筆記された低品質な文字では、本来ペンアップ区間が長い個所でも続けて筆記されることも多く、実験の結果、方向変化特徴量を仮想ストロークの長さに依存させないことが適切であると判明した。公開筆記文字データベース（HANDS-kuchibue_d-96-02）を用いた認識実験の結果、方向性特徴だけを用いた手法と比較して、77.89%から 86.32%に認識率が向上し、有効性を示している。

第５章では、本提案手法の改良に関係する実験結果に基づき、今後の見通しについて述べている。第６章では、本論文の結論について述べている。

Contents

1. General Introduction **1**
1.1 Background and goal of study 2
1.2 Summary of thesis 3

2. On-line Character String Separation Method using Network Expression **7**
2.1 Introduction 8
2.2 Character separation features 9
2.3 String separation method using physical and logical features 9
2.3.1 Separation process with a traditional method 9
2.3.2 Separation process with Multi-level network expression 10
2.3.3 Separation process with Unified network expression 11
2.4 Multi-level network expressions 12
2.4.1 Network expression (A) 12
2.4.2 Network expression (B) 22
2.5 Unified network expressions 25
2.5.1 Combination of network 26
2.5.2 Separation process with Unified networks 26
2.5.3 Effects of Unified network 29
2.6 Experiment 29
2.6.1 Handwritten data sets 29
2.6.2 Experimental results 31
2.6.3 Estimation of experiment results 33
2.7 Conclusion 37

3. Approaches to On-line Character Recognition **39**
3.1 Introduction 40
3.2 Structure analysis approaches 40
3.3 Statistical approaches (pattern matching) 41
3.4 Our approach (conclusion) 42

4. On-line Character Recognition Method using both Directional Features and Direction-Change Features 43
4.1 Introduction 44
4.2 Recognition Method 45
4.2.1 Acquisition of on-line character data 46
4.2.2 Transformation from on-line data to bitmap data 47
4.2.3 Nonlinear normalization 47
4.2.4 Directional features 48
4.2.5 Written-area feature 48
4.2.6 Direction-change features 49
4.2.7 Blurring 53
4.2.8 Pre-classification 53
4.2.9 Recognition 55
4.3 Experiment 55
4.4 Conclusion 65

5. Improvements of Recognition System 67
5.1 Introduction 68
5.2 Adjusting directional, direction-change and written-area features 68
5.2.1 Adjusting the weights of features 68
5.2.2 Experiments 68
5.3 Possibilities for improvements 70
5.4 Conclusion 74

6. Conclusion 75

Appendixes 79

Acknowledgments 93

References 95

Research achievements 101

Chapter 1

General Introduction

1.1 Background and goal of study

In recent years, a pen-based interface has received much attention since it offers easy operation to the user. The good points of a pen-based interface are listed below. The training of this interface is not necessary because a pen is a typical device for writing characters and drawing figures. Even if the writing area is small, as in the size of a card, the characters can be written clearly. Therefore, pen-based equipment for inputting characters can be made smaller than keyboard-based equipment. In addition, a direct pointing-operation with a pen is easier than an indirect use with a mouse. Thus, pen-based hand-held computers such as PDAs (Personal Digital Assistants) are being used for electronically recording personal information. In addition to PDAs, there are many potential applications for pen-based interfaces: inputting customers' names and addresses in jobs at a window, validating credit card signatures, interpreting handwritten notes, electronically mailing handwritten images with the common format called 'electronic ink'(1)(2), writing on electronic whiteboards (3), CAI systems(4) and so on. Generally, tablets are used to detect x, y coordinate data of pen-movements. In recent years, a new kind of pen that can detect these data by acceleration sensors, have been developed (5)(6). This kind of pen, which is convenient to carry about without tablets, may create new applications. In the future, pen-based interfaces will become more and more user-friendly.

A key technology for inputting characters with a pen is on-line character recognition. Methods to recognize characters that are neatly and individually written in single handwriting frames (Fig. 1.1) are being adopted in such applications as PDAs. However, many users are looking forward to a freer, less restrictive method of inputting characters. Therefore, it is important to realize a method to correctly segment a string of characters that have been input by writing them continuously as a string without handwriting frames (Fig. 1.2). Furthermore, there is a need for a highly accurate character recognition method that can recognize cursive-style characters.

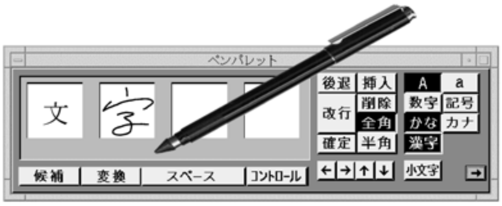


Fig. 1.1 Handwritten characters in single frames

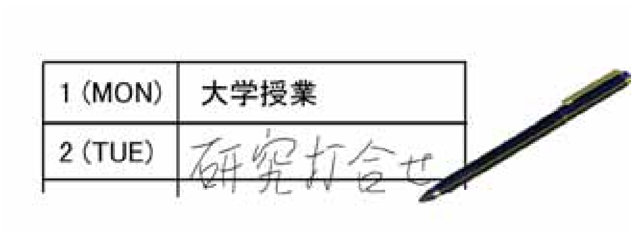


Fig. 1.2 Continuously handwritten characters as a string

This thesis describes our research on an on-line character string separation method and an on-line character recognition method. To segment a string of characters correctly, our string separation method combines physical features such as character pitch with logical features such as character recognition results and language processing results by network expression. Moreover, to correctly recognize cursive-style characters, which have varied shapes, stroke orders and stroke counts, our character recognition method is based on pattern matching that simultaneously uses both directional features, which are off-line features generally used in OCR, and direction-change features, which we designed as on-line features.

1.2 Summary of thesis

This thesis consists of 6 chapters. The background and goal of this thesis are described in Chapter 1. Our on-line character string separation method using network expression is described in Chapter 2. Approaches to on-line character recognition are described in Chapter 3. Our on-line character recognition method that uses both directional features and direction-change features is described in Chapter 4. Possibilities for future work to improve our method based on recognition results are described in Chapter 5. Chapter 6 concludes the thesis. The following paragraphs summarize chapters 2 through 4.

In Chapter 2, we propose a highly accurate string separation method that combines physical features such as character pitch with logical features such as character recognition results and language processing results by network expression. The following conventional string separation method is well known. In this method, string separator candidates are first extracted by physical features. Next, each stroke group between string separator candidates is recognized, and character recognition candidates and degree of similarity are obtained. Finally, separators are determined by judging whether the degrees of similarity are high and whether characters combined from character candidates are correct as a phrase by language processing. However, this method does not use the degree of a separator's likeness to separator candidates based on physical features, and the final separation results are obtained by only using the degree of similarity to characters and language processing results. Therefore, bad effects are sometimes caused by character recognition results and language processing results. For example, string strokes are sometimes incorrectly separated at positions where there should not be separators when there is/are incorrect separator(s) in separator candidates. This method necessarily gives a lot of weight to logical features such as character recognition results and language processing results. Our proposed method obtains separation results by the degree of separator likeness based on both physical features and logical features. In the experimental results, the separation rate is improved over the traditional method from 86.10% to 90.72%. Moreover, in our method, combinations of separator candidates are expressed as network expression with physical features and logical features. By searching for the shortest path of the network, separation results are obtained efficiently. It is important to improve character recognition accuracy to achieve a higher character separation rate.

Chapter 3 gives a summary of traditional on-line character recognition methods and describes our approach. In on-line character recognition, structure analysis approaches are generally used from the point of view of recognition processing time and software size. However, CPU accuracy has recently improved, the cost of memory has rapidly decreased, and various character features can now be used. Therefore, we think a statistical approach holds great promise because it has the advantage of its recognition dictionary being able to learn character features from many handwritten data without any manual work. Moreover, we think it is undesirable to only use off-line features, such as those used in OCRs, to handle free-stroke-order; rather, it is best to actively use effective on-line features based on handwritten strokes in combination with off-line features.

In Chapter 4, the basic theory of our proposed on-line character recognition method and its effectiveness are described. We confirmed that our method is better for handling shape variations. The direction-change features express where and in which direction the character's stroke changes. No previous study has been made on a statistical approach that uses direction-change features and that proves their effectiveness. Moreover, we confirmed that our method can better handle stroke count variations by extracting direction-change features after adding lines called "imaginary strokes" between a stroke's endpoint and the next stroke's starting point in the pen-up state. In a recognition experiment with a public on-line handwritten database (HANDS-kuchibue_d-96-02), recognition rate improved from 77.89% to 86.32% over the traditional method that only uses directional features, thus clearly demonstrating our method's effectiveness.

Chapter 2

On-line Character String Separation Method using Network Expression

SUMMARY

We propose an on-line handwritten character string separation method that uniformly deals with character separation features. This character separation method takes into account both physical and logical features. It is unique in that current methods get the character separation position using a score based on only logical features, such as the character recognition results and the language processing results from separation candidates classified by their physical features. Our methods obtain the character separation position from scores based on both physical features and logical features. We show that the character string separation problem relates to the shortest path problem by expressing character string features as a network expression. We devised our string separation method with Multi-level network expression and Unified network expression. Unifying the physical features and logical features within each network expression improves the separation rate. The separation rate of another method for Japanese kanji strings with only logical features is 85.5%. The separation rate of our first method with the Multi-level network expression is 86.7 %. The separation rate of our second method with the Unified network expression is 90.7%. Furthermore, we can speed up the string separation process with the Unified network expression.

2.1 Introduction

The idea of a pen-type interface has been received with great interest as a means of input [7]. Many effective on-line character recognition methods are being investigated for easy text input. Generally, in order to input a Japanese character string using a pen-type interface, writers must individually input characters in handwriting frames. However, this is a troublesome operation for writers. The entry frame should be eliminated, and a method to take notes using continuous strings of characters should be created. Therefore, a character string separation technology for separating each character from a string of characters is important.

It is difficult to correctly obtain the handwritten character separation position because the character pitches and stroke intervals vary [8] and characters sometimes overlap.

For on-line string separation technology, methods [9-13] that use logical features, such as character recognition results and language processing results, and a method [14] that considers individual fluctuation tendencies have also been examined. Other methods [15][16] using logical features for handprinted string separation technology have also been studied.

These methods first classify the separation candidate based on physical features such as character pitch. They then get the separation position using only logical features from separation candidates. These methods disregard the physical feature score when separating characters. We think that a method using only logical feature scores is not sufficient in correctly obtaining the separation position.

In this chapter, we present online handwritten string separation methods that unify physical features and logical features with Multi-level network expressions and Unified network expressions. We will then describe the improvements observed when using our methods.

2.2 Character separation features

For online handwritten character separation, the physical features, as shown in Fig.2.1, and the logical features, such as the degree of character recognition similarity and language processing results, are well known [14]. Our methods use the same features as shown in Table 2.1, but our methods are different from other methods in the separation process, as shown below.

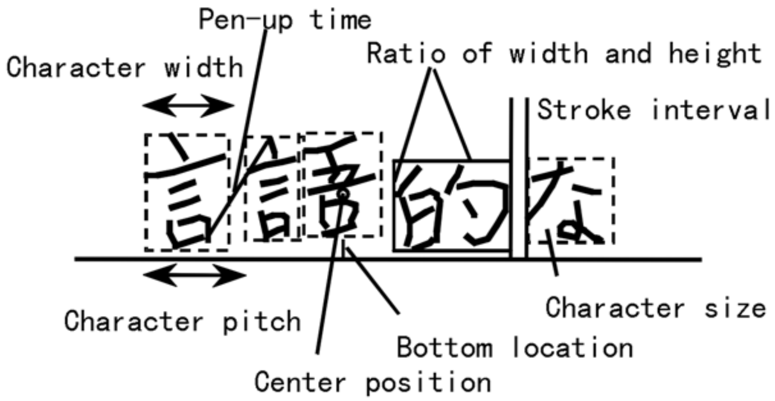


Fig.2.1 Physical features

Table 2.1 Features for handwritten character string separation

Physical features	Logical features
(1) Stroke interval (2) Pen-up time (3) Character width (4) Ratio of width and height (5) Character size (6) Bottom location (7) Center position (8) Character pitch	(1) Character recognition degree of similarity (2) Language processing result

2.3 String separation method using physical and logical features

2.3.1 Separation process with a traditional method
(Murase's method using Lattice method)

The typical method of string separation [9][10][14] is shown in Fig.2.2. This method first classifies the separation candidates by physical features. For example in this figure, five separator candidates are extracted and K separation candidates are selected from all combinations of separator candidates. Character recognition candidates are then obtained by examining the degree of similarity of stroke groups between separation points. The final separation is achieved by performing language processing on the character recognition candidates. This example shows when the final result is correct. However, this method sometimes causes mistakes as shown in Sec.6.3. Note that separation scores are not used in obtaining the final separation result [17][18].

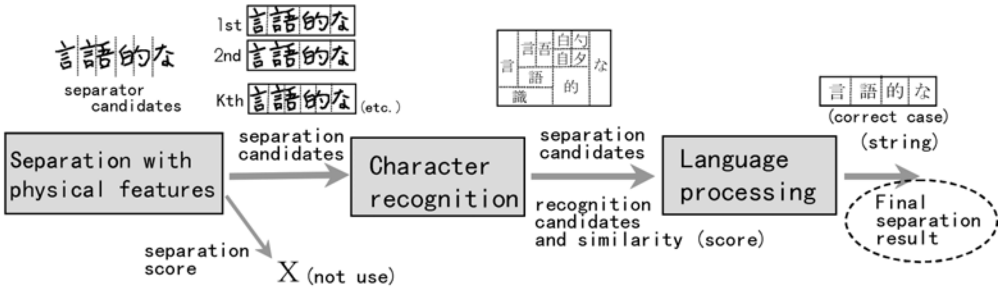


Fig.2.2 Separation process using traditional method

We devised our Multi-Level network expression method and the Unified network expression using both physical and logical feature scores.

2.3.2 Separation process with Multi-level network expression

Our first method, the Multi-Level Network Expression shown in Fig.2.3, first classifies the separation candidates on the basis of physical features in Network A. Once the process enters Network B, a Character Recognition phase is entered, and recognition candidates and the degree of similarity are passed to the Language Processing phase. The method then derives the language processing score for each separation candidate and passes this result to final separation processing. The separation result is obtained by adding the physical feature score and the language processing score.

This method considers not only the logical feature score but also the physical feature score from the initial separation when obtaining the final separation result.

This method creates two networks and searches K shortest paths from one network to the other, where K is the candidate count. After this is done, the physical and logical feature scores are obtained.

K shortest paths [19][20] are searched in the following way. The score of a path between position N and the start position is calculated by adding the score of the path between position N-1 and position N to the score of the path between position N-1 and the start position. Position N is moved from the start position to the end position in the network, while only the K paths with the higher score remain.

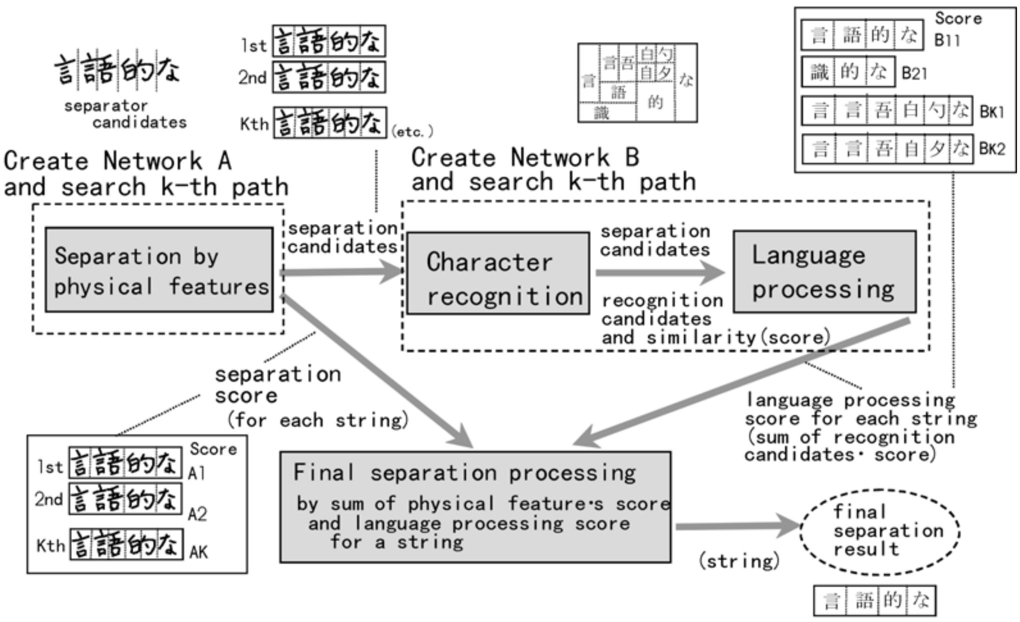


Fig.2.3 Separation process with Multi-Level network expression

2.3.3 Separation process with Unified network expression

Our second method uses a Unified Network Expression shown in Fig.2.4. Separation candidates are individually obtained and classified by separation scores. Character recognition is then performed on each set of separation candidates, and the degree of similarity to characters are sent to the Language Processing phase where a language processing score is assigned to each phrase. The sum of the scores of the logical and physical features of each phrase are then passed onto a final phase. The final separation result selects a suitable separation candidate among some combinations of phrase candidates that have the highest sum of physical and logical feature scores.

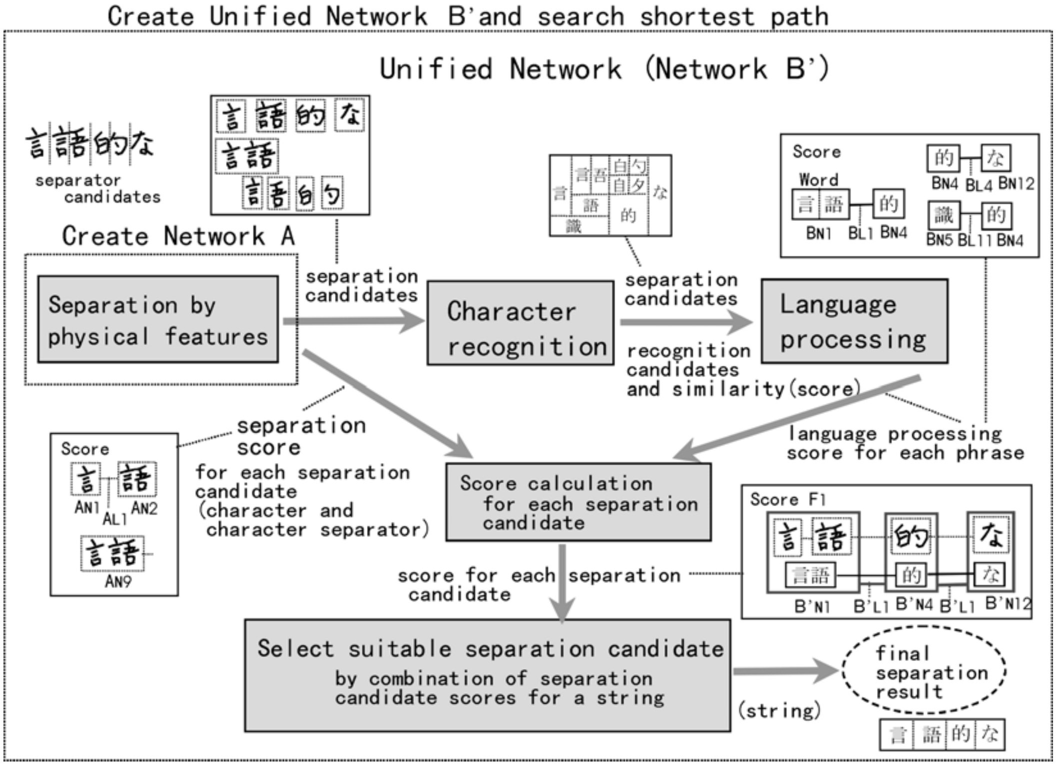


Fig.2.4 Separation process with Unified network expression

2.4 Multi-level network expressions

In this section we propose a Multi-level Network Expression as an effective unification method distinguishing string separation features.

2.4.1 Network expression (A)

Network Expression A is the expression to efficiently obtain the string separation candidates on the basis of the physical features.

2.4.1.1 Extraction of base segments

The stroke interval between stroke #N and stroke #N+1, where N shows the handwriting order, indicates X direction distance between the rectangular area that encloses the stroke group from stroke #1 to stroke #N, and the area that encloses the stroke group from the tail stroke to the stroke #N+1. An example of the stroke interval is shown in Fig.2.5.

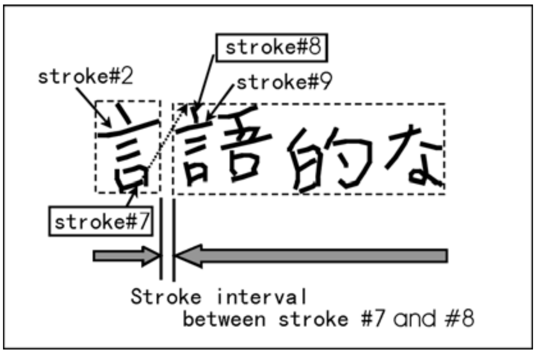


Fig.2.5 Stroke interval

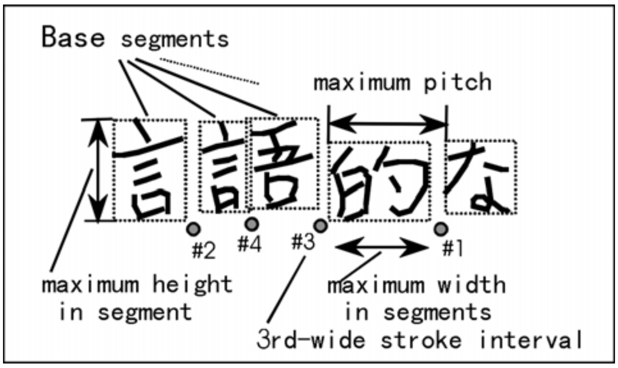


Fig.2.6 Base segments

A base segment is a stroke group that cannot be separated further by the stroke interval and pen-up time thresholds. Strokes are merged into a base segment either when their stroke interval is less than the stroke interval threshold, or when their pen-up time is less than the pen-up time threshold. A character is a combination of base segments. In Fig.2.6, the stroke groups indicated with rectangles are each base segment.

We analyzed the distributions of the physical features of the strings in the handwritten data set 1 (learning data) shown in Table 2.6. Based on the distributions of the pen-up times in data set 1, if the pen-up time is 0.1 seconds or less, it is in a character, and if the pen-up time is 2.0 seconds or more, it is between characters.

The maximum stroke interval in a character is set for each height in the character string, and based on Table 2.2, the minimum stroke interval between characters is set for each height in the character string. A negative stroke interval means an overlapping of strokes. If the stroke interval between the target stroke is larger than the maximum stroke interval in a character, the target stroke area is between characters. If the stroke interval between the target stroke area is smaller than the minimum stroke interval between characters, the target stroke area is in a character.

Therefore, we defined the pen-up time threshold as 0.1 seconds, and the stroke interval threshold for each string height is the minimum stroke interval as shown in Table 2.2.

Table 2.2 Threshold of stroke interval for base segment

String height	< 80	80	90	100	110	120	130	140	150	160	170	180	180 <
Minimun interval between characters (*SEGintMin*) [unit:0.1mm]	0	0	-10	-20	-20	-20	-30	-30	-30	-30	-30	-30	-30
Maximum interval in a character (*SEGintMax*) [unit:0.1mm]	30	30	30	40	40	45	45	45	50	50	50	50	50

2.4.1.2 Standard value for physical features

The physical features vary from writer to writer. Thus, we define the standard value for physical features from an inputted handwritten string every time a string is written. In our method, the ratio of the maximum width of base segments and the maximum height of base segments define the standard ratio of character width and character height. The standard character width and the standard character pitch are also defined from the input string.

The distributions of the physical features such as character pitch, character width, ratio of width and height, center position, bottom location and character size in data set 1 were analyzed. These distributions are shown in Table 2.3.

Table 2.3 Distributions of features

Features	Average	Standard deviation	
		Value	Label
$\log_2$(Height / Width)	0.16	0.62	*Dev_ratio*
Character pitch	12.8	33.5	*Dev_pitch*
$\log_2$(Character width / Character pitch)	-0.28	0.33	*Dev_width*
Center position / String height	0.03	0.08	*Dev_cent*
Bottom location / String height	0.11	0.1	*Dev_bottom*
(Character width * Character height) / (String height * String height)	0.68	0.22	*Dev_size*

The standard value for each physical feature is obtained by the following expressions based on Table 2.3.

$$W = maximum\ width\ of\ the\ base\ segment \quad \text{.................} (1)$$

$$P = maximum\ pitch\ of\ the\ base\ segment \quad \text{..................}(2)$$

$$H = maximum\ height\ of\ the\ base\ segment \quad \text{................} (3)$$

For example in Fig.2.6, the widest base segment is the 4th base segment from the left side. The maximum pitch is the pitch between the 4th base segment and 5th base segment. The highest base segment is the 1st segment.

(a) Standard ratio of width and height (Sr)

$$Sr = H/W \quad \text{......(4)}$$

(b) Standard character width (Sw)

$$Sw = W \qquad [\, when\ Log_2 Sr \le MaxR \,] \quad \text{......(5)}$$

$$Sw = W + Np \qquad [\, when\ Log_2 Sr > MaxR \,] \quad \text{......(6)}$$

where

$$MaxR = Average\ of\ \ learning\ data's\ Log_2(height\ /\ width) + Dev\_ratio$$
$$= 0.16 + 0.62 = 0.78 \quad \text{......(7)}$$

$$Np = the\ character\ pitch\ between\ the\ base\ segment\ that\ has\ maximum\ width\ and\ its\ nearest\ neighboring\ base\ segment \quad \text{......(8)}$$

(c) Standard character pitch (Sp)

$$Sp = P \qquad [\, when\ MinP \le P \,] \quad \text{......(9)}$$

$$Sp = P + Np \qquad [\, when\ MinP > P \,] \quad \text{......(10)}$$

where

$$MinP(Minimum\ character\ \ pitch\ value) = Ap - Dev\_pitch \quad \text{......(11)}$$

$$Ap = Avp \qquad [\, when\ H < 100\ (unit : 0.1mm) \,] \quad \text{......(12)}$$

$$Ap = Avp + 5/8 \times (H - 100) \qquad [\, when\ H \ge 100\ (unit : 0.1mm) \,] \quad \text{......(13)}$$

$$Avp = Average\ of\ \ character\ pitch\ = 128 \quad \text{......(14)}$$

(d) Standard character size (Ss)

$$Ss = H \times Sw \quad \text{......(15)}$$

(e) Standard bottom location (Sb)

$$Sb = Average\ of\ (bottom\ location\ \ /\ string\ \ height\ ) = 0.11 \quad \text{......(16)}$$

(f) Standard center position (Sc)

$$Sc = Average\,of\ (\,center\ \ position\ /\ string\ \ height\,) = 0.03 \quad (17)$$

(g) Standard character count ($Scount$)

$$Scount = Integer\left(\frac{String\ Lengh + Sp - Sw}{Sp} + 0.5\right) \quad (18)$$

(h) Minimum stroke interval ($LiMin$), Maximum stroke interval ($LiMax$), and Middle stroke interval (Mi)

$$LiMin = SEGintMin \quad (cf.\,Table2) \quad (19)$$

$$LiMax = SEGintMax \quad (cf.\,Table2) \quad (20)$$

$$Mi = (\,(Scount-1)th\,widest\,stroke\,interval\,) + (Scount-2)th\,widest\,stroke\,interval\,)\,)/2 \quad [\,when\,Scount \geq 2\,] \quad (21)$$

$$Mi = widest\,stroke\,interval \quad [\,when\,Scount = 1\,] \quad (22)$$

In Fig.2.6, when $Scount$ is 4, Mi is the average of the stroke interval of the 3rd-widest stroke interval and that of the 2nd-widest stroke interval.

