Res. Rep. Fac. Eng. Mie Univ., Vol. 17, pp. 53-64 (1992)

Original Paper

Definitions of Line Pattern and Thinning Algorithms for Digital Pictures

Shinji TSURUOKA, Fumitaka KIMURA+, Yasuji MIYAKE+
and Shigeki YOKOI*
(Department of Electrical and Electronic Engineering)
(+Department of Information Engineering)

(Received September 16