(i) Minimum pen-up time ($LuMin$), Maximum pen-up time ($LuMax$) and middle pen_up time (Mu)

$$LuMin = 0.1\,(m\sec) \quad (23)$$

$$LuMax = 0.7\,(m\sec) \quad (24)$$

$$Mu = (\,(Scount-1)th\,longest\ pen\text{-}up\,time\,) + (Scount-2)th\,longest\ pen\text{-}up\,time\,)\,)/2 \quad [\,when\,Scount \geq 2\,] \quad (25)$$

$$Mu = longest\ pen\text{-}up\,time \quad [\,when\,Scount = 1\,] \quad (26)$$

2.4.1.3 Extraction of estimation value for separation candidate

The separation candidate is obtained as a combination of base segments. An example of separation candidates is shown in Fig.2.2 and 2.3. The estimation value of the character likeness and character separator likeness is obtained for each feature type in respect to each character's stroke group candidate and character separator candidate. A character's stroke group candidate is a group of the base segments between separator candidates. For example, in Fig.2.2 and 2.3, "言-語-的-な"," 言語-的-な" are separation candidates, and "言","語" are character's stroke group candidates. The character likeness and character separator likeness for each feature type are expressed with symbols as shown below.

When the stroke interval is narrow, this is in a character. When the stroke interval is wide, this is between characters. Thus, the stroke interval is used for estimating both character likeness and separator likeness. When the pen-up time is short, this is in a character. When the pen-up time is wide, this is between characters. So, the pen-up time is also used for estimating both character likeness and separator likeness.

[Estimated value as a character]

(a) *Fa* (estimated value for the character size feature)

$$Fa = Min\left(1, \left|\frac{Is}{H \times H} - \frac{Ss}{H \times H}\right| \Big/ (Dev\_size \times 2)\right) \quad \text{......................................(27)}$$

$$where \quad Is = character\ size\ of\ each\ character\ candidate$$

(b) *Fb* (estimated value for the bottom location feature)

$$Fb = Min\left(1, \left|\frac{Ib}{H} - Sb\right| \Big/ (Dev\_bottom \times 2)\right) \quad \text{..(28)}$$

$$where \quad Ib = bottom\ location\ of\ each\ character\ candidate$$

(c) *Fc* (estimated value for the center position feature)

$$Fc = Min\left(1, \left|\frac{Ic}{H} - Sc\right| \Big/ (Dev\_cent \times 2)\right) \quad \text{...(29)}$$

$$where \quad Ic = center\ position\ of\ each\ character\ candidate$$

(d) *Fd* (estimated value for ratio of width and height feature)

$$Fd = Min\left(1, \left|Log_2 Ir - Log_2 Sr\right| \Big/ (Dev\_ratio \times 2)\right) \text{.......................................}(30)$$

$where \quad Ir = ratio\ of\ width\ and\ height\ of\ each\ character\ candidate$

(e) *Fe* (estimated value for the character width feature)

$$Fe = Min\left(1, \left|Log_2\left(\frac{Iw}{Ap}\right) - Log_2\left(\frac{Sw}{Ap}\right)\right| \Big/ (Dev\_width \times 2)\right) \text{...............................}(31)$$

$where \quad Iw = character\ width\ of\ each\ character\ candidate$

(f) *Ff* (estimated value for the stroke interval feature)

Wen the input stroke interval is Mi (middle stroke interval), the character likeness is 0.5, which means that the interval can be in the character or between the characters with a 50-50 chance.

$$Ff = 0 \quad [when\ Ii \le LiMin] \qquad \text{..}(32)$$
$$Ff = 1 \quad [when\ Ii \ge LiMax]$$
$$Ff = ((Ii - LiMin)/(Mi - LiMin)) \times 0.5 \quad [where\ LiMin < Ii \le Mi]$$
$$Ff = 0.5 + ((Ii - Mi)/(LiMax - Mi)) \times 0.5 \quad [where\ Mi < Ii < LiMax]$$
$$where\ \ Ii = Stroke\ Interval\ of\ each\ character\ candidate$$
$$Mi = Middle\ stroke\ interval$$

(g) *Fg* (estimated value for pen-up time feature)

$$Fg = 0 \quad [when\ Iu \le LuMin] \qquad \text{..}(33)$$
$$Fg = 1 \quad [when\ Iu \ge LuMax]$$
$$Fg = ((Iu - LuMin)/(Mu - LuMin)) \times 0.5 \quad [where\ LuMin < Iu \le Mu]$$
$$Fg = 0.5 + ((Iu - Mu)/(LuMax - Mu)) \times 0.5 \quad [where\ Mu < Iu < LuMax]$$
$$where\ \ Iu = pen - up\ time\ of\ each\ character\ candidate$$
$$Mu = Middle\ pen\_up\ time$$

$$(like\ a\ charcter)\ 0 \le Fa, Fb, Fc, Fd, Fe, Ff, Fg \le 1$$

[Estimated value as a separation]

(h) *Fh* (estimated value for the character pitch feature)

$$Fh = Min\left(1, \frac{|Ip - Sp|}{Sp}\right) \qquad (34)$$

where Ip = *character pitch of each character candidate*

(i) *Fi* (estimated value for the stroke interval feature)

$$Fi = 1 - Ff \qquad (35)$$

(j) *Fj* (estimated value for the pen-up time feature)

$$Fj = 1 - Fg \qquad (36)$$

$$(like\ a\ separator)\ 0 \le Fh, Fi, Fj \le 1$$

The character likeness V_N and character separator likeness V_L based on all feature levels for each character's stroke group candidate and character separator candidate are expressed as follows:

$$V_N = \frac{WaFa + WbFb + WcFc + WdFd + WeFe + WfFf + WgFg}{Wa + Wb + Wc + We + Wf + Wg} \qquad (37)$$

$$V_L = \frac{WhFh + WiFi + WjFj}{Wh + Wi + Wj} \qquad (38)$$

where

$Wa, Wb, Wc, Wd, We, Wf, Wg, Wh, Wi$ *and* Wj *are the weight value with respect to the estimation value*

2.4.1.4 Classification of separation candidates with Network A

The separation candidate is obtained from a combination of base segments. The "correct" combination is obtained as a conclusion of the network expression's K shortest paths problem. The combination of base segments is expressed as a network using the character's stroke group candidate as a node and the character separator candidate as a link. An example of the network expression is shown in Fig.2.7.

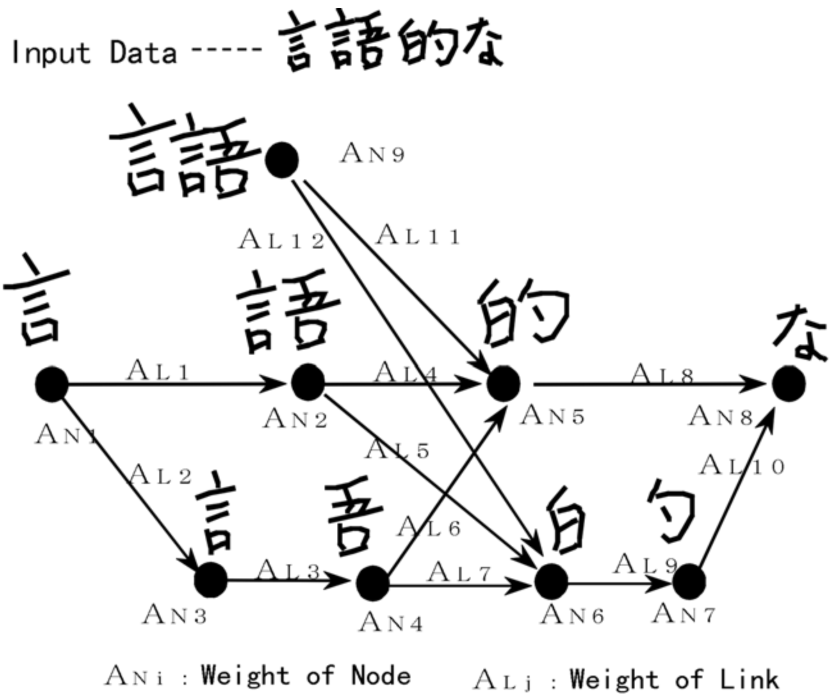


Fig.2.7 Network expression (A)

The node weight, A_N, and link weight, A_L, in Network A are defined as shown below.

$$A_N = V_N\,(\mathit{estimated\ value\ as\ a\ character}) \times \mathit{Base\ segment\ count\ of\ a\ node} \quad \text{.......}(39)$$

$$A_L = V_L\,(\mathit{estimated\ value\ as\ a\ separator}) \times \mathit{Base\ segment\ count\ of\ a\ former\ (left)\ node} \quad \text{.......}(40)$$

In Fig.2.7, the base segment count (stroke group count) of A_{N3} (node 3) is 1 and that of A_{N4} (node 4) is 1. The base segment count of A_{N2} (node 2) is 2 because A_{N2} consists of the base segments of A_{N3} and A_{N4}. In Equations (39)(40), the node weight (A_N) includes a base segment count of the node, and the link weight(A_L) includes a base segment count of former nodes because each path cost, whose node count and link count are unique, is obtained fairly.

For example, when considering the paths from node 1 to node 5, the path of node 1-2-5 [“言”-“語”-“的”] has one node (node 2)[“語”] between node 1 and node 5, but the path of node 1-3-4-5 [“言”-“言”-“吾”-“的”] has two nodes (node 3,4) [“言”,“吾”] between node 1 and node 5. Therefore, the weight (A_{N2}) of node 2 is obtained by calculating the estimated value (V_{N2}) as a character of node 2 by the base segment count (=2) of node 2. The path of node 1-2-5 has two links, but the path of node 1-3-4-5 has three links. The link weight (A_{L6}) between nodes 2 and 5 is obtained by calculating the estimated value (V_{L6}) by the base segment count (=2) of the former node (node 2). The other link weights from nodes 1 to 5 are obtained by calculating the estimated values by the segment counts (= all 1) of the former nodes.

While searching the path (critical path) of the network, the weight of each node is changed to the weight of its respective hypothetical link to reach the K shortest paths problem of a simple network that only includes the weight of the links (Fig.2.8).

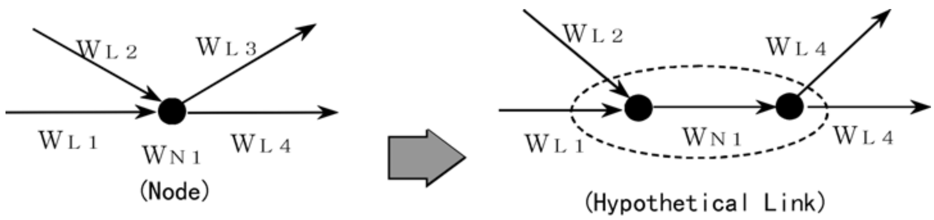


Fig.2.8 Change the weight of a node to the weight of a hypothetical link

When searching the path (critical path) of the network, the weight of each node (hypothetical link) and its respective link are added to the sum. The separation candidate of the path that has the minimum sum of the weights of its node links is the suitable candidate. Some separation candidates (paths) are obtained in order of sum by searching Network A. This searching is done by the search path algorithm called "K shortest paths". K refers to the number of separation candidates. Every separation candidate has a physical feature score. For example, the physical score of the candidate "言"(*gen*)-"語"(*go*)-"的"(*teki*)-"な"(*na*) in Fig.2.7 is the sum of A_{N1}, A_{L1}, A_{N2}, A_{L4}, A_{N5}, A_{L8} and A_{N8}.

The weight values Wa to Wj were changed from 0 to 30 in increments of 1, and each weight value for the best possible separation rate was set in respect to Data set 1.

These weight values are shown in Table 2.4

Table 2.4 Weight values for physical features

Weight	*Wa*	*Wb*	*Wc*	*Wd*	*We*	*Wf*	*Wg*	*Wh*	*Wi*	*Wj*
Value	0	12	1	12	18	12	10	22	18	16

2.4.2 Network expression (B)

Network expression B is the expression used to find the string separation candidates by efficiently using logical features.

Network B is created as the combination of character recognition candidates for the characters of the separation candidates obtained in Network A. First, the character recognition degree of similarity is obtained by character recognition. Next, characters from the string separation candidates are collected, and a morpheme (word) is extracted.

2.4.2.1 Character recognition

Each stroke group from the string separation candidates is recognized and its character recognition degree of similarity is obtained.

We used a character recognition method with Directional Features and Direction-Change Features [21]. The recognition rates of this method for the odd data of the on-line Japanese handwritten data base (TUAT Nakagawa Lab. HANDS-kuchibue_d-96-02)[22] are 87.97% for kanji characters, 77.37% for non-kanji characters, and 82.37% for all Japanese characters. Non-kanji refers to hiragana, katakana, numeric, alphabetic, and symbolic characters.

2.4.2.2 Extraction of morpheme candidates

After character recognition, the estimated values of the words and the phrases of separation candidates are obtained by language processing. Some morpheme candidates are extracted [23-33] by a combination of character recognition candidates from string separation candidates using the word language dictionary. We use a dictionary with approximately 50,000 Japanese words and the frequency information shown in Table 2.5. Next, morpheme candidates and grammatical connective-costs between each morpheme candidate and its neighbor are obtained by a grammar dictionary that describes 16 levels of connective-cost for 2,688 kinds of morpheme connections.

Table 2.5 Word language dictionary

Verb	9,807		Adverb	1,356
Noun	33,713		Conjunction	52
Adjective	567			500
Adjective verb	2,025		Particle	
Prefix	481		Auxiliary	
Suffix	876		Inflection	

2.4.2.3 Language processing with Network B

Network B is created by expressing the estimation value (score) of the morpheme as the weight of the node and the estimation value of the morpheme's connection as the weight of link. The weight (B_{Ni}) of the node and the weight (B_{Li}) of the link in Network B are defined below.

$$B_{Ni} = \frac{\sum_{j=1}^{m}(1 - R_j)}{m} - (m-1)\times\alpha - f\times\beta \quad \text{.......................................}(41)$$

where

R_j = *character recognition degree of similarity of each character of a morpheme candidate* $[0 \le R_j \le 1(\textit{high score})]$

m = *character count of a morpheme candidate*

f = *frequency of a mopheme candidate* $[0 \le f \le 15\ (\textit{high frequency})]$

$\alpha\,(\textit{weight of character count}) = 0.1,\ \beta\,(\textit{weight of frequency}) = 0.01$

$$B_{Li} = \gamma / g \quad \text{...}(42)$$

where

g = *grammatical connective-cost between a morpheme candidate and a former (left) morpheme candidate* $[0 \le g \le 15(\textit{best suitable})]$

$\gamma\ (\textit{weight of grammatical connective-cost}) = 300$

Fig.2.9 shows an example of Network B. This figure contains the word " 言語 " (*gengo*). The pattern containing " 言 "(*gen*) and " 語 "(*go*) is sometimes recognized as a character " 識 "(*shiki*) by mistake.

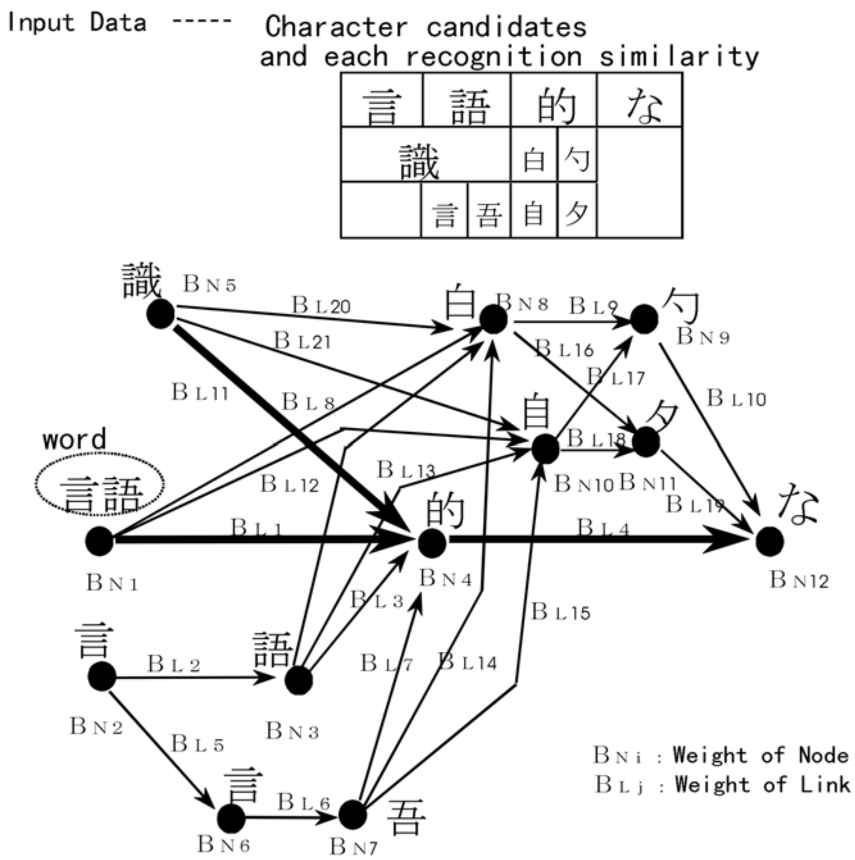


Fig.2.9 Network expression (B)

By solving the K shortest paths problem of Network B, some (=K) suitable paths (phrases) and estimated values (scores) for these paths are obtained. In this figure, the suitable paths (phrases) are [" 言 "(*gen*) " 語 "(*go*)]-" 的 "(*teki*)-" な "(*na*) (B_{N1}-B_{L1}-B_{N4}-B_{L4}-B_{N12}) and " 識 "(*shiki*)-" 的 "(*teki*)-" な "(*na*)(B_{N5}-B_{L11}-B_{N4}-B_{L4}-B_{N12}). The other paths are not good phrases.

2.4.2.4 Final separation result with Network A and Network B

Each score (estimated value) of the string separation candidate in Network A and candidates in Network B are added together without the weight of each value in Network A and B. The final separation result is obtained on the basis of the sum.

In Fig.2.10, the final result is [" 言 "(*gen*) " 語 "(*go*)]-" 的 "(*teki*)-" な "(*na*), because its total score is best.

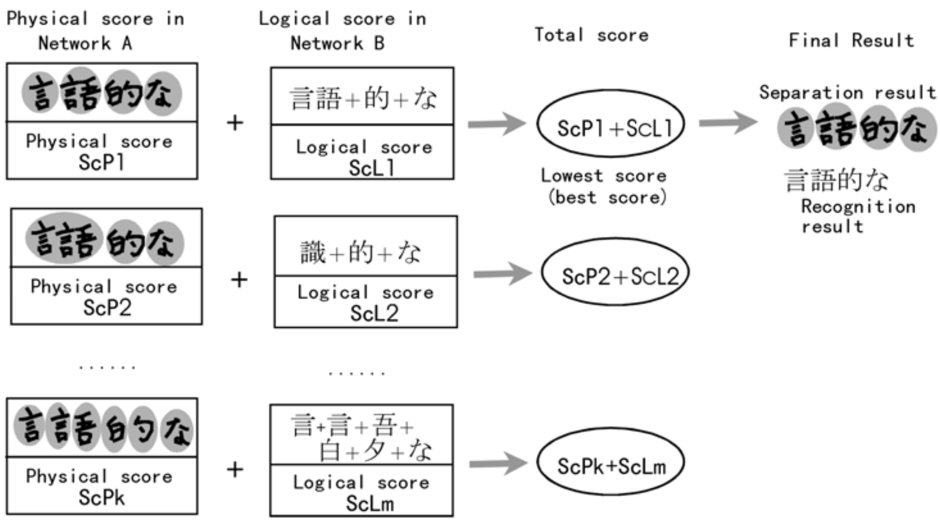


Fig.2.10 Final separation with sum of physical and logical scores

Murase's method obtains the final separation result only on the basis of the scores of candidates in Network B and does not consider the physical score. Therefore, Murase's method sometimes selects an incorrect candidate " 識 "(*shiki*)-" 的 "(*teki*)-" な "(*na*) as the final separation result. Our method considers the logical score as well as the physical score. Therefore, an incorrect candidate is not selected and the separation rate is improved.

2.5 Unified network expressions

In this chapter, we explain our second method using Unified network expressions. This method obtains the best separation results by unifying each network level. The Unified network expression executes the separation process, recognition process, language process, and calculations of the sum of the physical score and the language score in only one network expression.

2.5.1 Combination of network

The combination is gradually carried out from the low-level (only physical) network to the high-level (physical and logical) network. Some nodes in the low-level are unified and one new node in the high-level is created. At this time, the weight of nodes and links for the low- level network merges to the high-level network by the method shown in Fig.2.11. For example, the character recognition results ("言"(*gen*) and"語"(*go*)) of A_{N2} and A_{N3} are combined to form a word ("言語"(*gengo*) by language processing. In other words, the weight of the inside low-level nodes and the weight of the inside low-level links are combined for the weight of the high-level node. The weights of both sides of the low-level links are combined for the weight of high level links.

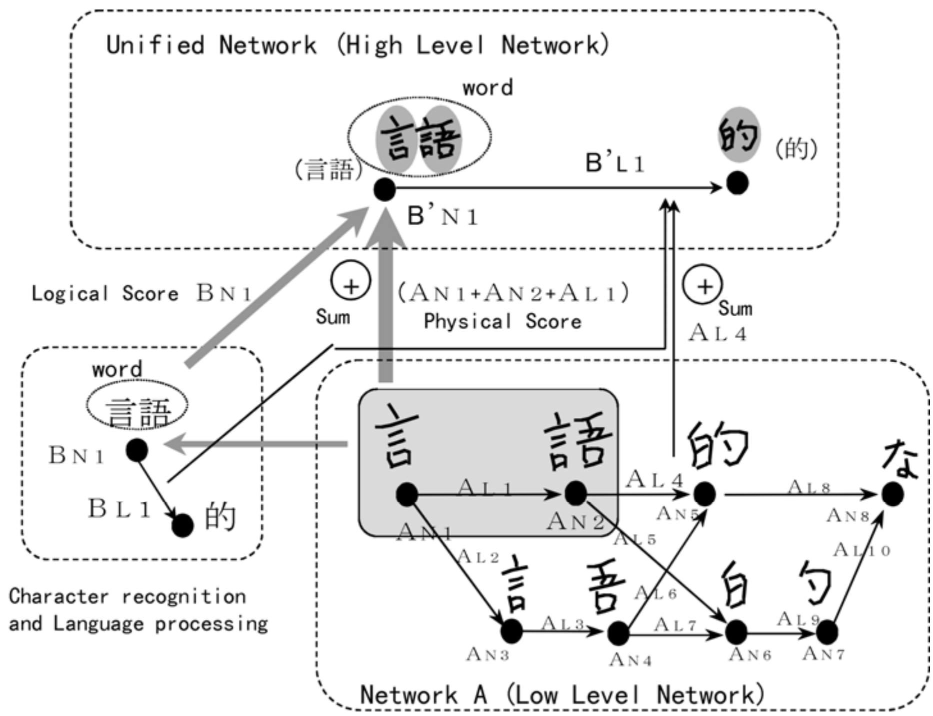


Fig.2.11 The succession of the node and the link weights

2.5.2 Separation process with Unified networks

The weights (B_{Ni},B_{Li}) of the nodes and links in the character recognition and the language processing phase are defined below.

$$B_{Ni} = \left\{ \frac{\sum_{j=1}^{m}(1 - R_j)}{m} - (m-1)\times\alpha - f\times\beta \right\} \times E_i \quad \text{.......................................(43)}$$

where

R_j = character recognition degree of similarity each character of a morpheme candidate $[0 \le R_j \le 1 (\text{high score})]$

m = character count of a morpheme candidate

f = frequency of a morpheme candidate $[0 \le f \le 15 \ (\text{high frequency})]$

α (weight of character count) $= 0.1$, β (weight of frequency) $= 0.01$

E_i = Base segment count of a morpheme candidate

$$B_{Li} = \gamma / g \times E_{former} \quad \text{.......................................(44)}$$

where

g = grammatical connective - cost between a morpheme candidate and a former (left) morpheme candidate $[0 \le g \le 15 (\text{best suitable})]$

γ (weight of grammatical connective - cost) $= 300$

E_{former} = Base segment count of a former (left) morpheme candidate

As in Network A, the weights (B_{Ni})(B_{Li}) of nodes and links in Network B of the Unified network include the base segment counts shown in Equation (43)(44).

In Fig.2.11, the weights (B'_N, B'_L) of the nodes and links in the Unified network are expressed as the sum of weights (A_N, A_L) of the nodes and links in Network A and the weights (B_N,B_L) of the nodes and links during character recognition and language processing. For example, the weights of the nodes and links in Fig.2.11 are obtained as shown below.

$$B'_{N1} = B_{N1} + (A_{N1} + A_{N2} + A_{L1}) \quad \text{.......................................(45)}$$

$$B'_{L1} = B_{L1} + (A_{L4}) \quad \text{.......................................(46)}$$

The weights of the nodes and links in Network B of the Multi-level network expression shown in Fig.2.9 express only the score (the estimated value) for the logical features. The weights of nodes and links in the Unified expression shown in

Fig.2.12 express the score (the estimated value) for both the physical features and the logical features.

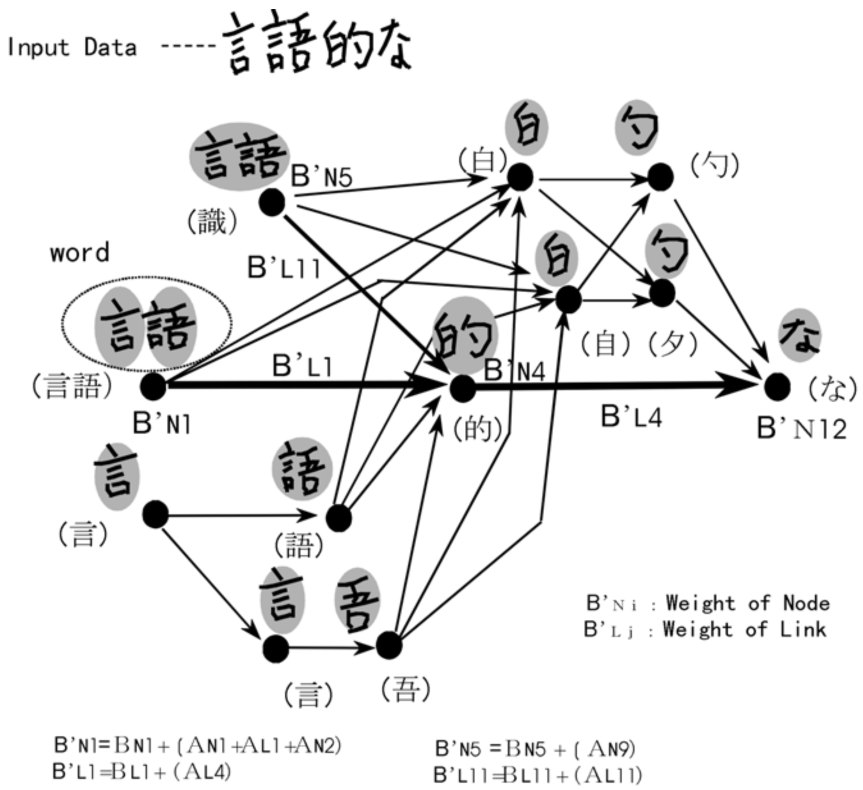


Fig.2.12 Unified Network Expression

In this method, each node (B'_N) and link (B'_L) has the physical feature score (A_N, A_L) and the logical feature score (B_N,B_L). The final separation result is obtained by solving the shortest path problem in this Unified network. In this network, the path of [" 言 "(*gen*) " 語 "(*go*)]-" 的 "(*teki*)-" な "(*na*) (B'_{N1}-B'_{L1}-B'_{N4}-B'_{L4}-B'_{N12}) and the path of " 識 "(*shiki*)-" 的 "(*teki*)-" な "(*na*) (B'_{N5}-B'_{L11}-B'_{N4}-B'_{L4}-B'_{N12}) have a good score as far as language is concerned. However, the score of the node B'_{N1} (" 言語 " (*gengo*)) is better than the score of the node B'_{N5} (" 識 "(*shiki*)), because the physical score (A_{N1}+A_{L1}+A_{N2}) of the node B'_{N1} (" 言語 "(*gengo*)) is better than the physical score (A_{N9}) of the node B'_{N5} (" 識 "(*shiki*)). Therefore, the path of [" 言 "(*gen*) " 語 "(*go*)]-" 的 "(*teki*)-" な "(*na*) (B'_{N1}-B'_{L1}-B'_{N4}-B'_{L4}-B'_{N12}) is the final result.

2.5.3 Effects of Unified network

Murase's method (10) does not consider physical features because the final result in his method is the same as the result of Network B using only logical features. Our methods (Multi-level network and Unified network) consider the physical features in the network expressions.

In a Multi-level network, creating Network B from separation candidates based on all paths in Network A and their paths' character recognition candidates results in Network B to count the total number of paths in Network A many times. When a string consists of many characters, the processing time searching Network B becomes enormous. Therefore, the separation candidate counts need to be limited in a Multi-level network.

In a Unified network, despite many path counts, the Unified network count is always one and only the shortest path in this network is obtained, so the processing time in the Unified network is insubstantial.

Network B (language processing) in the Multi-level network doesn't consider the number of base segments. However, Network B in the Unified Network does consider the number of base segments as shown in Equations (43)(44) by matching nodes and links in Network B to nodes and links in Network A as in Fig.2.11. Therefore, the Unified network is more suitable than the Multi-level network or Murase's method in obtaining the correct string separation results.

2.6 Experiment

2.6.1 Handwritten data sets

We experimented with character string separation using handwritten data freely written by 42 people regarding character size, shape, and pitch using a regular pen on a pressure-type LCD tablet. The resolution and the digitizing speed of this tablet were 76 dpi and 75 points per second. The string examples from data set are shown in Fig.2.13, where KANJI means words consisting of only Japanese kanji characters, MIX means phrases consisting of Japanese kanji, hiragana, katakana, numeric, alphabetic and symbolic characters. There are many strings where the shapes of characters are of low quality. Many strings also have varied stroke intervals between each character.

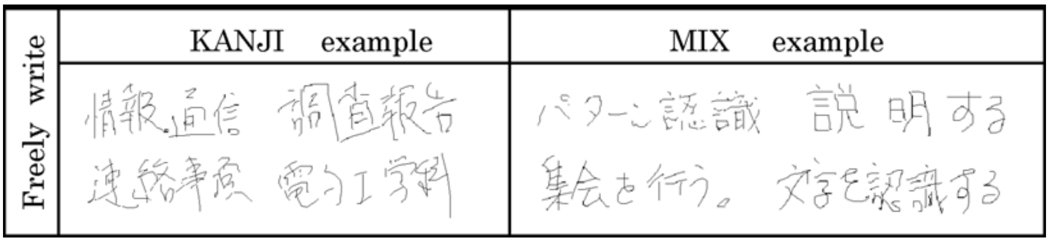


Fig.2.13 Example of data set

The entire data set consists of data set 1 written by 21 people and data set 2 written by another 21 people as shown in Table 2.6, where ALL means the sum of the KANJI string count and the MIX string count.

We used data set 1 to get the stroke interval threshold for basic segments as shown in Table 2.2, the distribution of features as shown in Table 2.3, and the weight values for physical features as shown in Table 2.4. Then, we set the standard value for physical features as shown in section 2.4.1.2. We used data set 1 as learning data and data set 2 as unknown data to estimate the string separation rate.

Table 2.6 Handwriting Data

Data set	Writer	String kind	Written string count per each character count of a string										Total	Average character count of a string
			2	3	4	5	6	7	8	9	10	11		
Data set 1 (Learning)	21 people	KANJI	111	145	128	79	23	0	8	0	0	0	494	3.57
		MIX	0	93	103	38	66	49	58	18	1	11	437	5.44
		All	111	238	231	117	89	49	66	18	1	11	931	4.45
Data set 2 (Unkown)	21 people	KANJI	111	141	125	77	30	0	6	0	0	0	490	3.59
		MIX	0	92	103	39	63	40	56	25	0	13	431	5.47
		All	111	233	228	116	93	40	62	25	0	13	921	4.47

(MIX: phrases consist of Japanese Kanji, Hiragana, Katakana, numeric, alphabetic and symbolic characters)

2.6.2 Experimental results

We compared the following four methods.

Method 1: method using **only physical features**

(The method that uses only Network A and carries out character recognition after classification by physical features.)

Method 2: **Murase's method** using **only logical feature scores** (10)

(The method that gets separation candidates from Network A and carries out character recognition and language processing from Network B then returns a result.)

Method 3: our 1st method using the **Multi-level network** expression

Method 4: our 2nd method using the **Unified network** expression

The maximum character recognition candidate count in each method is 5.

We obtained the suitable maximum separation candidate count of Network A to get high string separation rates in the Multi-level network (Method 3) from preliminary experiments. The string separation rates when changing maximum separation candidate counts using handwritten data set 1(learning data) are shown in Table 2.7. The string separation rate is defined below.

$$\textit{String separation rate} = \frac{\textit{Number of characters that can be separated correctly}}{\textit{Total number of characters in strings}} \quad \ldots\ldots\ldots\ldots(47)$$

Table 2.7 Separation rates at each maximum separation candidate count in Multi-level network [for data set 1]

Maximum candidate count	2	3	4	5	6	7	8	9	10	20	30
Separation rates (%)	87.57	88.15	88.41	88.56	88.51	88.32	88.25	88.08	87.86	87.47	87.13

We set the maximum candidate count of Network A in the Multi-level network (Method 3) at 5 when the best string separation rate is obtained based on Table 2.7.

We set the maximum candidate count in Murase’s method (Method 2) at 5 as in Method 3. The Unified network is created from all candidates of Network A.

The string separation rates for each character count of a string in all four methods using hand written data set 2 (unknown data) are shown in Fig.2.14.

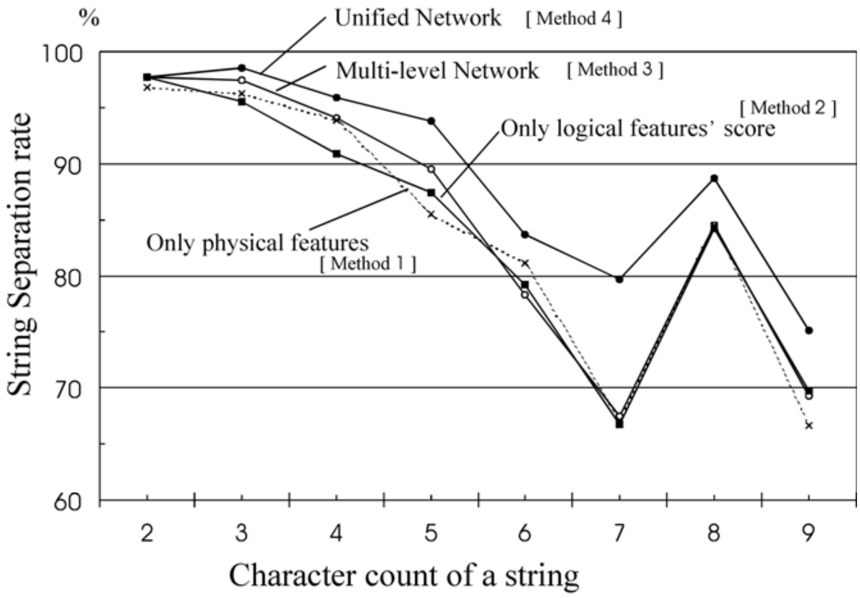


Fig.2.14 String separation rate for each character count of a string

When the character count of a string is 2, there is little difference between the string separation rates in all methods. However, when a character count of a string is much higher, there are greater differences between these rates. The reason why the string separation rates are too low when the character of a string is 7 and 9 is that there are a few strings in the data set and the quality of these strings is low.

The string separation rates of four methods are shown in Table 2.8.

Table 2.8 String separation rates (averages)

String separation rates [%]		Leaning Data (set 1)			Unkown Data (set 2)		
		KANJI	MIX	ALL	KANJI	MIX	ALL
Method 4	Unified Network	97.68	90.58	93.60	97.61	85.57	90.72
Method 3	Multi-level Network	96.94	82.33	88.56	96.42	79.51	86.73
Method 2	Only logical features' socre	95.02	81.83	87.45	94.31	79.00	85.54
Method 1	Only physical features	94.62	79.43	85.90	96.02	78.70	86.10

2.6.3 Estimation of experiment results

(a) Improved string separation rate

As shown in Table 2.8, the average separation rate of method 3 (Multi-level network) is better than that of method 2 (Murase’s method) and that of method 1 (only physical features) for all kinds of data; Kanji words and MIX (phrases that consist of kanji, hiragana, katakana, numeric, alphabetic and symbolic characters) in data sets 1 and 2. Furthermore, the separation rate of method 4 (Unified network) is better than that of method 3 (Multi-level network).

The separation rates of our methods (methods 3,4) are better than those of method 2 because the separation results in method 2 only use the score of the logical features from the separation candidates, but the results in our methods use the physical feature score and the logical feature score. Even if a method can judge that a separation position is a correct separator with physical features such as stroke interval, and the method does not use that information with the physical features, it will sometimes mistake a true separator as a non- separator.

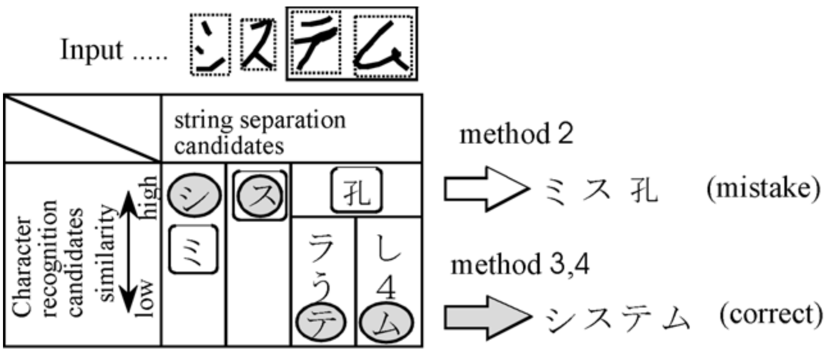


Fig.2.15 Combination of character recognition candidates

Using method 2 (Murase's method) for recognition of the handwritten character string "システム"(*shisutemu*) as shown in Fig.2.15, the language "ミス"(*misu*) and "孔"(*kou*) were extracted from the separation candidates and the character recognition candidates. Thus, the wrong result was sometimes obtained because the handwritten shape of "テ"(*te*) and"ム"(*mu*) were altered.

Methods 3 and 4 judge separators with not only the score of the logical features, but also the score of the character likeness and separator likeness on the basis of physical features. Therefore, even if they cannot judge separators using only logical features, they can find the separator correctly using physical features.

With our methods (methods 3 and 4), the correct result "システム"(*shisutemu*) is outputted. The character "孔"(*kou*) is not extracted with the understanding that the handwritten character pitch between "テ" (*te*) and "ム"(*mu*) is not too short. In addition, the width of this stroke group is too wide to be a character.

In Fig.2.14, the string separation rates of method 2 are worse than those of method 1 when a string character count is small (about 3 or 4) because there are many cases where language processing has a negative effect, as shown in Fig.2.15. This is because there are many incorrect candidates left which have too low a physical score when candidates are selected for Network B.

We think that the separation rate of method 4 (Unified network) was improved over that of method 3 (Multi-level network) for two reasons. First, separation results are obtained by combining both the physical features and logical features (character recognition results and language processing results) more effectively in the Unified network, where the weights of the nodes and links in Network B consider the base segment counts. Second, the Multi-level network limits the maximum string separation candidate count so the decision making process does not review every candidate.

(b) Improved character recognition rate

The character recognition rate is defined as the number of characters correctly recognized by the number of characters that were correctly separated. The character recognition rate was improved by language processing. For data set 2, the character recognition rates using only physical features were 73.76% for KANJI, 65.98% for MIX and 69.69% for ALL. Those with the Unified Network were 89.86% for KANJI, 85.57% for MIX, and 90.72% for ALL, but the number of characters correctly separated with this method were different from the number correctly separated using only physical features.

(c) Example of incorrect string separation

In Table 2.8, the separation rates are low due to the low quality string data, for example, overlapping character, large variations of character shapes, etc.

The separation result was sometimes not corrected using both physical features and logical features. When a string contains low quality characters, there often was no correct character in the character recognition candidates. As a result, the separation result was sometimes wrong. When an erroneous logical score was too high, even if the physical score was correct, the wrong separation candidate was sometimes selected. There were some cases when a character had very short strokes in its tail, for example [゛] in [ぶ](*bu*) or[ぐ](*gu*). These short strokes are incorrectly separated as a character such as [い]*(i)* and [り]*(ri)*. Since the character [ふ](*fu*) and [く](*ku*) are correct characters, [゛] can look like [い]*(i)* and [り] *(ri)*. Moreover, [ふい (に)](*fui(ni)*) and [くり](*kuri*) are correct phrases. However, [゛] is smaller than the surrounding character and is written on the top right-hand corner of a character. We think that a separation method using this information can prevent the short tail stroke [゛] from being incorrectly separated from the rest of the character.

In this experiment, the estimated value of physical features and logical features are added to the sum without the weight of each feature in the Multi-level network expression and the Unified network expression. When improving this method using the network expression, we think that it is important to use suitable weights of the estimated values for physical features and logical features and to correctly obtain the estimated value for physical and logical features.

(d) Processing time

The average processing times of all four methods for ALL (KANJI and MIX characters) are shown in Fig.2.16. They were measured in this experiment using a DOS/V PC (CPU: Pentium 166MHz: OS: Linux).

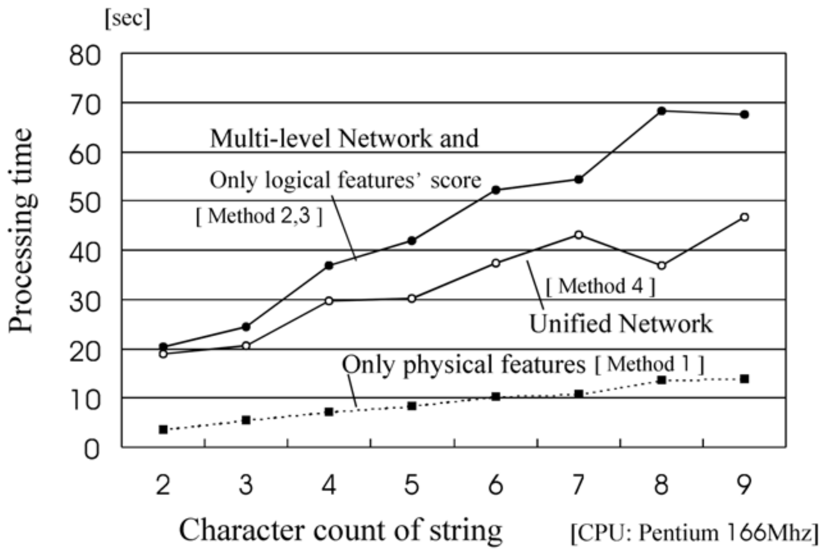


Fig.2.16 Comparison of processing speed

The processing times with method 2 (Multi-level network), 3 (Murase's method: using only logical features' score) shown in Fig.2.16 are the times when the maximum candidate count in Network A is 5. When that count is changed, the processing times with methods 2 and 3 are changed roughly in proportion to that count.

When a character count of a string is 5, the average processing times are 8.4 sec with method 1, 41.9 sec with methods 2 and 3 and 30.0 sec with method 4. If that count is not limited when a string character count is 5, the average candidate count is 46 and the processing time is 435.2 sec in methods 2 and 3. In methods 2 and 3, Network B is created, and Network A's candidate count times. Whenever one Network B is created, character recognition processes and language processes are executed and K-paths are searched. In method 4 (Unified network), even though the maximum separation candidate count is not limited, the processing time does not become enormous, as shown in Fig.2.16, because only one Unified network is created and only the shortest path is searched in the Unified network.

In method 4 (Unified network), when a string character count is 5, the detailed processing time is 0.04 sec by obtaining string separation candidate using physical features, 28.11 sec by characters recognition, 1.47 sec by language processing, and 0.45 sec by searching the shortest path.

2.7 Conclusion

In this paper, we propose on-line handwritten character string separation methods that unify physical features, character recognition, and language processing using network expressions.

We introduce two methods that use logical features such as character recognition results and language processing results as well as the score of physical features such as character pitch and stroke interval. The traditional Murase's method (Lattice method) uses only logical features after obtaining string separation candidates by physical features. Language processing estimates characters in the separation candidates as proper morphemes, words, phrases, and sentences

Our first method, using Multi-level network expressions, sums up the score of physical features in Network A and the score of logical features in Network B. Our second method, using Unified network expressions, unifies the score of physical features and the score of logical features using only one network.

The string separation rate could be improved by our methods for the unknown data set consisting of freely written Japanese strings from 21 different people. With the traditional Murase's method, the string separation rates were 94.31% for Japanese kanji words, 79.00% for MIX strings (phrases consist of Japanese kanji, hiragana, katakana, numeric, alphabetic and symbolic characters), and 85.54% for ALL strings (Kanji and MIX). With our Multi-level network expression, the string separation rates were 96.42% for kanji words, 79.51% for MIX strings, and 86.73% for ALL strings. With our Unified network expression, the string separation rates were 97.61 for kanji words, 85.57% for MIX strings, and 90.72% for ALL strings.

The reason the string separation rate was improved by our methods is because our methods obtain separation results using both physical features and logical features. The traditional Murase’s method is apt to select incorrect separation candidates that make morphemes, words, phrases, or sentences because the language processing often has a negative effect. Our methods seldom have negative influences from language processing because even if the incorrect separation candidates that make morphemes are chosen, the incorrect candidates are discarded when the physical feature score is low.

The rate of our second method (Unified network) is better than that of our first method (Multi-level network) is because the second method unifies the physical features, character recognition, and language processing more effectively than the first method. Another reason is that the first method limits the maximum string separation candidate count by physical features but the second method doesn’t limit it.

The processing time of our second method (Unified network) is shorter than that of the our first method (Multi-level network) and Murase’s method because our first method and Murase’s method create some logical networks(Network B) that include recognition process and search K shortest paths for all logical networks. However, our second method creates only one network, then searches the shortest path.

In the future, we will clarify how much weight should be given to each feature in the network expression. We think that the string separation rate can be further improved by better assigning weights to nodes and links in the network expression.

Chapter 3

Approaches to On-line Character Recognition

SUMMARY

We discuss structure analysis approaches and statistical approaches to on-line character recognition and then describe our approaches. We believe a statistical approach holds great promise because it has the advantage of its recognition dictionary being able to learn character features from many handwritten data without any manual work.

3.1 Introduction

When inputting characters with a pen, on-line character recognition technology is important. Many character recognition methods have been studied (7)(34). Most of these methods are based on stroke matching (35). Currently, if you neatly write Japanese characters, these characters are correctly recognized. However, it is difficult to recognize cursive-style-handwritten characters with stroke-number and stroke-order variations. Recently, some recognition methods for cursive-style-handwritten characters have been investigated. Typical methods are the Nakagawa Method with customizable recognition (36), which improves the recognition rate using pure online features, the Takahashi-Yasuda-Matsumoto Method (37) using a HMM model, the Wakahara Method using a stroke-based Affine transformation, and the Hamanaka-Yamada-Tsukumo Method (38)(39) with directional pattern matching(40-42) using off-line features.

Statistical approaches generally require faster processors and larger memory size than structure analytical approaches. However, since those hardware restrictions have been relaxed in these days, various character features can be used and more attention is being given to statistical approaches. We believe a statistical approach holds great promise because it has the advantage of its recognition dictionary being able to learn character features from many handwritten data without any manual work.

3.2 Structure analysis approaches

The base stroke matching method, which is well known in structure analysis approaches, is shown in Fig. 3.1. First, to extract the base stroke, features are obtained from each stroke. Conventional methods generally use, for example, positional relations between starting point, endpoint and bending point(s), degree of bending angle, and ratio of width and height. Next, these features are used to select the base strokes from the base stroke set, in which some base strokes (usually from 50 to 100) are pre-defined in a recognition dictionary by conditional equations. Then, the inputted character is recognized by comparing the base stroke codes of inputted characters with those of each character defined in the recognition dictionary. Some frequently occurring stroke orders are defined in this dictionary. When several recognition candidates have the same base stroke codes, one candidate is selected by fine recognition that uses the positional relations between strokes.

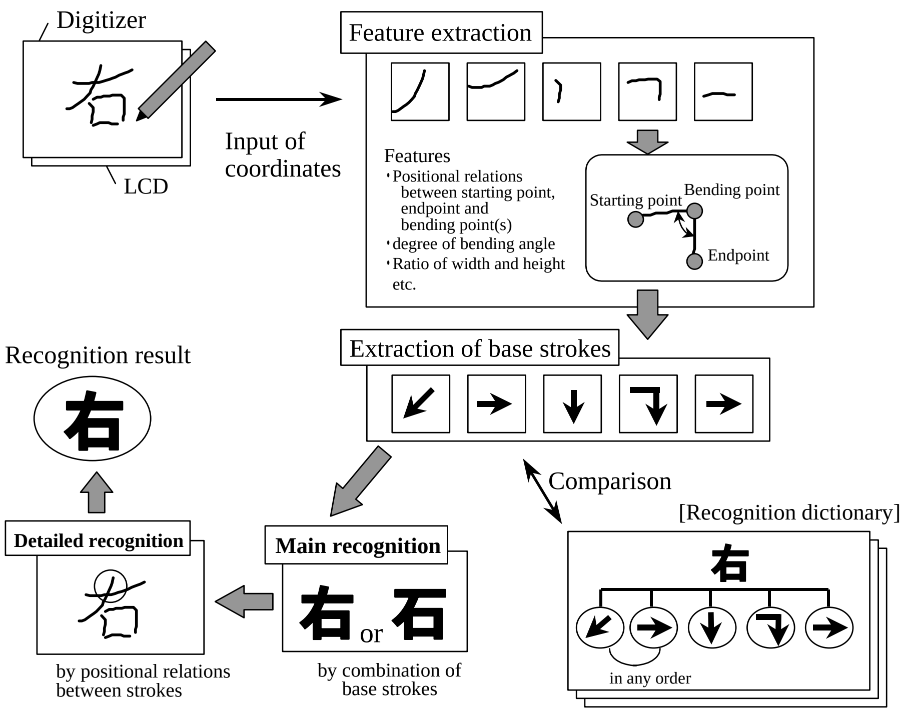


Fig. 3.1 Base stroke matching method

The base stroke matching method does not need much calculation and its software size is small. However, this method's dictionary has to be made manually.

3.3 Statistical approaches (pattern matching)

Statistical methods are usually used in OCRs. These methods recognize characters by pattern matching between the feature patterns of inputted characters and those of standard characters. One of the major methods is four-directional feature pattern matching. This method extracts feature patterns in each direction from an inputted character's bitmap data and compares them with the feature patterns of standard characters (Fig. 3.2). The character that has the highest degree of similarity is obtained as the recognition result. This method is described in detail in Section 4.2.

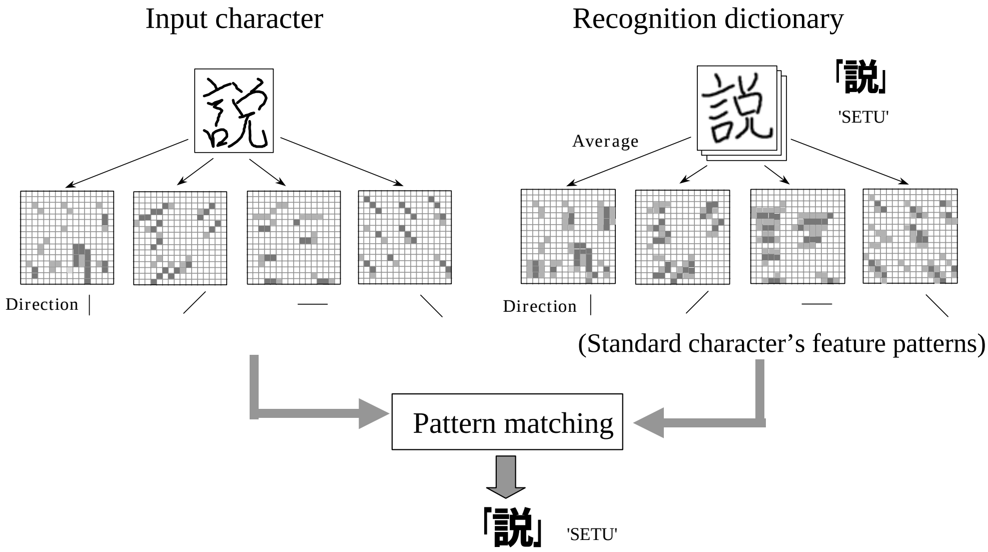


Fig. 3.2 Four-directional feature pattern matching method

In a statistical method, a recognition dictionary, consisting of standard character feature patterns, is created automatically from many handwritten (or printed) characters, (i.e., without manual work). Each character's feature pattern is created by averaging several characters having the same kind of feature patterns. Therefore, this method is highly advantageous for handling shape variations because it uses many data having shape variation to create the recognition dictionary. Therefore, we believe a statistical approach also holds great promise for future on-line character recognition.

3.4 Our approach (conclusion)

We conducted research on on-line character recognition based on the statistical approach. Consequently, we believe it is undesirable to use only off-line features, such as those used in OCRs, for recognizing free-stroke-order characters. Therefore, we actively used effective on-line features based on handwritten strokes in combination with off-line features. Our proposed method is described in Chapter 4.

Chapter 4

On-line Character Recognition Method using both Directional Features and Direction-Change Features

SUMMARY

We propose an on-line character recognition method that simultaneously uses both directional features, otherwise known as off-line features, and direction-change features, which we designed as on-line features. The directional features express the location and direction of each character's coordinates. The direction-change features express where and in which direction the character's written and unwritten imaginary stroke coordinates change, and the location of the circular parts of the character. We found suitable direction-change features with the imaginary strokes. The recognition rate was improved by our method, in comparison to the traditional method using only directional features.

4.1 Introduction

We have studied on-line character recognition methods based on statistical approaches. Among the statistical approaches to on-line character recognition, the Hamanaka-Yamada-Tsukumo Method is well known. This method first pre-classifies characters by stroke number as on-line features, then recognizes characters by the directional features of off-line features. Methods using directional features permit stroke shape variations of neatly written characters. However, freely written cursive characters are sometimes mistaken because, when characters are rapidly written, the shape widely varies. In addition, stroke connections are often created when the pen remains in the pen-down state. The directional features are steady in neatly written characters. However, in cursive characters, directional features are often unsteady.

Despite large shape variations and stroke connections, if we ignore whether the pen is in the up or down states, pen movement does not widely change, as shown in Fig.4.1. We think that human beings may remember pen-movements. Therefore, anybody can recognize cursive characters. So it is important to consider the pen-up state for cursive character recognition. In particular, we think that the direction-change features must be steady when the pen is up or down.

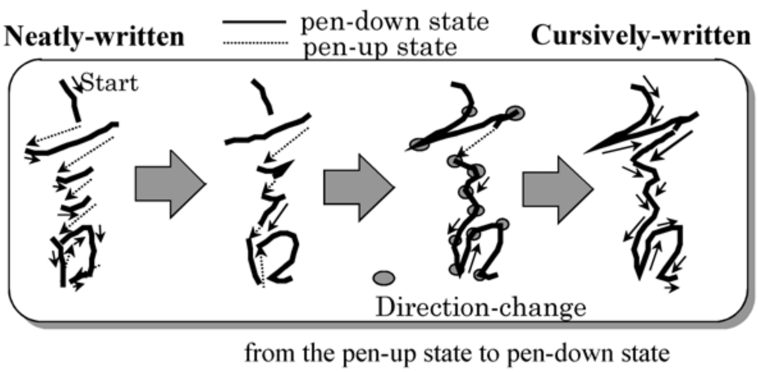


Fig.4.1 Pen movements

We propose a new handwritten character recognition method based on a statistical approach, called DDCPM (Directional and Direction-Change Pattern Matching), simultaneously using directional features (40-42) and direction-change features designed as on-line features. We think that directional features are static features, and direction-change features are dynamic features on the basis of the change in a hand's force when moving a pen. We think that directional features are suitable for detecting the consistent rough shape of cursive-style-handwritten characters, and direction-change features are suitable for detecting the consistent partial shape of these characters. By simultaneously using both of these features, our method takes a wide view of rough shape and partial shape.

4.2 Recognition Method

In on-line character recognition, there are many methods [(43)(44)] using only on-line features extracted from stroke data, and methods [(38)(39)] using mainly off-line features extracted from character bitmap data. Our method [(21)] uses on-line features that are direction-change features as well as off-line features that are directional features.

Fig.4.2 shows a block diagram of our character recognition method. First, on-line character data (x,y coordinate data) are transformed to bitmap data. Next, this bitmap data and on-line data are nonlinearly normalized. Directional features are then extracted from normalized bitmap data, and direction-change features are extracted from normalized on-line data. After these feature patterns are blurred, the patterns' dimensions are reduced. Afterwards, these reduced dimensional feature patterns of inputted characters are compared with reduced dimensional feature patterns of the standard characters, and the inputted characters are classified. Finally, original dimensional feature patterns of these characters are compared with original dimensional feature patterns of the standard characters by pattern matching, and inputted characters are recognized.

In traditional on-line character recognition methods, there are some structure analytical approaches [(35,45-48)] which use direction-change features as dynamic features. In these approaches, variations in stroke order degrade the recognition accuracy. On the other hand, our method, which is a statistical approach based on pattern matching, uses direction-change features by mapping the positions where the stroke direction changes onto 2-dimensional planes as described in Sec.4.2.6. Therefore, the degradation of the accuracy caused by variations in stroke order is less in our method.

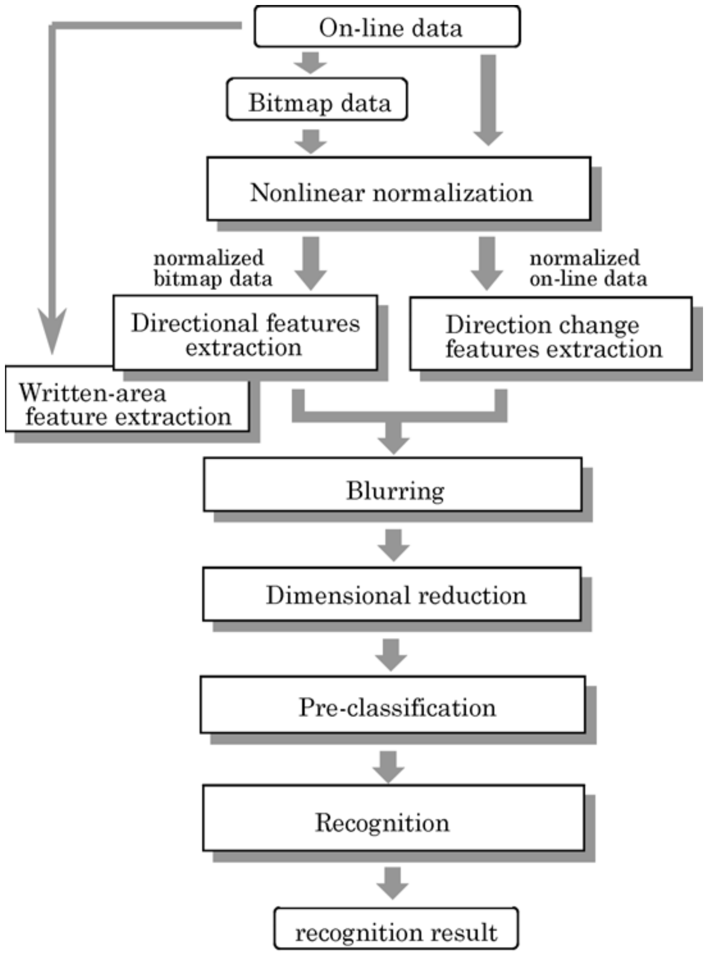


Fig. 4.2 DDCPM method diagram

4.2.1 Acquisition of on-line character data

On-line handwritten character data (x, y coordinate data) are obtained from tablets. Popular tablets used generally for PDAs and pen-type notebook PCs detect the coordinate data of pen movements by a pressure sensor or electromagnetic induction. Recently, there are some kinds of tablets that can detect coordinate data when the pen is either up or down, or with differing pressure levels and slope levels of the pen. However, popular tablets detect coordinate data only when the pen is down. Therefore, our character recognition method uses x, y coordinate data detected only when the pen is down. In the experiment shown in Sec.3, we use the on-line handwritten character data that are already inputted from some kinds of popular pen-based notebook PCs. On these PCs, the sampling rate is 50-100 points/second and the resolution is 3-10 points/mm.

4.2.2 Transformation from on-line data to bitmap data

When on-line data within a 64x64 pixel-size area is transformed to bitmap data (64x64 pixels), lines are drawn 3 pixels thick between each coordinate and its neighboring coordinate, then each coordinate point on the line is turned black in the bitmap. Thickening lines reduces local fluctuations of the directional features. The width 3 is determined empirically.

4.2.3 Nonlinear normalization

The transformed bitmap data is nonlinearly normalized by Line Density Equalization [49][50]. Fig.4.3(a) shows an example of nonlinear normalization. We use 1.0 for the α parameter which expresses a degree of normalization in Line Density Equalization [49]. Line Density Equalization normalizes the shape of characters so that the line density in both x and y directions are uniform. Incidental deformations of characters are reduced by this operation. The on-line data is also nonlinearly normalized by the same function used for the transformed bitmap. By nonlinear normalization of the on-line data, (x,y) coordinates of the character's strokes are changed while keeping the order of the coordinates intact as shown in Fig.4.3(b).

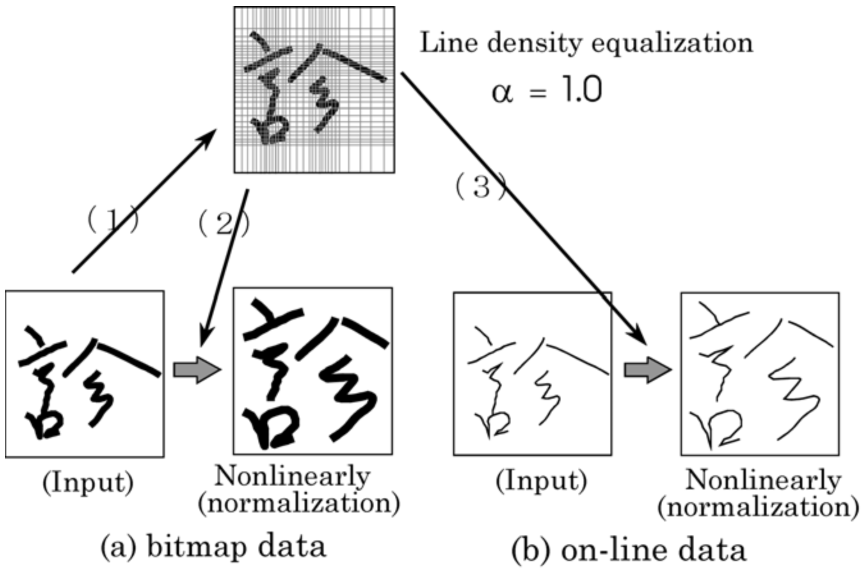


Fig.4.3 Nonlinear normalization

4.2.4 Directional features

After contour extraction of the bitmap data, 4 directions: vertical(|), right-up(/), horizontal(-), and left-up(＼), are detected from the contour line between each position and the point located after the next position in a clockwise direction. At this time, the vectors' positions with directions are moved from the contours to positions that are half of the average stroke's width away from the contours towards the centers of the strokes.

Four 16 x 16 meshes are created, each one representing a different direction pattern. The number of vectors in each mesh is then counted as shown in Fig.4.4, where the density of each mesh shows the number of vectors in each mesh [41-42].

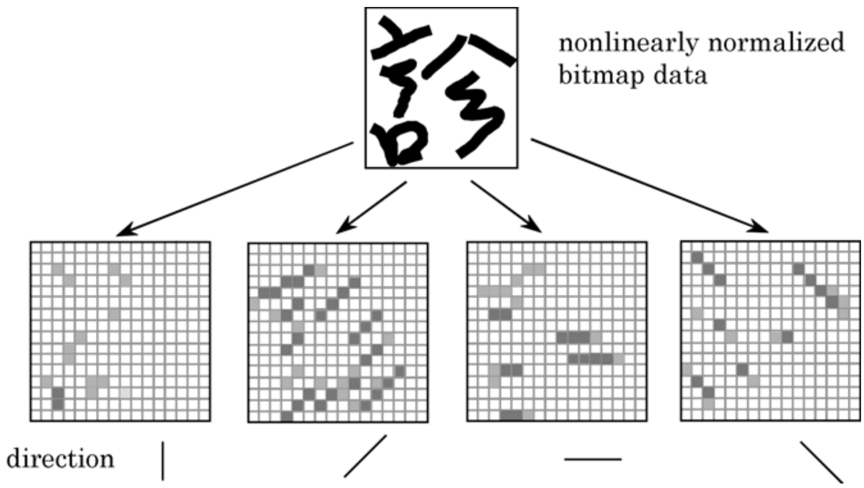


Fig.4.4 Directional patterns (16 x16 x 4)

4.2.5 Written-area feature

The written-area where the character is written is obtained as a feature. In Japanese characters, there are some similar characters that the size/position of them are different while the shapes of them are same, as three characters shown of the right in Fig.4.5. The written-area feature is useful for distinguishing such characters.

The written-area is expanded to a mesh between a 40 x 40 and 64x64 pixels by adding space around the area. This area is then compressed to 1/4 of its original size. The final mesh size is between 10x10 and 16x16 pixels. Examples of written-area features are shown in Fig.4.5.

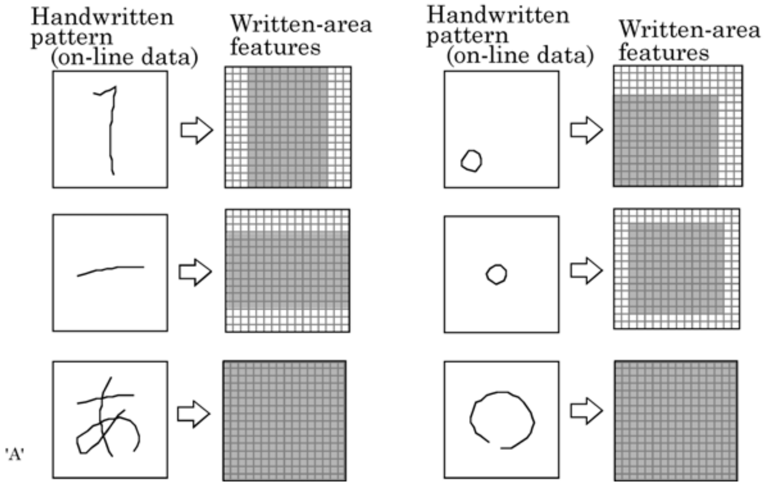


Fig.4.5 Examples of written-area features

4.2.6 Direction-change features

4.2.6.1 Direction-change features in the pen-down state

The direction-change positions, the direction-change degree and the directions after direction-change are obtained from the normalized on-line data as shown in Fig.4.6. First, each stroke coordinate's direction-change degree, that is the absolute value of the difference in direction from the target coordinate to the next coordinate, and the former coordinate's direction toward the target position, is calculated.

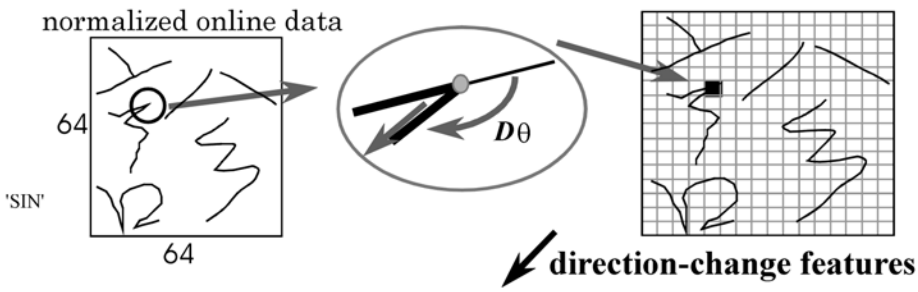


Fig.4.6 Extraction of Direction-change features in the pen-down state

The direction differences are expressed in 60 degree increments. The direction-change feature's degree (Fdc) is shown in Eq.4.1.

$$Fdc = \frac{|D\theta|}{60} + 1 \ldots\ldots(4.1)$$

$D\theta$: *direction difference in* 60 *degreeincrements*

$(-180 \leq D\theta \leq 180)$

The directions after direction-change are expressed by 8 kinds of direction in 45 degree increments. Next, by summing the direction difference degrees in the 16x16 meshes, every direction-change feature is mapped onto 8 of 9 separate 16x16 meshes as shown in Fig.4.8.

The circle-feature, the second of the direction-change features, is then mapped onto the 9th mesh. The circle parts of a character are found by searching a segment on which the stroke direction continuously changes in the same rotational orientation and comparing the distance between the starting point and the endpoint of the segment with a threshold. Since this threshold is calculated from the size of the rectangle which circumscribes the segment, it is also possible to find an incomplete circle.

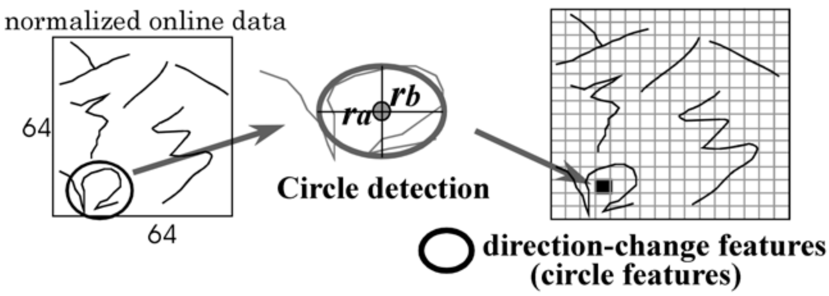


Fig.4.7 Extraction of circle features in the pen-down state

The degree of a circle-feature Fc(xc,yc) at the circle's center (xc,yc) is expressed by the circle's radius (rc) as shown in Eq. 4.2 below.

$$Fc(xc, yc) = Min\left(\frac{maximum\ radius}{rc},\ maximum\ Fc\right) \ldots(4.2)$$

$rc = (ra + rb)/2$

ra : *long radius of an ellipse,*

rb : *short radius of an ellipse*

maximum radius : 32

maximum Fc : 8

The circle-feature pattern is generated and added to 8 kinds of direction-change patterns as shown in Fig.4.8. The circle feature is useful for recognizing similar characters, especially in HIRAGANA, such as “ る / ろ “, “ ね / わ “, “ ぬ / め “, “ ぺ / べ “, etc. In these pairs, the shapes of characters are almost same except a loop exists or not.

In Fig.4.6, 4.7 and 4.8, each character pattern clearly explains the direction-change position, but does not express features.

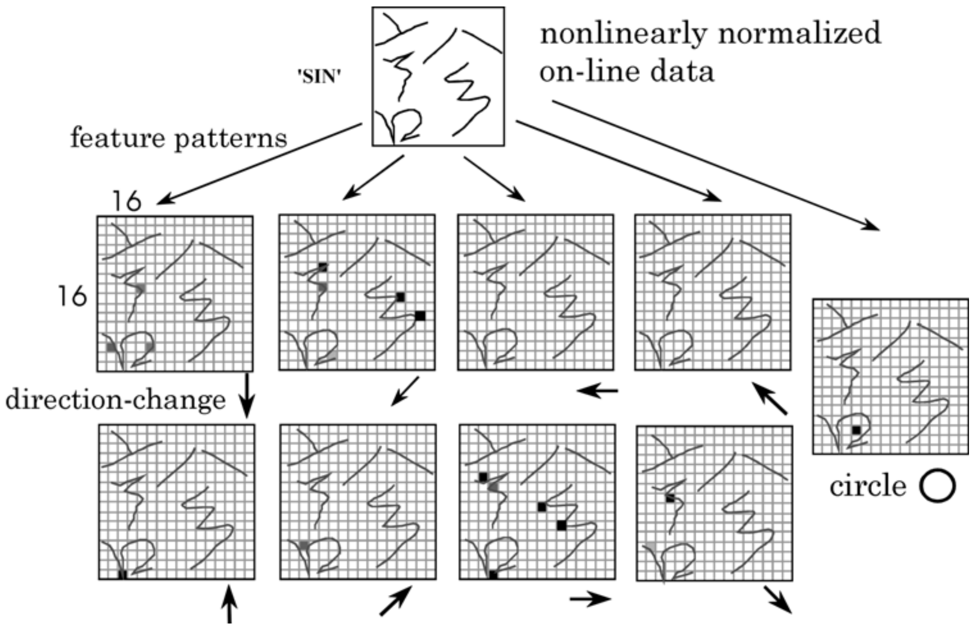


Fig.4.8 Direction-change features’ patterns (16 x 16 x 9) in the pen-down state

4.2.6.2 Direction-change features in the pen-down and pen-up state

When characters are written cursively, strokes are often connected with the next stroke. The faster characters are written, the more strokes are connected in the pen-down state. The closer the distance between a stroke and the next stroke, the more often these strokes are connected.

To recognize cursively-written characters with connected strokes and neatly-written characters without connected strokes, without increasing standard character patterns, our method extracts the features from the imaginary strokes in the pen-up state and the written strokes as shown in Fig.4.9.

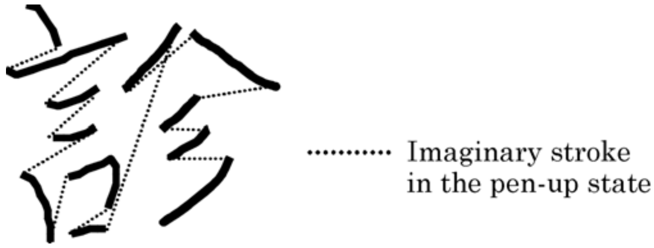


Fig.4.9 Imaginary strokes

Our method extracts the direction-change features with imaginary strokes which are lines between each stroke's end position and the next stroke's start position as shown in Fig.4.10.

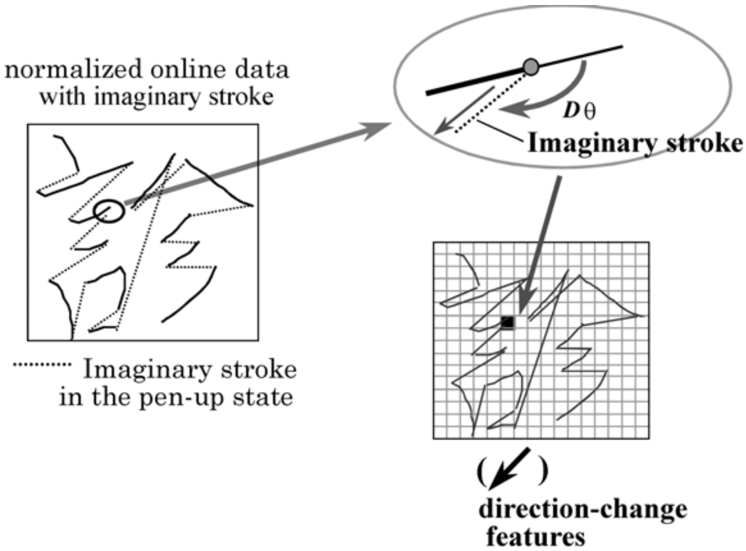


Fig.4.10 Extraction of Direction-change features with imaginary strokes

The direction-change feature's degree (Fdc’) of the connected imaginary stroke is shown in Eq.4.3. The shorter the length of the imaginary stroke, the stronger the feature is expressed.

$$Fdc = \left(\frac{|D\theta|}{60} + 1 \right) \times Min\left(\frac{maximum\ length}{Limag} \times weight,\ 0.5 \right) \qquad (4.3)$$

$D\theta$: *direction difference in 60 degree increments*

$Limag$: *length of imaginary stroke* (≥ 1)

$maximum\ length$: $64\sqrt{2}$, $weight$: $1/8$

By summing the direction-change features of the character with imaginary strokes and written strokes in 16x16 meshes, 8 kinds of direction-change patterns and circle-feature pattern are generated as shown in Fig.4.11.

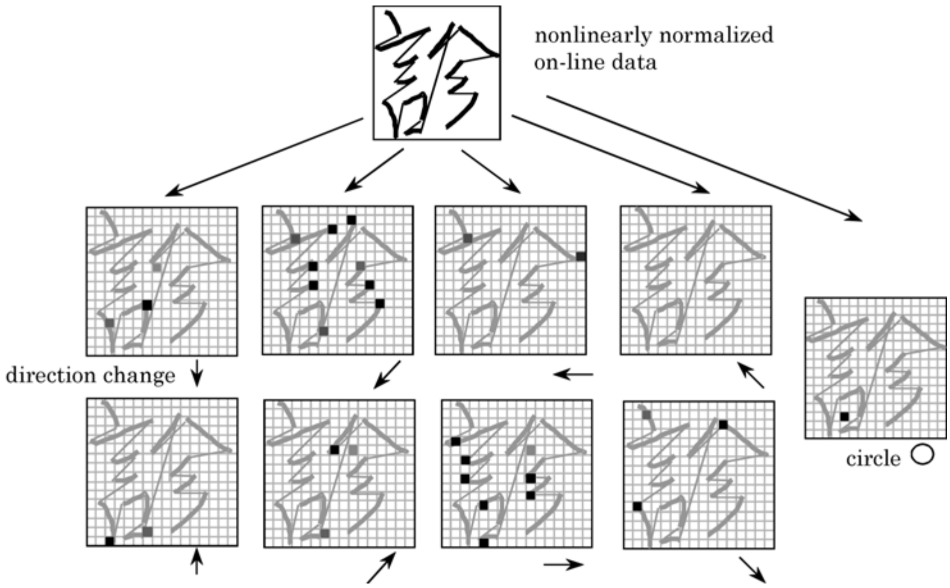


Fig.4.11 Direction-change features' patterns (16 x 16 x 9)
in the pen-down and pen-up states

4.2.7 Blurring

The directional and direction-change patterns in the 16x16 meshes are set in the 24x24 meshes. Each feature pattern in the 24x24 meshes is expressed as $f(x,y,\nu)$ [x,y=1~24] [ν=1~4 (4 kinds of directional pattern), 5(1 kind of written-area pattern), 6~14 (9 kinds of direction-change pattern)].

Each feature pattern $f(x,y,\nu)$ is blurred by a Gaussian function, generating each feature pattern $f^1(x,y,\nu)$.

4.2.8 Pre-classification

4.2.8.1 Dimensional reduction

For faster classification, dimensions are reduced and candidates are narrowed down before the recognition step.

The dimension number of each blurred feature pattern $f^1(x,y,\nu)$ is reduced to 4x4 dimensions, then the feature pattern $g^1(x,y,\nu)$ is created as shown in Eq.4.4.

$$g(i,j,\nu)=\sum_{p,q=1}^{4} f^{1}(4i+p,4j+q,\nu)................(4.4)$$

As described below, this feature pattern $g(i,j,\nu)$ is used in the candidates selection step, and the feature pattern $f^{1}(x,y,\nu)$ before dimensional reduction is used in the recognition step, respectively.

4.2.8.2 Candidates selection

By comparing the inputted character's reduced dimensional feature patterns $g(i,j,\nu)$ and the standard characters' reduced dimensional feature patterns $\tilde{g}_c(i,j,\nu)$,which were already created in the character classification dictionary as shown in Fig.4.12, the resemblance (r_{gc}) between $g(i,j,\nu)$ and $\tilde{g}_c(i,j,\nu)$ is calculated on the basis of Eq.4.5. Then, using the resemblance, the possible candidates are reduced to 100 characters.

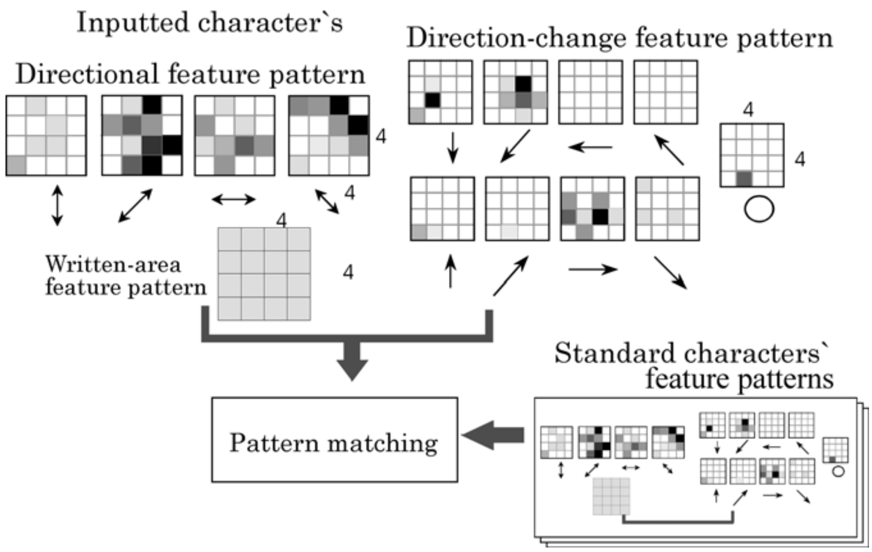


Fig.4.12 Pre-classification

$$r_{gc}=\frac{\sum_{\nu=1}^{14}\sum_{i,j=1}^{4} g(i,j,\nu)\cdot\tilde{g}_c(i,j,\nu)}{\sqrt{\sum_{\nu=1}^{14}\sum_{i,j=1}^{4} g(i,j,\nu)^2}\cdot\sqrt{\sum_{\nu=1}^{14}\sum_{i,j=1}^{4}\tilde{g}_c(i,j,\nu)^2}}........(4.5)$$

4.2.9 Recognition

Within the character candidates, by comparing the inputted character's original feature pattern $f^1(x,y,\nu)$ and the standard characters' original feature patterns $\tilde{f}_c^{\,1}(x,y,\nu)$, which are already created in the character recognition dictionary, the resemblance (r'_{fc}) between $f^1(x,y,\nu)$ and $\tilde{f}_c^{\,1}(x,y,\nu)$ is calculated using Eq.4.6. The character whose resemblance is highest is obtained as the character recognition result.

$$r_{fc} = \frac{\sum_{\nu=1}^{14}\sum_{x,y=1}^{16} f^1(x,y,\nu)\cdot\tilde{f}_c^{\,1}(x,y,\nu)}{\sqrt{\sum_{\nu=1}^{14}\sum_{x,y=1}^{16} f^1(x,y,\nu)^2}\cdot\sqrt{\sum_{\nu=1}^{14}\sum_{x,y=1}^{16}\tilde{f}_c^{\,1}(x,y,\nu)^2}} \quad ....(4.6)$$

4.3 Experiment

We experimented with character recognition using our recognition method and the on-line Japanese handwritten data base (TUAT Nakagawa Lab. HANDS-kuchibue_d-96-02) [51][22], containing handwritten data from 81 people, where each person's data contains 11,962 character samples. We use the samples in even-numbered sets as learning data, and use the samples in odd-numbered sets as unknown data. Examples of not-neatly-written data from this data base are shown in Fig.4.13.

Fig.4.13 Examples of database (HANDS-kuchibue_d-96-02)

For the character size to be within 64x64 pixels, we change the coordinates of each sample in the experiment.

(a) Experiments comparing the traditional method and our 1st and 2nd methods

The recognition rates of the method using only directional features and written-area feature are shown in Table 4.1. The recognition rates of our 1st method using both directional features and direction-change features in only the pen-down state are shown in Table 4.2. The recognition rates of our 2nd method using both directional features and direction-change features in both the pen-down and pen-up states are shown in Table 4.3. In Table 4.1-3, non-KANJI means all Japanese characters except KANJI characters, for example, HIRAGANA, KATAKANA, numeric, alphabetic and symbolic characters.

Table 4.1 Recognition rates of the method using only directional features (traditional method)

Recognition Rate	Unknown Data		
	KANJI	Non-KANJI	All
1st candidate	82.58 %	73.71 %	77.89 %
2nd candidate	89.60 %	80.92 %	85.01 %
3rd candidate	92.28 %	85.25 %	88.57 %
4th candidate	93.67 %	87.72 %	90.52 %
5th candidate	94.58 %	89.30 %	91.79 %
10th candidate	96.68 %	92.72 %	94.59 %
20th candidate	97.90 %	94.99 %	96.36 %
100th candidate	99.02 %	98.44 %	98.71 %

Table 4.2 Recognition rates of the method using the directional features and direction-change features in only the pen-down state (our 1st method)

Recognition Rate	Unknown Data		
	KANJI	Non-KANJI	All
1st candidate	84.71 %	76.90 %	80.59 %
2nd candidate	91.48 %	83.72 %	87.38 %
3rd candidate	93.92 %	88.00 %	90.79 %
4th candidate	95.15 %	90.27 %	92.58 %
5th candidate	95.95 %	91.76 %	93.73 %
10th candidate	97.63 %	94.71 %	96.09 %
20th candidate	98.62 %	96.37 %	97.43 %
100th candidate	99.51 %	98.95 %	99.22 %

Table 4.3 Recognition rates of the method using the directional features and direction-change features in the pen-down and pen-up state (our 2nd method)

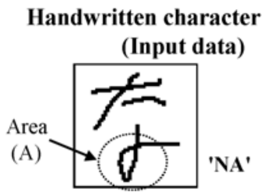
Recognition Rate	Unknown Data		
	KANJI	Non-KANJI	All
1st candidate	87.97 %	77.37 %	82.37 %
2nd candidate	93.44 %	84.06 %	88.49 %
3rd candidate	95.34 %	88.24 %	91.59 %
4th candidate	96.31 %	90.47 %	93.23 %
5th candidate	96.90 %	91.95 %	94.28 %
10th candidate	98.21 %	94.87 %	96.44 %
20th candidate	98.94 %	96.53 %	97.67 %
100th candidate	99.62 %	99.00 %	99.29 %

From these experiments, we found that our methods using both directional features and direction-change features were able to obtain higher recognition rates than the traditional method using only directional features. Moreover, we found that the recognition rate was further improved using direction-change features including imaginary strokes in the pen-up state. Our 1st method improves the recognition rate for all characters by 2.70 % as compared with the traditional method. Using our 2nd method, the recognition rate is improved by 1.78 % over our 1st method, and 4.48% as compared with the traditional method.

Note that although the number of dimensions in our 1st method is equal to the number of dimensions in our 2nd method, the 2nd method has a higher recognition rate.

In the odd data set, the over all recognition rate for the 3rd person's neatly written data was 93.33%; 95.57% for KANJI characters and 91.33% for non-KANJI characters.

Fig.4.14 shows a good example of the candidates obtained by our method for the character "Na".



Recognition result

Method	Recognition candidates 1	2	3	4	5	6	7	8	9	10
Traditional method	奔	委	蚕	卒	奈	字	奉	奏	黍	泰
Our 1st method	な	ぼ	ほ	ば	奔	ま	委	ぽ	蚕	は
Our 2nd method	な	ぼ	ほ	ば	蚕	奔	ま	奈	奏	泰

Fig.4.14 Example of the recognition result

Using only directional features, the traditional method did not recognize the input character “NA” properly. However, our 1st and 2nd methods could recognize this character correctly because the circle pattern is extracted in area (A) in Fig.4.14. In our methods, some recognition candidates which have a circle pattern in a low position are given a high rank. The reason for this is explained in detail below.

The directional features of the input character “NA”, the standard directional features of the correct character “NA”, and the mistaken character “HON” obtained by the traditional method are shown in Fig.4.15. As shown in Fig.4.15, the inputted character’s directional features look more like the mistaken character’s features than the correct character’s features. So, this input character “NA” is recognized incorrectly as the mistaken character “HON”.

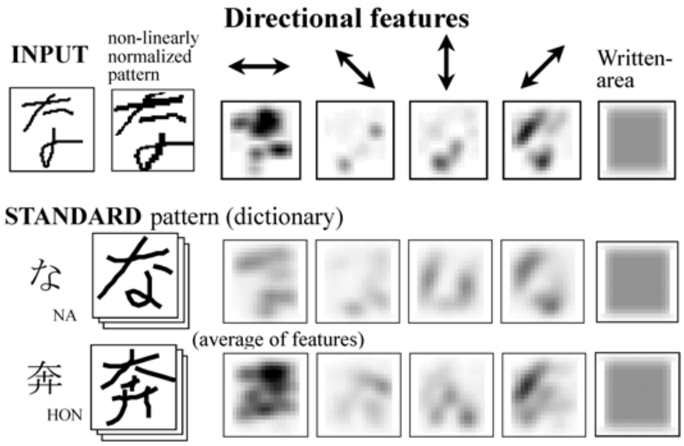


Fig.4.15 Example of directional features

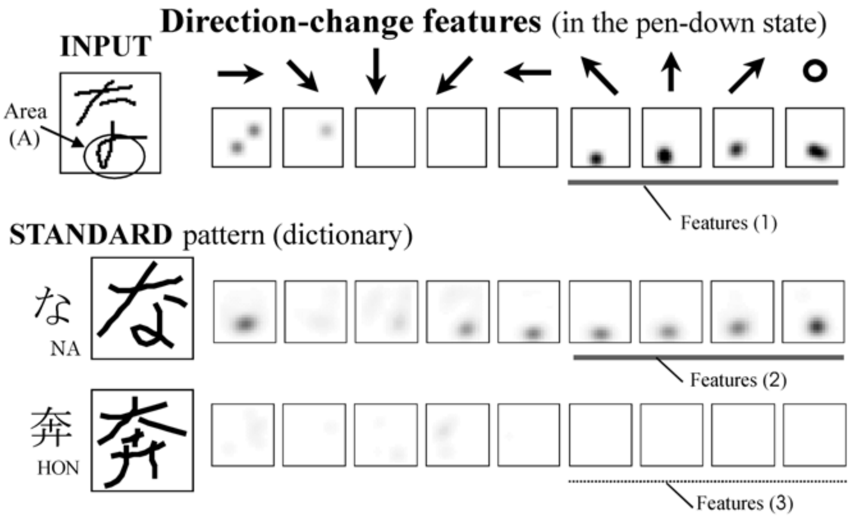


Fig.4.16 Example of direction-change features

The direction-change features in the pen-down state are shown in Fig.4.16. In the inputted character, the direction-change features (1) are extracted from the area (A). Similar features (2) are also extracted from the correct character “NA”. However, these kinds of features are not extracted from the character “HON”. So, This input character “NA” is recognized correctly by our method.

Fig.4.17 shows examples where the recognition results are improved by our 1st method using the direction-change features in the pen-down state only. The improvements are due to the circle-features that are extracted from areas (B)(C)(D) and the direction change features that are extracted in positions(E)(F) in this figure.

When only the circle feature isn’t used, the recognition rates are 84.77% for KANJI characters, 76.70% for non-KANJI characters, 80.51% for all characters. This result shows circle feature is effective especially for non-KANJI characters.

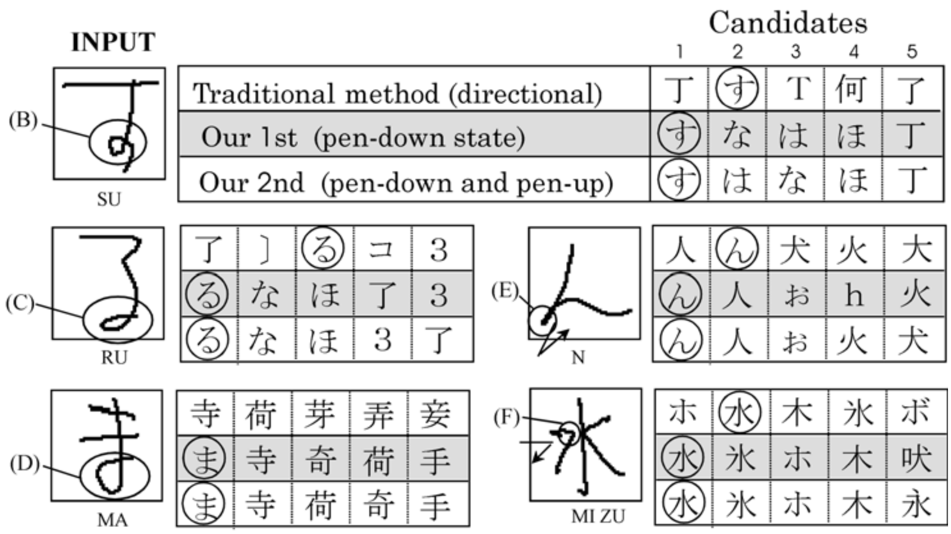


Fig.4.17 Examples of Improvements by our 1st method

As shown in Fig.4.16-17, our 1st method using simultaneously both the direction-change features and the directional features permits shape variations caused by writing quickly.

Negative influences of our first method were sometimes caused when an inputted character’s stroke count varied. In Fig.4.18, the numbers near the strokes of the inputted handwritten characters express stroke order. The character “BUN” is usually written as shown in the figure of the standard character “BUN” where the 2nd and 3rd strokes are written separately. However, there is one connected stroke in the inputted character “BUN”. Although the inputted character “BUN” is correctly recognized by the traditional method using only directional features, this inputted character was incorrectly recognized by our first method. This is because the direction-change feature is extracted from the inputted character at the second stroke’s bending point, but the direction-change feature is not extracted from the standard character at the same position of the endpoint of the second stroke. However, our second method can correctly recognize such characters as those shown in Fig. 4.18 because the direction-change features are extracted from not only the inputted character at the second stroke’s bending point but also from the standard character at the second stroke’s endpoint by using imaginary strokes.

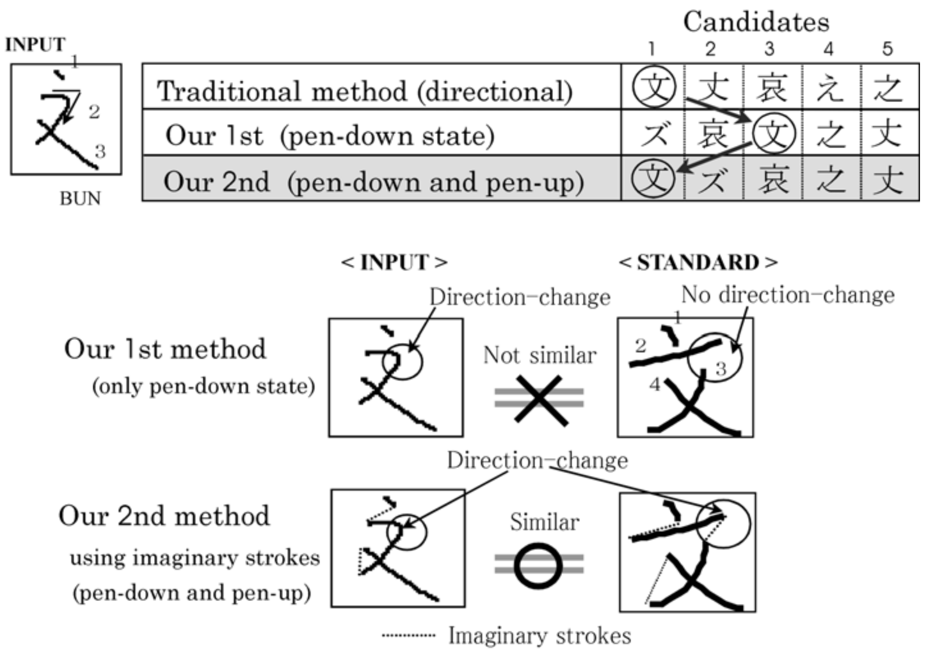


Fig.4.18 Example of negative influence of our 1st method and improvement by our 2nd method

Fig.4.19 also shows an example where the recognition result is improved by our 2nd method using the direction-change features in both the pen-down and pen-up states. The character “SYOU” is usually written as shown in the figure of the standard character “SYOU” where the 2nd and 3rd strokes are written separately. However, there is one connected stroke in the inputted character “SYOU”. So, the horizontal directional features of the inputted character “SYOU” look more like those of the standard character “NA” than those of the correct standard character “SYOU” as shown at S1 in Fig.4.19. Furthermore the direction-change features of these character are in the pen-down state only. Therefore, both the traditional method and our 1st method can not recognize this character correctly.

However, direction-change features in both the pen-down and pen-up states of the inputted character are similar to standard features of the correct character as shown at S2,S3 in Fig.4.19. So, the inputted character “SYOU” is recognized correctly by our 2nd method.

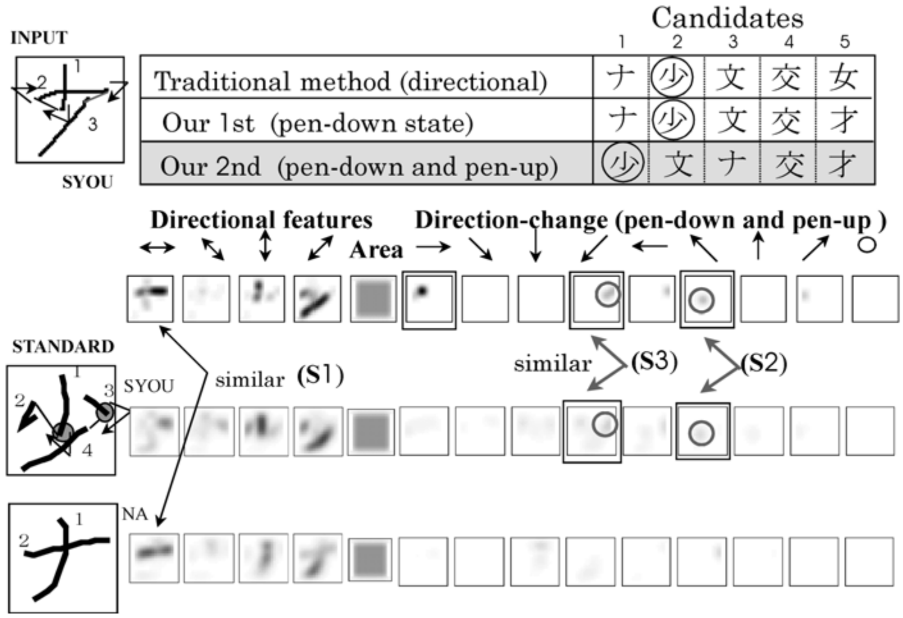


Fig.4.19 Example of improvement by our 2nd method

Fig.4.20 shows other examples where the recognition results are improved by our 2nd method. On the character “MATU” in Fig.4.20, because the 5th stroke is written much close to the 4th stroke, this character is recognized incorrectly by the traditional method and our 1st method. However, because there is a direction-change at the position (A) where the 4th stroke is connected with the imaginary between this stroke and 5th stroke, this character is recognized correctly. In the same way, because there are direction-changes at the positions (B) (C) in the character “MAI”, (D) in the character “NA”, (E) (F) in the character “SETU” and (G) in the character “SU”, these character are recognized correctly.

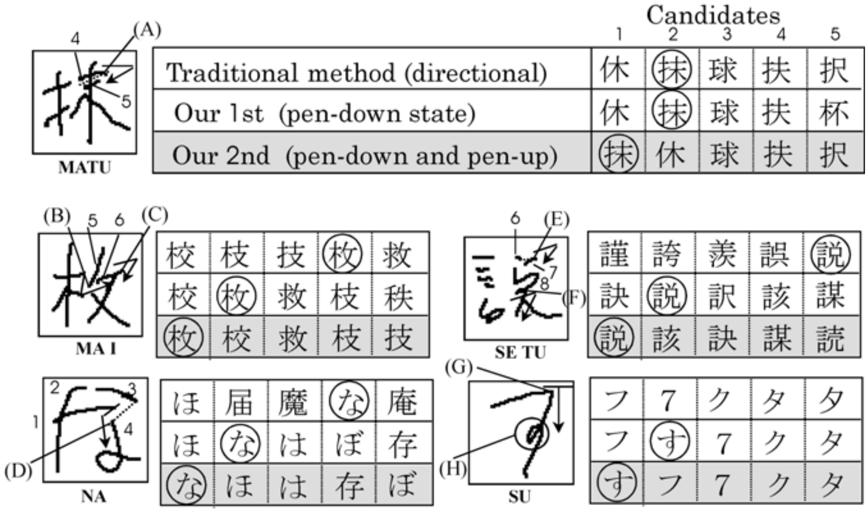


Fig.4.20 Examples of Improvements by our 2nd method

As shown in Figs. 18-20, our second method, which extracts the direction-change features in the pen-down and pen-up states by using imaginary strokes, is effective in handling stroke count variations due to stroke connections.

(b) 3rd method experiment

In our 3rd method, the direction-change feature's degree (Fdc') of the imaginary stroke is shown in Eq.4.7, where W(Limag) is the weight function whose input is the length of imaginary stroke Limag.

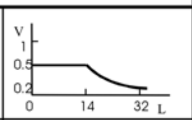
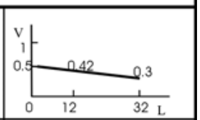
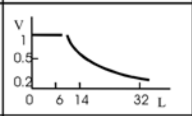
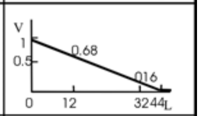
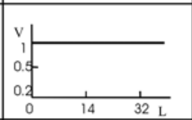
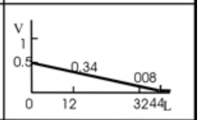
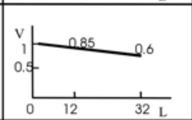
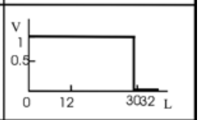
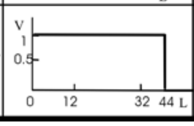
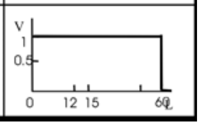
$$Fdc' = \left(\frac{|D\theta|}{60} + 1 \right) \times W\left(Limag\right) \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.7)$$

$D\theta$: *direction difference in* 60 *degree increments*

$Limag$: *length of imaginary stroke* (≥ 1)

We tried to examine the influences on character recognition rates when changing the functions used to get each direction-change feature based on the imaginary stroke lengths. We set some W(Limag) weight functions, as shown in Eq.4.7. These functions are shown in Table 4.4; where L shows Limag (imaginary stroke lengths).

Table 4.4 W functions for Direction-change features

No	Function		No	Function	
0	$W=\frac{60}{L+1}\times\frac{1}{8}$ when W>0.5 : W=0.5		5	$W=\frac{64\times\sqrt{2}-(L+1)}{64\times\sqrt{2}}\times\frac{1}{2}$	
1	$W=\frac{60}{L+1}\times\frac{1}{8}$ when W>1.0 : W=1.0		6	$W=\frac{64\times\sqrt{2}-(L\times 2)}{64\times\sqrt{2}}$ when $L<\frac{64\times\sqrt{2}}{2}$; W=0 when others	
2	W= 1		7	$W=\frac{64\times\sqrt{2}-(L\times 2)}{64\times\sqrt{2}}\times\frac{1}{2}$ when $L<\frac{64\times\sqrt{2}}{2}$; W=0 when others	
3	$W=\frac{64\times\sqrt{2}-(L+1)}{64\times\sqrt{2}}$		8	W= 1 when $L<64\times\sqrt{2}\times\frac{1}{3}$; W=0 when others	
4	W= 1 when $(L+1)\leqq\frac{64\times\sqrt{2}}{2}$; W= 0 when others		9	W= 1 when $L<64\times\sqrt{2}\times\frac{2}{3}$; W=0 when others	

The closer the distance between a stroke and the next stroke, the more often these strokes are connected. So, We hypothesized that function no.0 would be the most suitable function for our 2nd method. In this function, the shorter the length of the imaginary stroke, the stronger the feature is expressed.

First, we tried to examine the influences to character recognition rates when changing W(Limag) with 10 people's data (top 10 sets of even-numbered sets). Each recognition rate using W(Limag) function is shown in Table 4.5.

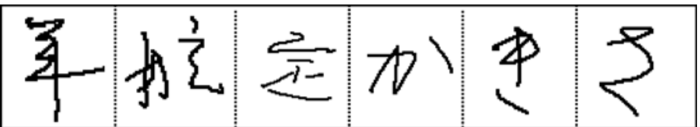
Table 4.5 Recognition rates using each W(Limag) function

Function	KANJI	non-KANJI	ALL
Fnc.2 (New)	92.60	79.37	85.61
Fnc.9	92.60	79.36	85.61
Fnc.4	92.43	79.18	85.44
Fnc.8	91.92	77.87	84.50
Fnc.3	91.61	76.83	83.81
Fnc.6	90.62	75.53	82.65
Fnc.1	89.98	75.11	82.12
Fnc.0 (Old)	88.26	74.41	80.94
Fnc.5	87.10	74.17	80.27
Fnc.7	86.36	74.01	79.84

[even-numbered 10 data sets]

We were surprised at the high recognition rate of function no.2, which has no weight on the imaginary stroke lengths. The reason for this is that in the handwritten data base sets, there are many cursively-written characters where strokes are connected even though the strokes are far apart. Some examples of characters which are recognized correctly when using function no.2, but are not recognized when using function no.0, are shown in Table 4.6.

Table 4.6 Examples of recognition results using our new method (Function no.2)

Handwritten character						
Recognition result	年 (NEN)	舷 (GEN)	定 (TEI)	か (KA)	き (KI)	さ (SA)

The recognition rates of our 3rd method with 41 people's data sets is shown in Table 4.7.

Table 4.7 Recognition rates of the method using the function no.2 (our 3rd method)

Recognition Rate	Unknown Data		
	KANJI	Non-KANJI	All
1st candidate	91.41 %	81.77 %	86.32 %
2nd candidate	95.37 %	87.62 %	91.27 %
3rd candidate	96.75 %	91.26 %	93.85 %
4th candidate	97.39 %	93.18 %	95.16 %
5th candidate	97.84 %	94.45 %	96.05 %
10th candidate	98.78 %	96.75 %	97.71 %
20th candidate	99.30 %	98.04 %	98.63 %
100th candidate	99.74 %	99.53 %	99.63 %

[41 people`s data set]

The recognition rates of the traditional method and our three methods on 41 people's data sets (odd-numbered sets) are shown in Table 4.8. From these experiments, we found that the recognition rate was further improved by using a suitable function to get direction-change features using imaginary strokes.

Table 4.8 Recognition results

Recognition Rate	Unknown Data		
	KANJI	Non-KANJI	All
Traditional method	82.58 %	73.71 %	77.89 %
Our 1st method	84.71 %	76.90 %	80.59 %
Our 2nd method	87.97 %	77.37 %	82.37 %
Our 3rd method	91.41 %	81.77 %	86.32 %

[41 people`s data set]

By the above analysis, we confirmed that our methods permit stroke shape variations caused by writing quickly, stroke order variations and stroke count variation due to stroke connections.

The recognition processing times of the traditional method and our three methods are 0.9 sec, 2.4 sec respectively [CPU: Pentium,133MHz].

4.4 Conclusion

In this paper, we propose a new on-line recognition method simultaneously using both directional features, otherwise known as off-line features, and direction-change features which are designed as on-line features simultaneously, to correctly recognize handwritten cursive-style Japanese characters.

From the results of experiments using an on-line Japanese handwritten data base (HANDS-kuchibue_d-96-02), we found that our methods using both directional features and direction-change features were able to obtain higher recognition rates than the traditional method using only directional features, The recognition rate of the traditional method is 77.89%. The recognition rate of our first method using direction-change features in only pen-down state only is 80.59%. The recognition rate of our 2nd method using direction-change features in both the pen-down and pen-up states is 82.37%. The recognition rate of our 3rd method using a function which has no weight (always 1) is 86.32%. We found that the recognition rate was further improved with the direction-change features taking into account imaginary strokes in the pen-up state. We found that the most suitable function to get direction-change features is the function which has no weight (always 1), and has no effect on the length of the imaginary strokes in the pen-up state. This is because when characters are written freely, strokes are often connected even though strokes are far apart.

As a result, we confirmed that our methods permit stroke shape variations caused by writing quickly, stroke order variations, and stroke count variation due to stroke connections.

Essentially, the directional features are not affected by stroke order variations and stroke count variations because these features are extracted from bitmap data of on-line written character data. However these features are greatly affected by the instability of the stroke directions caused by stroke shape variations. The direction-change features are steady for stroke shape variations. Moreover, the direction-change features are steady for stroke order variations because these features express where and in which direction each of character’s coordinates change which is independent of stroke order variations. Furthermore, the direction-change features extracted from imaginary strokes in the pen-up state are steady for stroke count variations due to stroke connections. We think that because our methods can make the most of the strong points of both the directional features and the direction-change features, the recognition rates are improved.

Chapter 5

Improvements of Recognition System

SUMMARY

Our statistical approach to on-line character recognition, which simultaneously uses directional features, direction-change features and written-area features, improves the recognition rate by adjusting the weights of these features.

Then, we discuss possibilities for future work to improve the method based on recognition results.

5.1 Introduction

Our basic recognition method is already described in Chapter 4. In this chapter, we discuss improvements made to our method. First, the experimental results of improvements made by adjusting the weights of the features are described in Sec. 5.1. Next, in Sec. 5.2 we discuss possibilities for future work to improve the method based on recognition results.

5.2 Adjusting directional, direction-change and written-area features

5.2.1 Adjusting the weights of features

In our basic recognition method described in Chapter 4, directional features, direction-change features and written-area features have no weights. In this section, we show how to improve the recognition rate by assigning and adjusting weights of these features.

Our recognition method was slightly changed to incorporate a diagram that plots the weights of directional features, direction-change features and written-area features. After the feature patterns are blurred, each feature is multiplied by its weight. The other processes of this diagram are the same as those of our basic method's diagram (our fourth method) described in Chapter 4.

5.2.2 Experiments

In the experiments, we used samples in even-numbered sets as learning data and samples in odd-numbered sets as unknown data, in the same way as shown in Chapter 4.

First, we obtained the feature weights that give the best recognition rate for 10 even-numbered data sets. The recognition rates for various weights of the written-area feature and of the direction-change feature are shown in Figs. 5.1 and 5.2, respectively, when each weight is changed by 0.1 steps.

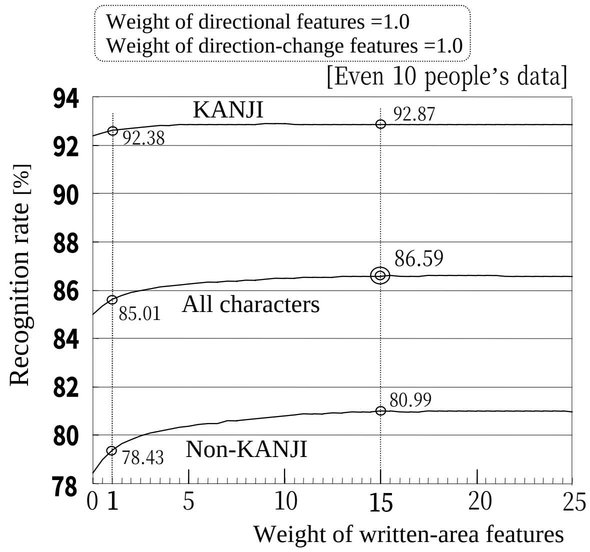


Fig. 5.1 Recognition rates for various values of written-area feature weights

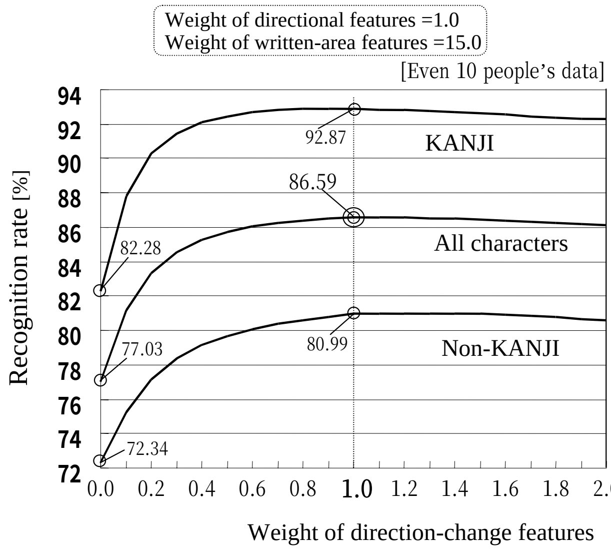


Fig. 5.2 Recognition rates for various values of direction-change feature weights

The best recognition rate for all characters is obtained when the weights of directional features, written-area features and direction-change features are 1, 15 and 1, respectively. Then the recognition rates and classification rates are obtained for all odd-numbered sets (41 data sets) by using those features' weights, as shown in Table 5.1 (best weight). Table 5.1 indicates that the recognition rate and classification rate improve by adjusting the feature weights.

Table.5.1 Recognition rates and classification rates

	Features' weight			Recognition rate			Classification rate (20th)		
	Directional	Written-area	Direction-change	KANJI	Non-KANJI	ALL	KANJI	Non-KANJI	ALL
Best weight	1	15	1	91.79%	83.10%	87.20%	99.27%	98.95%	99.10%
NO weight	1	1	1	91.41%	81.77%	86.32%	99.30%	98.04%	98.63%
case A	1	15	0	83.14%	75.46%	79.08%	98.12%	97.23%	97.65%
case B	1	1	0	82.58%	73.71%	77.89%	97.90%	94.99%	96.36%
case C	1	0	0	81.94%	72.13%	76.76%	97.47%	92.58%	94.89%

(41 odd-numbered data sets)

5.3 Possibilities for improvements

Strong points and results of our recognition method are discussed in Chapters 3 and 4. However, a few problems remain to be solved. We discuss possibilities for improving our method below. We believe that the accuracy of our recognition method can be further improved by making the best use of its strong points.

(a) Dependence on stroke-order variations

As shown in Sec. 4.3 and Sec. 5.2, our method's recognition rate improves from 77.89% to 87.20% (9.31% up) over the traditional method. Of this improvement, 11.21% of all data were correctly changed while 1.90% of all data were incorrectly changed.

For the cases incorrectly changed, we found that the major problem was stroke-order variations in few-stroke-count characters. Fundamentally, our method must be made more effective for stroke-order variations than a structure analysis method that uses stroke-order; this is because directional features are off-line features and direction-change features express where and in which direction strokes are changed. However, incorrect recognition can be caused by dependence on stroke-order because direction-change features at starting points and endpoints of imaginary strokes are influenced by stroke-order (Fig. 5.3). Such incorrect recognition rarely occurs in many-stroke-count characters because features only in the positions where strokeorder varies are influenced. However, such cases of incorrect recognition sometimes occur in few-stroke-count characters.

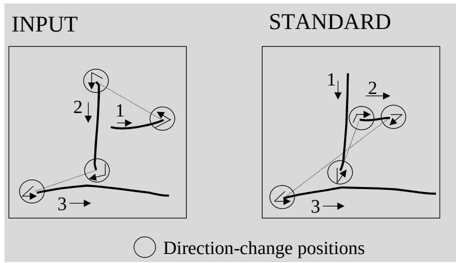


Fig. 5.3 Dependence on stroke-order variations

We believe that multi-template matching is an appropriate way to this problem. This is a well known method in which multi-standard patterns are used for a character. When the differences in direction-change features in the same character is large, multi-standard feature patterns are created (Fig. 5.4).

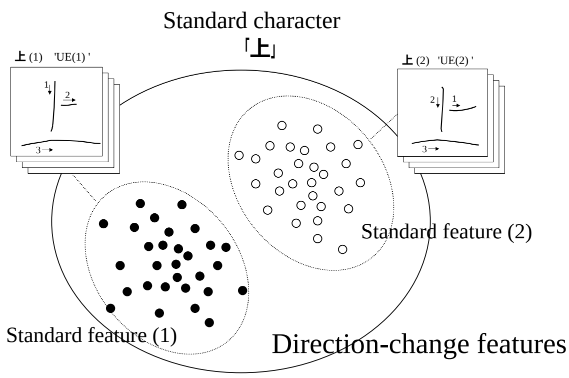


Fig. 5.4 Multi-template matching

These wrong cases fortunately only occur in few-stroke-count characters. Therefore, the recognition dictionary’s size does not greatly increase by adopting multi-template matching.

(b) Learning dictionary

In a statistical method, increasing the learning data improves the recognition accuracy because many features’ variations are automatically reflected in the recognition dictionary. Therefore, our method’s recognition accuracy can be improved by using many more handwritten data. That is, a sufficiently large on-line handwritten database is critical. Accordingly, we are now cooperating with others who are creating a handwritten database (HANDS-kuchibue) by providing handwritten data to them.

Moreover, a statistical method makes it easy to recognize not only Japanese characters but also other language's characters, such as Chinese characters, by simply adopting a learning dictionary.

(C) Adaptation to personal handwritten characters

The recognition results for each database set (numbers corresponding to each writer) are shown in Table 5.2. Examples of each writer's database set are shown in the Appendix (2).

Table 5.2 Recognition rate for each data set

Even-numbered data sets (Learning data)

No.	Writer#	KANJI	Non-KANJI	ALL
1	32	98.85	93.65	96.10
2	50	98.78	93.50	95.99
3	72	98.21	93.15	95.54
4	64	98.19	92.10	94.98
5	6	98.05	90.85	94.25
6	44	98.53	90.38	94.22
7	62	99.11	89.81	94.20
8	56	98.32	90.17	94.01
9	28	96.15	91.34	93.61
10	68	97.06	89.14	92.88
11	38	97.87	87.70	92.50
12	12	97.45	87.34	92.11
13	58	95.68	88.73	92.01
14	54	94.72	89.59	92.01
15	66	95.34	88.15	91.54
16	40	96.17	86.71	91.17
17	20	97.24	85.65	91.11
18	18	94.81	86.52	90.43
19	48	95.82	85.41	90.32
20	30	97.13	83.53	89.94
21	52	95.43	84.57	89.69
22	74	94.45	85.44	89.69
23	70	93.90	85.71	89.58
24	42	95.43	82.83	88.77
25	60	93.00	84.44	88.48
26	4	95.43	80.98	87.79
27	36	96.07	78.92	87.01
28	34	93.53	80.90	86.86
29	76	94.77	79.13	86.51
30	2	97.04	76.42	86.15
31	26	94.56	77.50	85.55
32	10	93.07	75.27	83.66
33	16	93.20	75.12	83.65
34	22	88.96	76.47	82.36
35	14	83.54	79.70	81.51
36	24	89.03	67.37	77.59
37	46	82.44	70.66	76.22
38	8	78.89	72.04	75.27
39	80	63.55	62.53	63.01
40	78	60.73	45.42	52.64
All		93.01	82.62	87.52

(rate:%)

Odd-numbered data sets (Unknown data)

No.	Writer#	KANJI	Non-KANJI	ALL
1	5	98.26	94.78	96.42
2	17	97.77	91.50	94.46
3	61	97.70	91.55	94.45
4	7	98.21	89.59	93.65
5	69	97.50	90.03	93.55
6	47	97.32	90.01	93.46
7	19	97.77	89.14	93.21
8	59	97.66	88.53	92.84
9	43	96.23	88.72	92.26
10	45	96.79	88.16	92.23
11	53	95.29	88.87	91.90
12	49	97.04	87.09	91.78
13	65	95.46	88.35	91.71
14	27	95.64	87.32	91.25
15	33	97.43	85.41	91.08
16	67	94.81	85.41	89.84
17	55	95.02	85.16	89.81
18	15	95.57	84.14	89.53
19	21	96.79	82.85	89.42
20	31	92.65	85.49	88.86
21	23	92.04	86.01	88.86
22	9	95.27	82.70	88.63
23	73	94.26	83.07	88.35
24	41	97.02	80.46	88.27
25	29	92.45	84.11	88.05
26	37	91.56	84.25	87.70
27	25	93.94	80.42	86.80
28	79	89.76	84.00	86.72
29	81	91.51	82.23	86.61
30	77	92.29	79.97	85.78
31	13	90.16	76.42	82.90
32	51	89.56	76.64	82.74
33	3	87.65	77.54	82.31
34	75	86.89	76.85	81.58
35	57	84.05	78.51	81.12
36	71	82.86	78.46	80.54
37	39	88.13	72.40	79.82
38	63	87.21	71.42	78.87
39	1	77.65	78.67	78.19
40	11	77.90	65.39	71.29
41	35	50.47	65.55	58.44
All		91.79	83.10	87.20

(rate:%)

As shown in Table 5.2, some data sets have low recognition rates. Most characters in these data sets have very low quality. To deal with these very-low-quality characters, it is effective to utilize a learning dictionary for each writer. As shown above, our method based on a statistical approach is suitable for use with a learning dictionary. To confirm this potential, we conducted preparatory recognition experiments on each data set by using a dictionary learned by its own data. This experiment's results are shown in Table 5.3.

Table 5.3 Recognition rates using dictionaries learned by each data set

	Data set nb.	Dictionary: Even-numberd 40 sets			Dictionary: Each own set		
		KANJI	Non-KANJI	ALL	KANJI	Non-KANJI	ALL
Best 3	5	98.26	94.78	96.42	99.89	98.69	99.26
	17	97.77	91.50	94.46	99.91	98.20	99.01
	61	97.70	91.55	94.45	99.77	98.61	99.16
Worst 3	1	77.65	78.67	78.19	99.56	95.19	97.25
	11	77.90	65.39	71.29	99.59	91.60	95.37
	35	50.47	65.55	58.44	99.17	93.21	96.02

(rate:%)

As shown in Table 5.3, the recognition rate for each data set is improved by using a dictionary learned by its own data, even if the quality of the data set is very low. This result would seem natural for a statistical method. However, in a structure analysis method such a learning dictionary is not easy to apply.

(D) Classification accuracy for post-processing and string separation

In recognition systems using post-processing and string separation, not only recognition rates but also classification rates for candidates are important. Consequently, classification speed is also important. Table 5.4 shows classification rates. Table 5.4 (a) shows the classification rates by recognition step, and Table 5.4 (b) shows those by only pre-classification step.

The 20th classification rate for all unknown data sets by recognition step is 99.10%. That by pre-classification step is 98.75%. Since the 0.35% difference between these classification rates is so small, pre-classification alone, without the recognition step, can be used for post-processing and string separation.

Table 5.3 Classification rates

(a) Recognition

Candidates	KANJI	Non-KANJI	ALL
1st	91.79	83.10	87.20
2nd	95.58	89.48	92.36
3rd	96.93	93.16	94.93
4th	97.65	94.91	96.20
5th	98.02	95.96	96.93
6th	98.26	96.59	97.38
7th	98.45	97.08	97.73
8th	98.61	97.45	97.99
9th	98.72	97.72	98.19
10th	98.82	97.92	98.35
11th	98.89	98.11	98.48
12th	98.95	98.26	98.59
13th	99.01	98.39	98.68
14th	99.06	98.50	98.76
15th	99.11	98.60	98.84
16th	99.15	98.69	98.91
17th	99.19	98.76	98.96
18th	99.22	98.83	99.02
19th	99.25	98.90	99.06
20th	99.27	98.95	99.10

[odd numbered 41 data sets] (rate:%)

(b) Pre-classification

Candidates	KANJI	Non-KANJI	ALL
1st	89.08	81.53	85.09
2nd	93.79	88.26	90.87
3rd	95.44	92.07	93.66
4th	96.41	93.87	95.07
5th	96.96	95.06	95.95
6th	97.32	95.79	96.51
7th	97.56	96.37	96.93
8th	97.76	96.82	97.27
9th	97.93	97.17	97.53
10th	98.07	97.46	97.75
11th	98.18	97.72	97.94
12th	98.28	97.92	98.09
13th	98.37	98.07	98.22
14th	98.45	98.23	98.33
15th	98.51	98.34	98.42
16th	98.57	98.45	98.51
17th	98.62	98.54	98.58
18th	98.68	98.62	98.65
19th	98.72	98.69	98.70
20th	98.76	98.75	98.75

[odd numbered 41 data sets] (rate:%)

5.4 Conclusion

Adjusting the weights of features improves the recognition rate. Our method's final recognition rates are 97.20% for all characters, 91.79% for KANJI characters and 83.10% for non-KANJI characters of unknown data.

We investigated ways to improve our recognition system. Although wrong results are sometimes caused by a dependence on stroke-order variations in few-stroke-count characters, this problem can be solved by adopting multi-template matching.

Our method, which is based on a statistical approach, has the strong advantage of having an automatic learning dictionary. Accordingly, the recognition rate can be improved by using a learning dictionary that contains a very large quantity of handwritten data. Moreover, it is possible to apply our method to recognition of other language's characters and to adapt the system to personal handwritten characters. It was also found that pre-classification without a recognition step is adequate for post-processing and string separation.

In the future, we will investigate better methods of extracting direction-change features of the imaginary strokes in the pen-up state to improve the recognition rate as well as ways of efficiently reducing the features' dimensions to speed up the recognition process.

Chapter 6

Conclusion

This thesis described our research on on-line character string separation and on-line character recognition. These technologies are important for easily inputting characters with a pen. Our string separation method unifies physical features, character recognition and language processing using network expressions to correctly segment a string of characters. Our character recognition method is based on pattern matching that simultaneously uses both directional features that are off-line features and direction-change features that we designed as on-line features. Our methods improved string separation accuracy and character recognition accuracy.

(a) Character string separation method

We introduced two string separation methods. Our first method, using Multi-level network expressions, sums up the score of physical features in Network A and the score of logical features in Network B. Our second method, using Unified network expressions, unifies the score of physical features and the score of logical features by using only one network.

The string separation rate could be improved by our methods for unknown data set consisting of Japanese strings freely written by 21 different people. Using the conventional Murase method, the string separation rate is 85.54% for all strings; using our Multi-level network expression, the string separation rate is 86.73% for all strings; using our Unified network expression, the string separation rate is 90.72% for all strings.

Our methods improved the string separation rate because they obtained separation results by using both physical features and logical features. The rate of our second method (Unified network) is better than that of our first method (Multi-level network) because the second method unifies the physical features, character recognition, and language processing more effectively than the first method.

In the future, we will clarify how much weight should be given to each feature in the network expression. We believe that the string separation rate can be further improved by better assigning weights to nodes and links in the network expression. Moreover, we believe it is important to improve character recognition accuracy to achieve a higher character separation rate.

(b) Character recognition method

On-line character recognition generally uses structure analysis approaches. However, CPU accuracy has recently improved, the cost of memory has rapidly decreased, and various character features can be used. Therefore, we believe a statistical approach holds great promise because it has the advantage of its recognition dictionary being able to learn character features from many handwritten data without any manual work. Therefore, we have pursued character recognition based on a statistical approach.

We propose a new on-line recognition method that simultaneously uses both directional features, otherwise known as off-line features, and direction-change features, which we designed as on-line features. The direction-change features express where and in which direction the character’s stroke changes.

In a recognition experiment with a public on-line handwritten database (HANDS-kuchibue_d-96-02), recognition rate improved from 77.89% to 87.20% over the traditional method of using only directional features. We confirmed our method is effective in handling stroke shape, stroke-order and stroke count variations. Furthermore, we think our method can effectively unify the stroke points of directional features and direction-change features. The directional features are fundamentally appropriate for handling stroke order variations because these features are off-line features. By simultaneously using both directional features and direction-change features in the pen down state, the recognition method can more effectively handle stroke shape variations. Moreover, by using direction-change features in the both pen down and pen up states, the recognition method can better handle stroke-count variations caused by stroke connections.

In our method based on a statistical approach, the recognition dictionary can easily learn character features without manual work. Accordingly, our method’s accuracy can be improved by augmenting its learning dictionary with large quantities of handwritten data. In the future, we will investigate better methods of extracting direction-change features of the imaginary strokes in the pen-up state for improving the recognition rate and how to efficiently reduce the features’ dimensions to speed up the recognition process. From such investigations, not only character recognition accuracy but also character string separation accuracy is expected to improve.

Appendixes

Appendix (1)

[String Separation]

Table A1-1 Kinds of handwritten character strings for separation experiments

(a) Kanji character strings

九州　山川　問題　関係　優勝　予算　以上　住宅　経費　時間
委員会　可能性　昼食会　冷蔵庫　電話機　自動車　小学校　名古屋
交通費　北海道　交際費　英会話　化粧品
調査報告　代表説明　国際会議　電話番号　時間短縮　重要書類
連絡事項　千代田区　情報通信　携帯電話
電子工学科　日本武道館　三栄不動産　世界選手権　中央郵便局
国家公務員　従業員募集　統一地方選挙　地方公共団体
三洋電機株式会社

(b) Mixture character strings

打合せ　値下げ　考える　代わる　賃上げ　重量上げ　お父さん
打ち合せ　説明する　案内する　受け取る　梅雨明け　セット販売
アンケート調査　グループ分け　ＦＡＸ受信　エネルギー消費
イメージ処理　電子メール　パターン認識　自動切り出し技術
集会を行う。　集会に参加した　明日先生に会う。　受け付け時間
文字を認識する　ＦＡＸを送信する　彼にメモを渡す。
システムを導入する　ワープロを練習する　電話して下さい
時間を変更したい　私は、集会に参加した。　第２段階
１１２９番地　７２０番地　３４７番地　５丁目　７の１
７５９－３１　０５８４－８１－０５４１
０５８４６４－３８６５　０３－８８１－０５４１

Table A1-2 String separation rates by using Unified Network

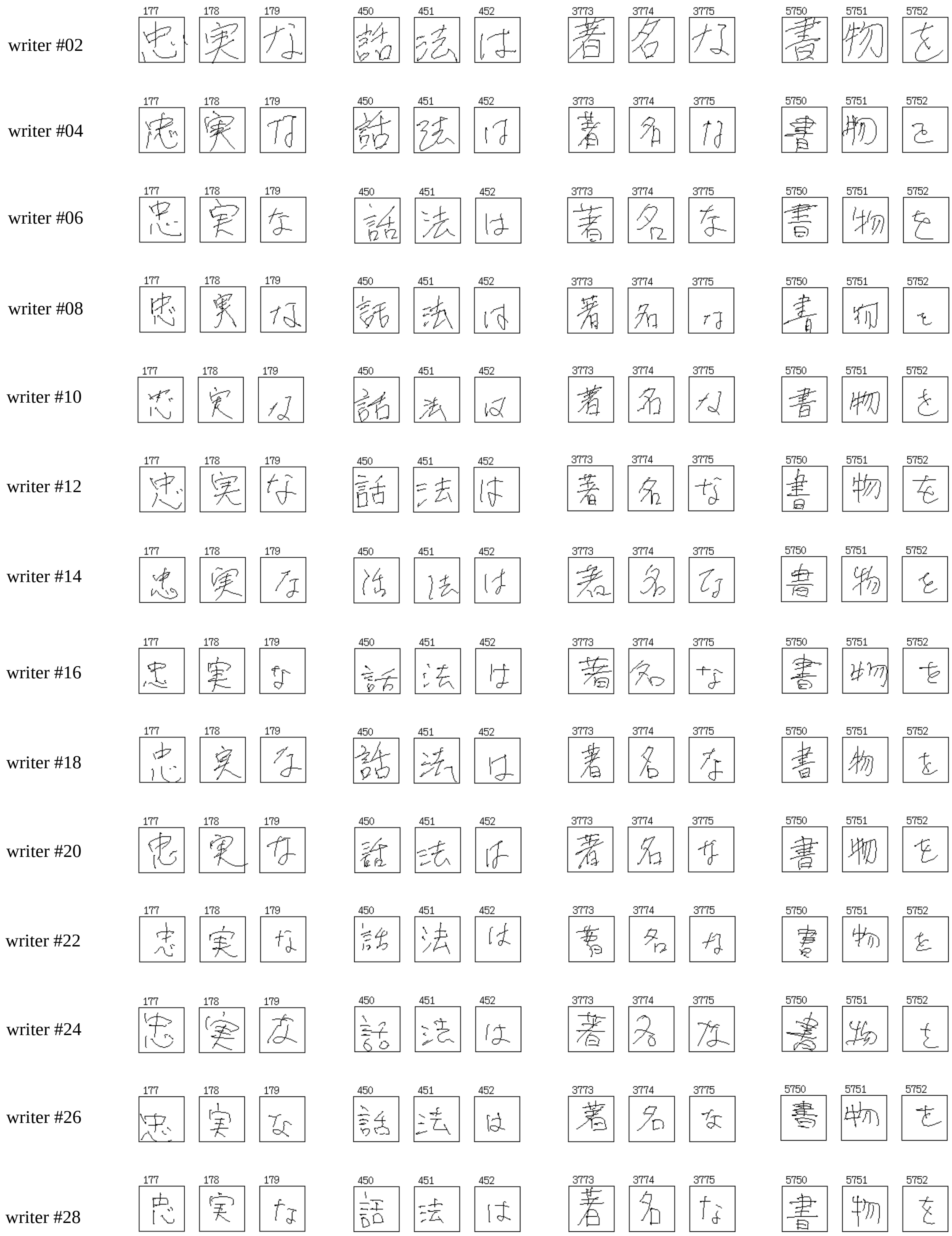
(a) Learning data (set1)

no.	Writer	KANJI	MIX	ALL
1	6	98.28	97.85	98.01
2	4	99.26	95.65	97.18
3	18	100.00	94.06	96.47
4	8	100.00	93.88	96.20
5	19	98.57	93.62	95.73
6	11	96.97	94.57	95.57
7	9	98.39	93.27	95.18
8	13	100.00	89.74	94.67
9	17	95.89	93.62	94.61
10	16	96.88	91.95	94.04
11	2	99.04	90.86	93.91
12	14	100.00	89.13	93.83
13	1	99.18	89.84	93.53
14	20	97.73	90.29	93.49
15	10	100.00	86.42	92.67
16	21	93.02	90.85	91.81
17	3	97.81	85.63	91.12
18	7	100.00	77.08	90.83
19	5	100.00	82.95	90.07
20	12	94.44	86.60	89.94
21	15	86.57	84.62	85.52
all		97.68	90.58	93.60

(b) Unknown data (set 2)

no.	Writer	KANJI	MIX	ALL
1	41	98.48	96.77	97.48
2	29	95.59	96.59	96.15
3	38	98.53	93.83	95.97
4	24	98.39	92.31	95.13
5	31	95.65	94.62	95.06
6	39	100.00	90.43	94.34
7	40	100.00	90.22	94.08
8	35	98.00	91.14	93.80
9	27	96.83	90.30	93.13
10	30	100.00	87.21	93.04
11	26	96.06	90.37	92.68
12	42	100.00	87.13	92.53
13	28	99.31	86.56	92.15
14	34	94.44	88.64	91.25
15	36	100.00	80.43	89.02
16	33	98.28	82.47	88.39
17	37	97.06	80.25	87.92
18	25	97.41	81.52	87.67
19	23	98.36	78.31	86.81
20	43	97.12	59.60	77.59
21	32	89.83	66.67	75.66
all		97.61	85.57	90.72

Appendix (2) **[Character Recognition]**

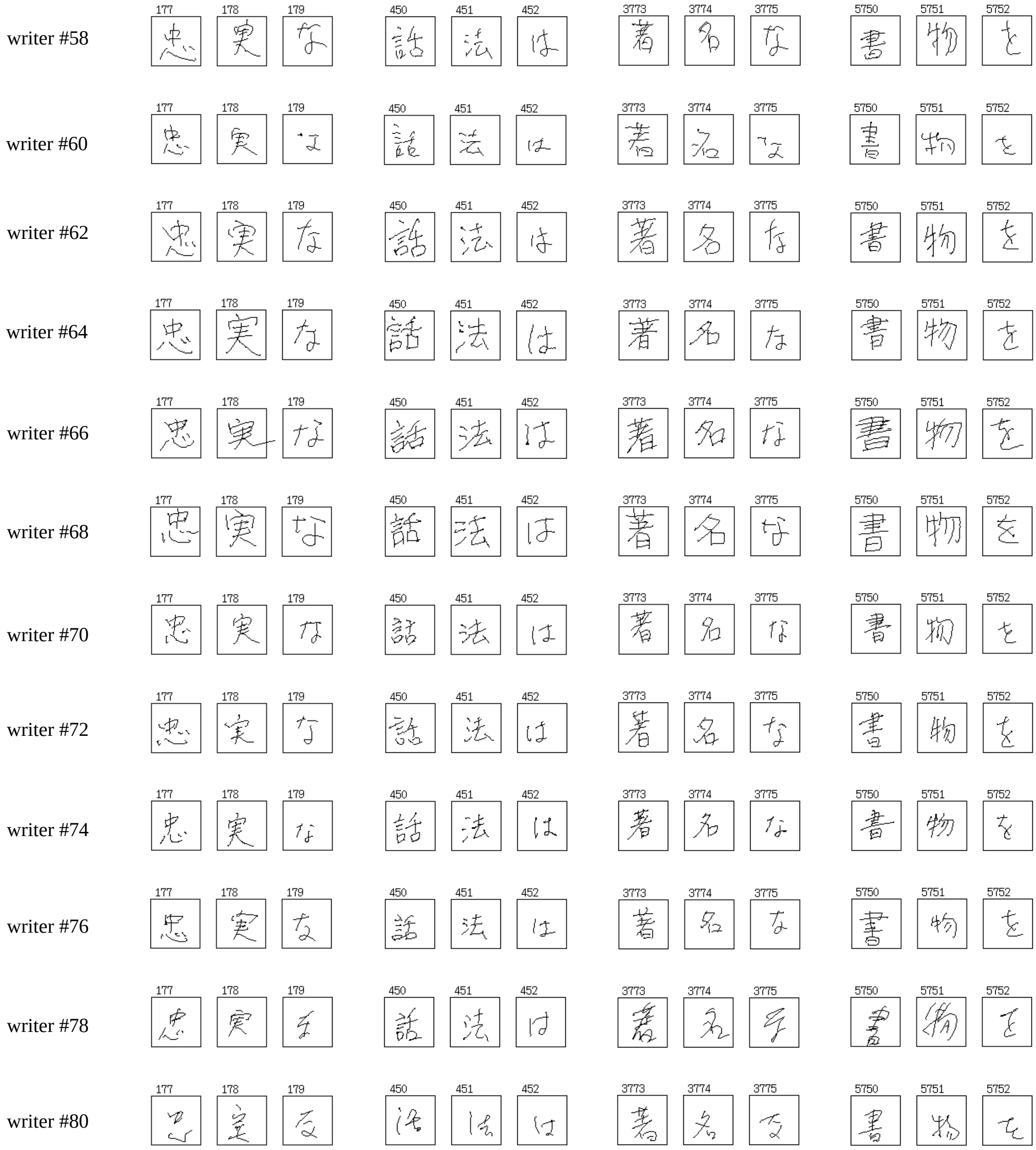


(a)

Fig. A2-1 Examples of handwritten characters for character recognition experiments (HANDS-kuchibue_d-96-02) (Even numbered data set)

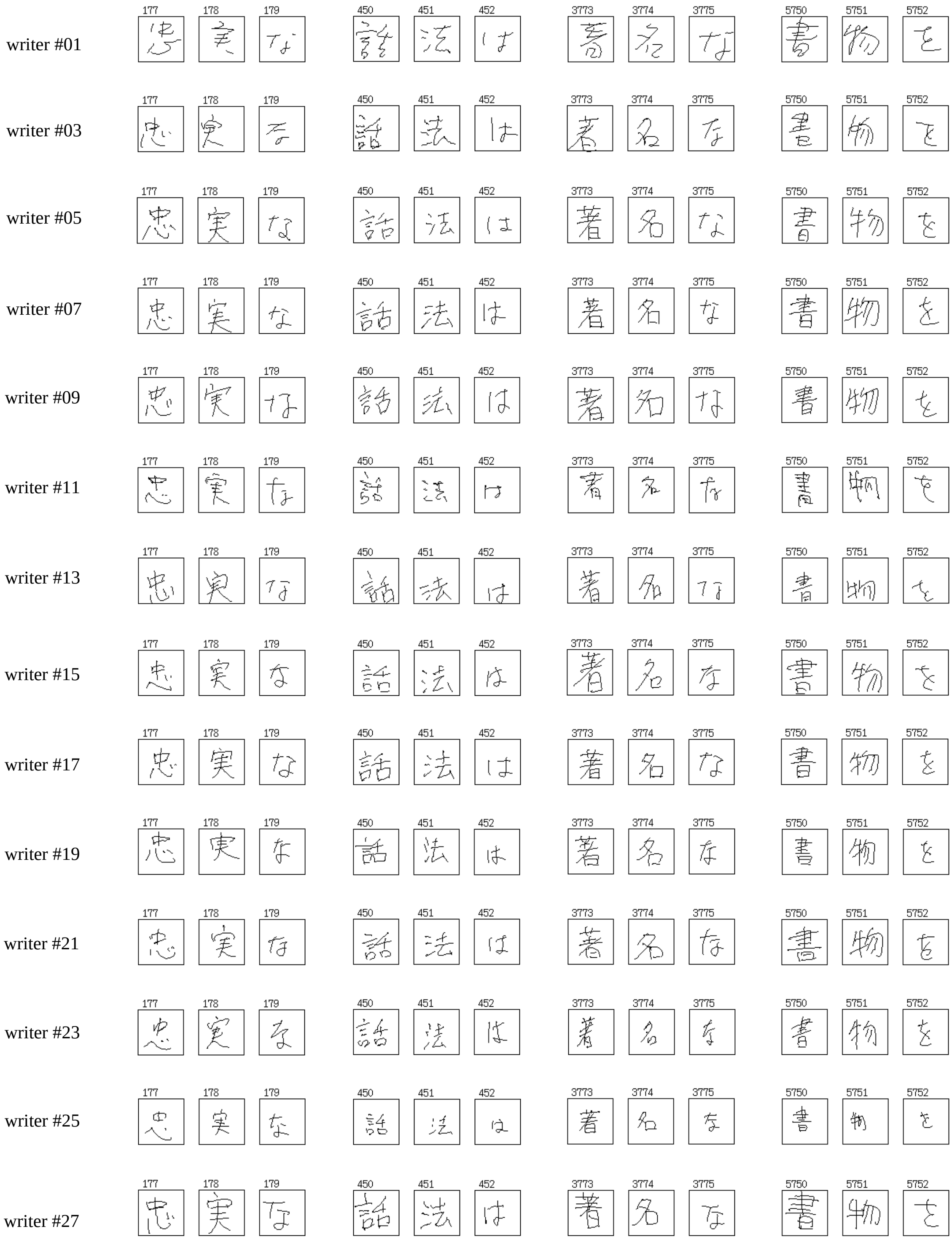
(b)

Fig. A2-1 Examples of handwritten characters for character recognition experiments (HANDS-kuchibue_d-96-02) (Even numbered data set)



(c)

Fig. A2-1 Examples of handwritten characters for character recognition experiments (HANDS-kuchibue_d-96-02) (Even numbered data set)

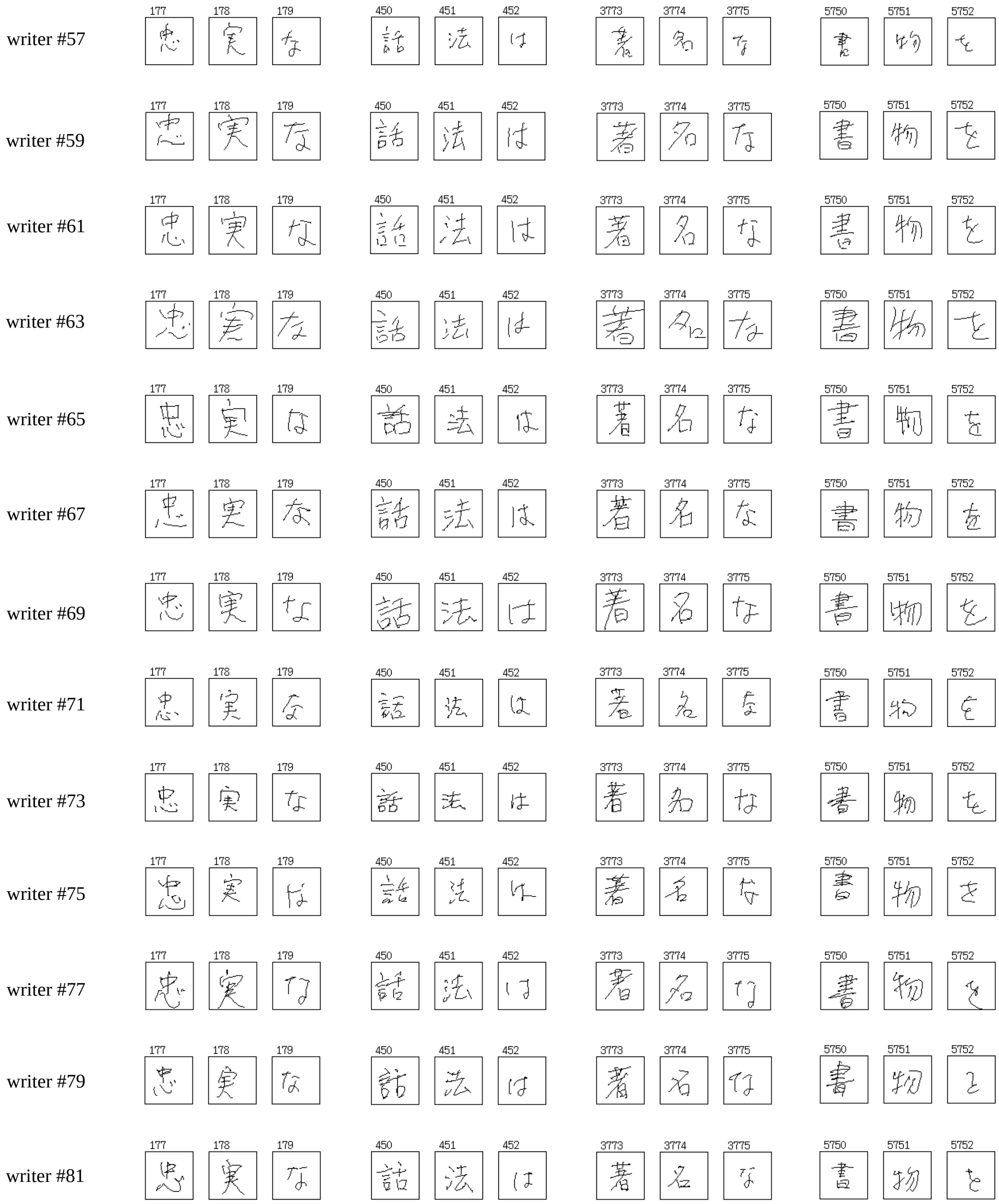


(a)

Fig. A2-2 Examples of handwritten characters for character recognition experiments (HANDS-kuchibue_d-96-02) (Odd numbered data set)

(b)

Fig. A2-2 Examples of handwritten characters for character recognition experiments (HANDS-kuchibue_d-96-02) (Odd numbered data set)



(c)

Fig. A2-2 Examples of handwritten characters for character recognition experiments (HANDS-kuchibue_d-96-02) (Odd numbered data set)

Table A2-1 Similar character pairs which are counted as identical categories when calculating recognition rates

Hiragana		
ぁ	−	あ
ぃ	−	い
ぅ	−	う
ぇ	−	え
ぉ	−	お
っ	−	つ
ゃ	−	や
ゅ	−	ゆ
ょ	−	よ
ゎ	−	わ

Katakana		
ァ	−	ア
ィ	−	イ
ゥ	−	ウ
ェ	−	エ
ォ	−	オ
ヵ	−	カ
ヶ	−	ケ
ッ	−	ツ
ャ	−	ヤ
ュ	−	ユ
ョ	−	ヨ
ヮ	−	ワ

Roman Alphabets		
c	−	C
o	−	O
p	−	P
s	−	S
u	−	U
v	−	V
x	−	X
w	−	W
z	−	Z

Table A2-2 Characters which are 100% recognized correctly

(358 characters)

契 峠 透 禿 噸 瀞 刑 戟 欺 努　灯 稽 罫 引 敦 乳 断 畷 矩 葱
寧 祢 隈 歳 杜 串 馴 楢 曇 虎　亭 誇 挺 菰 汀 梯 醐 珍 垢 誤
跨 壷 娯 糊 牽 形 纏 極 砥 澱　剣 諺 鼎 汐 敵 添 軒 鹸 玖 汗
肝 艦 款 廟 豹 柑 斐 吃 匪 碑　既 亀 罷 姦 埠 侃 刊 苅 斧 刈
怖 蒜 謄 塗 彬 勘 雁 斌 矧 悉　僅 萩 函 曝 剥 埜 吟 棚 悩 婆

勢 机 卿 脚 称 筏 競 燕 荊 黍　静 禦 肌 渠 髪 醗 《 逝 洲 誓
愁 戚 趣 懇 雛 趨 叔 酬 征 衆　週 遮 協 舛 凹 銭 滋 璽 糎 尖
謝 窃 赦 繊 煽 蕊 抄 陥 招 廠　邸 拭 将 醤 詔 裳 孟 昇 丞 沼
嘗 厨 諏 竣 薮 淑 髄 粛 匠 疹　熙 鋤 尽 舜 醇 疎 茶 智 恥 抽
剛 峡 甑 坐 斎 堺 辿 蘇 淡 沙　砿 琵 宏 巧 琶 孔 勅 湖 肱 酵

悠 昂 牒 控 彫 遭 暫 仔 徳 燦　珊 蚕 双 鼠 遡 匝 孜 斯 獅 桟
咲 腿 替 榊 托 啄 瀧 汰 翠 其　柁 匙 冊 岱 符 勃 穆 ＿ 臥 ／
梁 猟 墨 凌 蛾 ∴ 茄 嘉 伽 〒　俄 峨 ＼ 奔 葎 傘 褒 忙 拐 商
迅 獄 麺 魅 冠 献 裁 灰 械 就　魁 琳 咳 憲 芥 凱 → 蔚 耶 欝
余 勝 餅 亥 籾 爺 貫 碧 隠 鼓　臓 幽 ◇ 遊 熱 咽 癒 刺 掩 鯵

岬 霧 依 羅 荻 ε 艶 唖 叡 曳　茂 戯 瓜 詠 液 盈 慰 尉 筋 逃
腕 聯 圃 套 甫 囲 弗 蔽 赫 箆　距 轄 封 椛 威 毒 覆 廃 急 傑
隔 芙 脇 賦 塀 漱 弁 聾 腐 暦　糞 那 倣 沸 馨 穂 幣 岳

Table A2-3 Worst 100 characters in recognition rates

(The characters in 1st candidates' list are in order of frequency)

no.	Characters	Recognition rate (%)	1st candidates' list
1	l (Alphabet)	12.20	｜ 1 l ／ 火 y て x χ モ
2	ー (Katakana)	13.15	ー ─ — ー _ ‾ ／ ′
3	x (Alphabet)	17.07	× x χ X X 入 ブ 北
4	t	26.02	七 t も ナ ι λ 寸 ＋ 十 キ
5	ロ (Katakana)	31.01	口 ロ □ D 2 ▽ n ヮ コ ク
6	ペ (Katakana)	31.71	ぺ ペ ↑ や へ パ べ υ
7	× (Symbol)	31.71	× X X メ x 丈 χ 人
8	エ (Katakana)	31.71	エ 工 I ェ ユ I 二 ヱ z Z
9	一 (Kanji)	33.33	一 ─ — ー _ ‾ ／ ‥
10	0 (Numeric)	33.41	0 O o o θ 〇 U ι し 6
11	口 (Kanji)	34.15	口 ロ □ D コ ク り 白 2
12	べ (Katakana)	35.77	べ ベ へ 心 バ V
13	— (Symbol)	36.59	— ─ 一 ー ‾ _
14	カ (Katakana)	36.79	カ 力 ヵ 刀 ャ つ ウ フ D 巾
15	— (Symbol)	37.80	─ — 一 ー _ ‾ ＼
16	く	39.53	く 〈 ＜ C （ c 乙 ィ Δ 又
17	1 (Numeric)	39.84	1 ｜ ／ I ！ ノ 〕 l 工 へ
18	ニ	40.77	二 ニ こ ＝ ン ュ _ I
19	Σ	41.46	Σ Σ ε 乙
20	I	41.46	I I ｜ エ 工 τ
21	o (Alphabet)	43.29	o o O 。 0 a θ 〇 σ 〇
22	タ (Katakana)	43.36	タ 夕 勺 ヌ フ ク 与 ヲ ラ y
23	二	44.60	二 ニ こ ＝ ュ ノ ン _ 三 I
24	力 (Kanji)	45.53	力 カ ヵ 刀 や ↑ 巾 ρ ャ つ
25	＋	46.34	＋ 十 t f T 斗 ナ
26	〉	46.34	〉 ＞ ゝ つ ノ ヌ
27	I	47.56	I I ｜ エ 1 工 了 2
28	7	48.78	7 ワ ク つ フ ヮ ？ ／ ノ ァ
29	X	48.78	X × X x 必 メ
30	〕	48.78	〕 , ） コ
31	ゑ	48.78	ゑ 乏 急 怠 す 免 烹 之 蒐 浮
32	八	48.78	八 ハ 入 へ ∧ へ 凡 け バ い
33	σ	48.78	σ O 6 J o 〆 o の u 「
34	工 (Kanji)	51.22	工 エ I ェ ヱ I ユ T τ
35	ν	51.22	ν υ レ ∨ γ ン V v ひ r
36	ξ	51.22	ξ § 弓 言 ζ 3 亨 ε ；
37	V	51.22	V ∨ v V Ⅳ T r
38	o (Symbol)	51.22	o o O ① 0 a 。 θ D 〇
39	O (Alphabet)	52.85	O 0 o 〇 o θ 〇 も C ◇
40	〈	53.66	〈 ＜ く （
41	ι	53.66	ι レ し l 1 ν ： 七 τ 乙
42	Σ	56.10	Σ Σ ミ
43	Δ	56.10	Δ △ 八
44	丁	56.10	丁 T J 「
45	討	58.54	討 計 訂 初 対
46	q	58.54	q 9 φ g o ξ τ 元 F 切
47	ト (Katakana)	59.44	ト 卜 人 r k に i L f h
48	!	59.76	！ ｜ 1 ／ i ノ （ ｝
49	十 (Kanji)	60.06	十 ＋ t ナ T 斗 寸 f ÷ 4
50	ぺ (Katakana)	60.28	ぺ ペ べ ベ パ ↑ ∧ プ へ ポ
51	鳥	60.98	鳥 烏 鳶 良 馬 套 島
52	ゝ	60.98	ゝ ＞ 〉 ＼ へ s
53	べ	60.98	べ ベ へ 心 バ ↑ ～ ド M N
54	ろ	61.52	ろ 3 う ぅ 弓 ら ？ 了 〕 万
55	ら	61.98	ら 5 s 3 ゥ ぅ う ラ ； ち
56	n	62.20	n m h u の μ わ ～ 凡 れ
57	y	62.60	y ／ g ソ な ょ ノ ン γ 〉
58	て	62.94	て τ 7 乙 ？ マ フ z つ λ
59	□ (Symbol)	63.41	□ 口 ロ 位 O ∨
60	令	63.41	令 今 々 冷 分 予 市
61	＜	63.41	＜ 〈 く
62	τ	63.41	τ て 乙 T ν 2 公 λ 七
63	.	64.23	. 、 ・
64	へ (Katakana)	64.81	へ ヘ ∧ ～ 八 ＾ ＼ 入 l
65	J	65.85	J 丁 j 」 ナ
66	κ	65.85	κ K k ん ι 尤 x h
67	j	65.85	j J よ 」 i ば g ょ 〆 ；
68	γ	65.85	γ r ν ツ 〆 Y ↑ y
69	問	65.85	問 間 同 閃 内 開 閉 関 す 伺
70	‾	65.85	‾ 一 — ─ ー _
71	η	65.85	η ク n わ ル の ） 九 へ 川
72	ζ	65.85	ζ ら 弓 ち s S 万 ξ
73	リ (Katakana)	66.55	リ り ソ ノ 川 ‖ ッ ） ク y
74	’	67.07	’ ′ ° ） ノ 1 野 プ 〉
75	ヌ	67.07	ヌ 又 ス タ フ マ
76	上	67.25	上 土 ± 工 二 ト 卜 L
77	メ	67.99	メ × X ノ y 人 ／ x 丈 X
78	治	68.29	冶 治 沿 右 そ
79	失	68.29	失 矢 夫 天 ≠ 快
80	☆	68.29	☆ Δ 丸 々 六 ャ o ※ 台 女
81	ハ	68.29	ハ 八 ∧ い に
82	#	68.29	＃ 井 ≠ チ 牛
83	φ	68.29	φ Φ ϕ 中 す ① 1
84	も	68.51	も モ 毛 t キ を 七 ≠ ぇ ｛
85	間	68.58	間 問 同 聞 閏 何 内 旬 閃 回
86	チ	68.99	チ 千 干 4 テ f ヶ 升 ヂ ヲ
87	ブ	70.33	ブ づ グ ゾ ス ズ √ 久 ？ ゴ
88	◎	70.73	◎ O o @ ⑧ ⑦ ⑤
89	・	70.73	・ 、 . ′ ─ ー
90	才	70.73	才 ォ オ ナ 丁 木 f
91	負	70.73	負 貞 員 項 頁 貝 偵 見 塾
92	ϕ	70.73	ϕ φ υ 十 χ
93	Q	70.73	Q a 0 な ⑥ e ③ θ O
94	宮	70.73	宮 宕 官 宣 育 書 冨 苦 富 言
95	f	70.73	f ナ 十 子 「 ノ t ヲ ｛ チ
96	氷	70.73	氷 水 永
97	鳩	70.73	鳩 嶋 鳴 進 鴻 傾 嬬 隔 磁
98	鳥	70.73	烏 鳥 馬 鳴 隻 島
99	♂	70.73	♂ ↑ 6 る 杏 舌 合
100	ト (Kanji)	70.73	卜 ト i

Table A2-4 Charatcers improved better than 20% in recognition rates by using direction-change features

(379 characters)

q	t	模	ヶ	抜	棒	采	案	ブ	び	X	妻	ヰ	七	大	穫	べ	番	審	寝
侮	庚	蕃	彼	伍	秩	養	九	η	枝	ツ	友	カ	0	ぺ	横	普	晴	シ	る
女	べ	仮	諾	鹿	巣	渥	宇	蓉	穐	寒	f	7	奏	裕	尾	υ	ン	か	実
菓	フ	も	た	梓	τ	族	Ⅵ	橋	在	叉	精	僚	侠	搭	僧	媛	闇	宥	又
専	費	香	矢	祐	適	移	春	纂	零	接	棄	稚	借	来	徒	年	便	プ	使
貴	達	道	ち	ク	左	展	屋	償	ャ	ボ	O	ニ	音	著	犠	杵	俸	諌	d
読	頑	遺	億	東	樽	#	皮	o	僑	♂	紬	鳴	槽	逼	雰	更	鏑	賛	巻
椿	擢	ぼ	採	董	稿	菜	薦	錫	グ	責	呑	秦	ぉ	署	拳	孤	釆	昆	ッ
ケ	業	チ	度	支	メ	優	技	遠	十	せ	疏	｛	各	殊	亮	賃	章	永	浜
禾	係	午	升	錘	鴬	栗	憲	倍	黒	犬	丸	奈	底	樗	被	壊	朱	頂	根
藩	g	蓄	港	n	促	蓮	袷	錯	徳	微	妙	拾	噴	庭	穣	擾	粉	震	樟
千	汲	ら	害	線	乗	者	土	二	づ	件	賞	な	ヌ	姓	毎	義	衝	井	美
援	タ	ゃ	て	y	頼	待	太	集	素	故	話	で	そ	子	徹	護	説	丁	粋
尭	芹	慣	失	Ⅶ	哨	鷹	項	噌	ォ	櫓	莱	貰	j	烏	礼	央	秘	情	右
簾	兵	磯	駄	墓	撤	委	披	＋	蔭	謡	痩	佐	唐	授	蓑	住	稔	尼	π
去	嘆	浪	権	！	ゅ	敢	懲	鋲	腸	ι	塞	脊	米	析	隻	餐	虜	寡	宍
ホ	焦	χ	眠	桃	寮	豊	昔	握	ぎ	端	済	寸	課	低	J	慮	後	々	れ
事	セ	察	語	文	先	鉄	連	像	ナ	炭	種	丹	統	喜	夫	含	任	様	位
意	続	ス	キ	島	本	農	手	表	買	T	l	が	水	持	分	検	ワ	違	

Table A2-5 Charatcers degenerated worse than 5% in recognition rates by using direction-change features

(40 characters)

ゑ	☆	◎	.	・	I	Q	閉	’	上	：	Σ	⑥	⑯	註	ぱ	否	②	門	納
ゐ	Δ	乍	釦	‘	魚	狗	⑫	⑤	緋	巡	凸	φ	鞘	ξ	④	蕗	進	○	○

Table A2-6 Classification rates of even numbered 40 data sets (Learning data sets)

(Four) Directional features		100	100	100	100	100	100	100	100	100	0	0	0	0
Written-area feature		100	0	100	100	100	1500	1500	1500	1500	100	100	0	0
(Eight) Direction change features		100	0	0	100	0	100	0	100	0	100	100	100	100
Circle feature		100	0	0	0	100	100	0	0	100	100	0	100	0
1st	KANJI	92.65	82.52	83.23	92.60	83.35	93.01	83.93	92.97	84.10	87.13	87.07	85.58	85.57
	Non-KANJI	81.31	71.20	72.82	80.98	73.91	82.62	74.70	82.33	75.63	69.14	68.46	58.72	58.08
	ALL	86.66	76.54	77.73	86.46	78.36	87.52	79.05	87.35	79.63	77.63	77.24	71.39	71.04
5th	KANJI	97.81	93.66	94.18	97.79	94.29	98.03	94.82	98.00	94.94	95.51	95.46	93.67	93.72
	Non-KANJI	93.66	86.11	88.33	93.45	89.03	95.18	91.19	95.05	91.67	87.28	86.60	71.09	70.84
	ALL	95.62	89.67	91.09	95.50	91.51	96.52	92.91	96.44	93.21	91.16	90.78	81.74	81.64
10th	KANJI	98.67	95.70	96.19	98.65	96.27	98.74	96.61	98.73	96.70	97.01	96.96	94.90	94.99
	Non-KANJI	96.14	89.52	91.92	96.01	92.45	97.31	94.59	97.25	94.90	92.32	91.76	72.60	72.64
	ALL	97.33	92.43	93.93	97.25	94.26	97.99	95.54	97.95	95.75	94.53	94.21	83.12	83.19
20th	KANJI	99.15	96.95	97.45	99.13	97.53	99.17	97.76	99.16	97.83	98.00	97.97	95.67	95.80
	Non-KANJI	97.61	91.73	94.41	97.52	94.76	98.53	96.66	98.51	96.81	95.46	95.14	73.05	73.25
	ALL	98.33	94.19	95.84	98.28	96.06	98.83	97.18	98.82	97.29	96.66	96.47	83.72	83.89
50th	KANJI	99.51	97.87	98.38	99.50	98.43	99.45	98.50	99.43	98.55	98.81	98.79	96.23	96.39
	Non-KANJI	98.68	93.60	96.60	98.65	96.77	99.26	98.14	99.25	98.22	97.92	97.88	73.17	73.46
	ALL	99.07	95.62	97.44	99.05	97.56	99.35	98.31	99.34	98.37	98.34	98.31	84.05	84.27
100th	KANJI	99.62	98.14	86.45	99.62	98.71	99.51	98.67	99.50	98.71	99.08	99.06	96.36	96.53
	Non-KANJI	99.25	94.43	74.87	99.24	98.08	99.57	98.76	99.57	98.82	99.14	99.21	73.19	73.49
	ALL	99.42	96.18	85.14	99.42	98.38	99.54	98.72	99.53	98.77	99.11	99.14	84.12	84.36

Table A2-7 Classification rates of odd numbered 41 data sets (Unknown data sets)

(Four) Directional features		100	100	100	100	100	100	100	100	100	0	0	0	0
Written-area feature		100	0	100	100	100	1500	1500	1500	1500	100	100	0	0
(Eight) Direction change features		100	0	0	100	0	100	0	100	0	100	100	100	100
Circle feature		100	0	0	0	100	100	0	0	100	100	0	100	0
1st	KANJI	91.41	81.94	82.58	91.39	82.63	91.79	83.14	91.77	83.20	84.44	84.41	82.87	82.86
	Non-KANJI	81.77	72.13	73.71	81.29	75.04	83.10	75.46	82.65	76.69	68.71	67.98	57.73	57.02
	ALL	86.32	76.76	77.89	86.06	78.62	87.20	79.08	86.95	79.76	76.13	75.73	69.59	69.21
5th	KANJI	97.84	94.11	94.58	97.82	94.64	98.02	95.06	98.01	95.13	94.75	94.73	92.87	92.91
	Non-KANJI	94.45	87.18	89.30	94.24	90.11	95.96	92.10	95.84	92.65	87.41	86.54	70.75	70.40
	ALL	96.05	90.45	91.79	95.93	92.25	96.93	93.50	96.86	93.82	90.87	90.40	81.19	81.02
10th	KANJI	98.78	96.20	96.68	98.77	96.72	98.82	96.97	98.81	97.02	96.59	96.58	94.51	94.57
	Non-KANJI	96.75	90.45	92.72	96.62	93.27	97.92	95.33	97.88	95.63	92.67	91.94	72.27	72.20
	ALL	97.71	93.17	94.59	97.64	94.90	98.35	96.10	98.32	96.29	94.52	94.13	82.76	82.75
20th	KANJI	99.30	97.47	97.90	99.29	97.94	99.27	98.12	99.27	98.14	97.77	97.75	95.52	95.60
	Non-KANJI	98.04	92.58	94.99	97.96	95.36	98.95	97.23	98.93	97.39	95.76	95.38	72.64	72.73
	ALL	98.63	94.89	96.36	98.59	96.57	99.10	97.65	99.09	97.75	96.71	96.50	83.44	83.52
50th	KANJI	99.65	98.32	98.76	99.65	98.77	99.54	98.80	99.54	98.83	98.77	98.76	96.26	96.35
	Non-KANJI	99.03	94.35	97.05	99.00	97.24	99.54	98.63	99.54	98.71	98.13	98.09	72.75	72.91
	ALL	99.32	96.22	97.85	99.31	97.96	99.54	98.71	99.54	98.77	98.43	98.40	83.84	83.97
100th	KANJI	99.74	98.57	99.02	99.74	99.03	99.59	98.95	99.59	98.98	99.06	99.05	96.47	96.56
	Non-KANJI	99.53	95.18	98.44	99.52	98.55	99.72	99.15	99.72	99.21	99.22	99.26	72.77	72.95
	ALL	99.63	96.78	98.71	99.63	98.78	99.66	99.06	99.66	99.10	99.15	99.16	83.95	84.09

Table A2-8 Recognition rates

Data sets	ODD numberd 41 data sets (Unknown)			Even numberd 40 data sets (Learning)		
Character kinds	KANJI	Non-KANJI	ALL	KANJI	Non-KANJI	ALL
Recognition rates	91.79	83.10	87.20	93.01	82.62	87.52

(Weights of directional features, direction-change features and written-area features= 1.0, 1.0 and 15.0) (rate:%)

$$\left(Recognition\ rate = \frac{characters\ recognized\ correctly}{all\ handwritten\ data\ count} \right)$$

Table A2-9 Averages of character’s recognition rates

Data sets	ODD numberd 41 data sets (Unknown)			Even numberd 40 data sets (Learning)		
Character kinds	KANJI	Non-KANJI	ALL	KANJI	Non-KANJI	ALL
Average recognition rates	92.75	81.20	91.44	94.86	82.76	93.49

(Weights of directional features, direction-change features and written-area features= 1.0, 1.0 and 15.0) (rate:%)

$$\left(Average\ recognition\ rate = \frac{\sum each\ character's\ recognition\ rate}{Character\ kinds\ count} \right)$$

Acknowledgments

Acknowledgments

The author is greatly indebted to Prof. Kazuhiko Yamamoto, who gave numerous helpful suggestions and guidance to the author.

The author would like to thank Mr. Yuuichi Hoshino of CSK CORPORATION for his cooperation in character recognition experiments while he was a student at Gifu University, Dr. Kiyoshi Sawada of the University of Marketing and Distribution Sciences, Mr. Hideto Yamamoto of SANYO Information Business Co., Ltd. for their cooperation in research on string separation, Dr. Masaki Nakagawa of Tokyo University of Agriculture and Technology for giving his on-line handwritten database to the author.

The author wishes to acknowledge Dr. Fusao Terada, director of the R&D Headquarters, and Mr. Osamu Suzuki, general manager of the Hypermedia Research Center, SANYO Electric Co., Ltd., providing an opportunity for research to the author. The author wishes to acknowledge Mr. Shoji Takeda, senior manger of SANYO Information Business Co., Ltd., Mr. Yoshikazu Tomida, senior manager, Mr. Hiroshi Horii, manager, and Mr. Kanji Toyokura, manager of the Hypermedia Research Center, SANYO Electric Co., Ltd., giving many advises to the author. The author wishes to acknowledge the encouragement of Mr. Noriyuki Mori, general manager of the Information & Telecommunications Development Center, SANYO Electric Co., Ltd., Mr. Tosiharu Tanaka, senior manger, Mr. Yukiteru Nisida, senior manger, Mr. Tomohiisa Yosikawa, manager and Mr. Masashi Ochiiwa, manager of SANYO Information Business Co., Ltd.. The author thanks Mr. Takatoshi Yoshikawa, manager, Mr. Akira Nakamura, chief researcher, Dr. Mamoru Yasumoto, chief researcher, Mr. Hitoshi Hongo, chief researcher, and other members of the Human Interface Laboratory of the Hypermedia Research Center, SANYO Electric Co., Ltd. and members of the Yamamoto Laboratory of GIFU University for their cooperation and encouragement.

The author is deeply grateful Mr. Isamu Kameda, ex-president of SANYO Information Business Co., Ltd., for his guidance to the author for many years.

References

References

[1] A.Noito, M.Nakagawa: “A Method of Compression and Restoration of Digital-Ink”, *SIG note*, IPS Japan,96,94,pp.9-16, 1996.

[2] I.Guyon, L.Schomaker, R.Plamondon, M.Liberman, S.Janet: “UNIPEN project of on-line data exchange and recognizer benchmark”, *Proc. 12th ICPR*, pp.20-23, October 1994.

[3] M.Nakagawa, T.Oguni, T.Yoshino: “Human Interface and Application on IdeaBoard”, *Proc.IFIP TC13 Int’l Conf. On Human-Computer Interaction*, pp.501-508, July1997.

[4] T.Yamaguchi, K.Watanabe, Y.Okazaki, H.Kondo, M.Okamoto, et al.: “An Implementation of the Pen Based Interface for ITS for Guiding Fraction Calculation”, *Proc. ICCE97*, pp.504-511, December 1997.

[5] R.Plamondon, J.J.Brault, P.N.Robillard: “Optimizing the Design of an Accelerometric pen for Signature Verification”, *Proc. Crime Countermeasures and Security*, pp.35-40, 1983.

[6] L.Claesen, D.Beullens, R.Martens, R.Mertens, S.De Schrijver, W.de Jong: “SmartPen: An Application of Integrated Microsystem and Embedded Hardware/Software CoDesign”, *ED&RC’96 User Forum*, pp.201-205, 1996.

[7] M.Nakagawa: “Non-keyboard Input of Japanese Text On-line Recognition of Handwritten Characters as the Most Hopeful Approach”, *Journal of Information Processing*, vol.13,no.1, 1990,pp.15-34.

[8] N.Babaguchi, M.Tsukamoto, T.Aibara: “A Fundamental Study on Character Segmentation from Handwritten Japanese Character Strings”, *IEICE Trans. (D)*,vol.J68-D,no.12, 1985,pp.2123-2131. *(in Japanese)*

[9] H.Murase, T.Wakahara, M.Umeda: “Online Writing-Box Free Character String Recognition by Candidate Character Lattice Method”, *IEICE Trans.(D)*,vol.J68-D,no.4, 1985,pp.765-772.

[10] H.Murase, M.Sinya, T.Wakahara, K.Odaka: “Segmentation and Recognition of Hand-Witten Character String using Linguistic Information”, *IEICE Trans. (D)*,vol.J69-D,no.9, 1985,pp.1292-1301. *(in Japanese)*

[11] E.Anquetil, G.Lorette: “Perceputual Model of Hanwriting Drawing Application to the Handwriting Segmentation Problem”, *Proc. 4th ICDAR*, vol.1, Ulm,Germany, 1997, pp.112-117.

[12] J.J.Lee, J.H.Kim: “A Unified Network-based Approach for Online Recognition of Multi-Lingual Cursive Handwritings, *Proc. 5th IWFHR*, Colchester, England,1996, pp.393-397.

[13] K.Toyokawa, K.Kitamura, S.Katoh, H.Kaneko, N.Itoh, M.Fujita: “An On-Line Character Recognition System for Effective Japanese Input”, *Proc. 2nd ICDAR*, vol.1, Tsukuba Science,Japan,1993, pp.208-213.

[14] H.Murase: "Writer Adaptive Character Segmentation Method for Free Format Handwriting", *IEICE Trans.(D)*,vol.J72-D-2,no.1, 1989,pp.132-139. *(in Japanese)*

[15] K.Nakabayashi, T.Kitamura, T.Kawaoka: "A High-Speed Recognition Method for Handwritten Character String Using Inexact Word Matching", *IEICE Trans. (D)*,vol.J74-D-2,no.11, 1991,pp.1528-1537. *(in Japanese)*

[16] A.Suzuki, S.Miyahara: "Recognition Method Handling Character Position Gaps in Handwritten Japanese Addresses", *IEICE Trans. (D)*,vol.J77-D-2,no.1, 1994,pp.20-28. *(in Japanese)*

[17] M.Okamoto, H.Yamamoto, T.Yosikawa, H.Horii, K.Toyokura, K.Yamamoto: "Theme on On-line Handwriting Character Recognition Method and String Separation Method using Language Processing", *Proc. 5th IWFHR*, Colchester, England, 1996,pp.165-170.

[18] M.Okamoto, H.Yamamoto, K.Sawada, K.Yamamoto: "On-Line Handwriting Character String Separation Method Using Network Expression", *Proc. 13th ICPR*, vol.4, track D,Vienna,Austria,1996, pp.422-425.

[19] D.Shier: "On algorithm for finding the K shortest paths in a network", *Networks*, vol.9, 1979, pp.195-214.

[20] R.Bellman, R.Kalaba: "On kth Best Policies", *J.Soc. Indust. Appl. Math.*, vol.8,no.4, 1960,pp.582-588.

[21] M.Okamoto, K.Yamamoto: "On-line Handwriting Character Recognition Method with Directional Features and Direction-Change Features", *Proc. 4th ICDAR*, vol.2, Ulm,Germany,1997,pp.926-930.

[22] M.Nakagawa, T.Higashiyama, Y.Yamanaka, S.Sawada, L.Higashigawa, K.Akiyama: "On-line Handwritten Character Pattern Database Sampled in a Sequence of Sentences without Any Writing Instructions", *Proc. 4th ICDAR*, vol.1, Ulm,Germany, 1997,pp.376-381.

[23] M.Nagao: "Natural Language Processing", *The Iwanami Software Science Series*, no.15, 1996, pp.117-137. *(in Japanese)*

[24] M.Shinya, M.Umeda: "Evaluation of Compound Post-Processing Method in Character Recognition", *IEICE Trans. (D)*, vol.J68-D,no.5,pp.1118-1124. *(in Japanese)*

[25] T.Sugimura, T.Saito: "A study of Reject Correction for Character Recognition Based on Binary n-Gram", *IEICE Trans. (D)*, vol.J68-D,no.1, 1985,pp.64-71. *(in Japanese)*

[26] F.Nishino: "Natural Language Processing in Text Recognition", *Information Processing Society of Japan Trans.* vol.34,no.10, 1993,pp.1274-1280. *(in Japanese)*

[27] T.Takao, F.Nishino: "Implementation and Evaluation of Post-processing for Japanese Document Readers", *Information Processing Society of Japan Trans.* vol.30,no.11, 1989,pp.1394-1401. *(in Japanese)*

[28] K.Ikeda,Y.Ohta,E.Ueno: “Vocabular and Contextual Postprocessing for the Recognition of Handprinted Japanese Manuscript”, *Information Processing Society of Japan Trans.*, vol.26,no.5, 1985,pp.862-869. *(in Japanese)*

[29] H.Saito, H.Nogami: “Natural Language Processing for Japanese Word Processor”, *Information Processing Society of Japan Trans.*,vol.34,no.10, 1993,pp.1241-1248. *(in Japanese)*

[30] N.Itoh, H.Maruyama: “A Method of Detecting and Correcting Errors in the Results of Japanese OCR”, *Information Processing Society of Japan Trans.* vol.33,no.5, 1992,pp.664-670. *(in Japanese)*

[31] T.Ikeda: “Syntactic and Semantic Analysis of Japanese Sentences by Lexico-Grammatical Method”, *Information Processing Society of Japan Trans.*,vol.26,no6, 1985, pp.1079-1088. *(in Japanese)*

[32] T.Kawada, S.Amano, K.Sakai: “Linguistic error correction of Japanese sentences”, *COLING 80,pp.257-261,1980.*

[33] E.M.Riseman, R.W.Ehrich: “Contextual Word Recognition Using Binary Diagrams”, *IEEE Trans. Comput.*, vol.C-20, no.4,1971,pp.397-403.

[34] C.Tappert, Y.Suen and T.Wakahara: “The State of the Art in On-Line Handwriting Recognition”, *IEEE Transaction of Pattern Analysis and Machine Intelligence*, vol.12, no.8, pp.787-808, August 1990.

[35] M.Nakagawa, T.Manabe, K.Aoki, Y.Ikeda, N.Takahashi: “On-line Handwritten Character Recognition as a Japanese Input Method”, *Proc. ICTP’83*, pp.191-196,1983.

[36] M.Nakagawa, K.Akiyama, Le Van Tu, A.Homma and T.Higashiyama: “Robust and Highly Customizable Recognition of On-Line Handwritten Japanese Characters”, *Proc. 13th ICPR*, vol.3, track C, pp.269-273, August 1996.

[37] K.Takahashi, H.Yasuda, T.Matsumoto: “A Fast HMM Algorithm for On-line Handwritten Character Recognition”, *Proc. 4th ICDAR*,vol.1,pp.369-375,August 1997.

[38] M.Hamanaka, K.Yamada and J.Tsukumo: “On-Line Character Recognition Experiments by an Off-Line Method Based on Normalization-cooperated Feature Extraction”, *Proc. 2nd Int. Conf. on Document Analysis and Recognition*, pp.204-207, October 1993.

[39] M.Hamanaka, K.Yamada and J.Tsukumo: “On-Line Japanese Character Recognition Based on Flexible Pattern Matching Method Using Normalization-Cooperative Feature Extraction”, *IEICE Trans.inf.&SYST.*, vol.E77-D, no.7, 825-831, July 1994.

[40] M.Yasuda and H.Fujisawa: “An improvement of Correlation Method for Character Recognition”, *IEICE Trans.*, vol.J62-D,no.3,pp.217-224, March 1979.

[41] T.Saito, H.Yamada, K.Yamamoto: “On the Data Base ETL9 of Handprinted Characters in JIS Chinese Characters and Its Analysis”, *IEICE trans.*, vol.68-D, no.4, pp.757-764, April 1985.

[42] T.Saito, H.Yamada, K.Yamamoto: "An Analysis of Handprinted Characters by Directional Pattern Matching Approach", *IEICE Trans.*, vol.J65-D, no.5, pp.550-557, May 1982.

[43] T.Wakahara, N.Nakajima, S.Miyahara and K.Odaka: "On-Line Cursive Kanji Character Recognition Using Stroke-Based Affine Transformation", *Proc. 13th ICPR*, vol.3, track C, pp.204-209, August 1996.

[44] K.Yoshida and H.Saeo, "Online Handwritten Character Recognition for a Personal Computer System", *IEEE trans., Consumer Electronics*, vol.CE-28, no.3, pp.202-208,1982.

[45] Y.Liu, L.Zhang, J.Tai: "A New Approach to On-line Handwritten Chinese Character Recognition",*2nd ICDAR*,pp.192-195,October 1993.

[46] A.Leroy: "Progressive lexicon reduction for on-line handwriting",*5th IWFHR*,pp.399-404,September 1996.

[47] O.Kwon, M.Kim, M.Park, Y.Kwon: "A Cursive On-line Hangul Recognition System Based on the Combination of Line Segments", *2nd ICDAR*,pp.200-203, October 1993.

[48] C.Kim, K.Park, B.Jun, J.Kim: "Substroke Matching by Segmenting and Merging for On-Line Korean Cursive Character Recognition", *14th ICPR*,vol.2,pp.1110-1113, Ausust 1998.

[49] H.Yamada, T.Saito and K.Yamamoto: "Line Density Equalization – A Nonlinear Normalization for Correlation Method –",*IEICE Trans.*, vol.J67-D, no.11, pp.1379-1383, November 1984.

[50] H.Yamada, K.Yamamoto and T.Saito: "A nonlinear normalization method for handprinted Kanji character recognition – line density equalization", *Pattern Recognition*, vol.23, no.9, pp.1023-1029, 1990.

[51] M.Nakagawa, T.Higashiyama, Y.Yamanaka, S.Sawada, T.Le and K.Akiyama: "Collection and utilization of on-line handwritten character patterns sampled in a sequence of sentences without any writing instructions", *IEICE Technical Report*, PRU95-110, vol.95, no.278, pp.43-48, September 1995.

Research achievements

Research achievements

Journals

1. “On-line Handwriting Character Recognition using Direction-Change Features that Consider Imaginary strokes”, The Journal of the Pattern Recognition Society, 1999,
   M.Okamoto and K.Yamamoto （本件採録決定済み：掲載号未定：出版待ち）

2. ”On-line Handwriting Character Recognition Method using Directional and Direction-Change Features”, International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), 1999,
   M.Okamoto and K.Yamamoto （本件採録決定済み：掲載号未定）

3. ”On-line Handwriting Character Recognition using Directional Features and Direction-Change Features”, The Transactions of The Institute of Electrical Engineers of Japan, C, 1999, *(in Japanese)* [電気学会論文誌 Ｃ]
   M.Okamoto and K.Yamamoto （本件採録決定済み：平成１１年３月号予定）

4. ”Unified Method of Physical Features, Character Recognition and Language Processing for On-line Handwritten Character String Separation”, Journal of Computer Processing of Oriental Languages, vol.12,No.2, 1999,
   M.Okamoto, K.Sawada and K.Yamamoto
   （本件採録決定済み：出版待ち）

International Conferences

1. ”Theme on On-line Handwriting Character Recognition Method and String Separation Method using Language Processing”, Proc.5th IWFHR (International Workshop on Frontiers in Handwriting Recognition), Colchester, England, pp.165-170,September 1996,
M.Okamoto, H.Yamamoto, T.Yosikawa, H.Horii, K.Toyokura and K.Yamamoto

2. “On-Line Handwriting Character String Separation Method Using Network Expression”, Proc. 13th ICPR (International Conference on Pattern Recognition), vol.4, track D,Vienna,Austria, pp.422-425, August 1996,
M.Okamoto, H.Yamamoto, K.Sawada and K.Yamamoto

3. ”On-line Handwriting Character Recognition Method with Directional Features and Direction-Change Features”, Proc. 4th ICDAR (International Conference on Document Analysis and Recognition), vol.2, Ulm,Germany, pp.926-930, August 1997,
M.Okamoto and K.Yamamoto

4. “Direction-Change Features of Imaginary Strokes for On-line Handwriting Character Recognition”, 14th ICPR (International Conference on Pattern Recognition), vol.2, Brisbane,Australia, pp.1747-1751, August 1998,
M.Okamoto, A.Nakamura and K.Yamamoto

5. “The Balance of Directional, Direction-Change and Written-area Features for On-line Character Recognition”, 3rd DAS (International Association for Pattern Recognition Workshop on Document Analysis Systems), Nagano, Japan, pp.141-144,November 1998,
M.Okamoto and K.Yamamoto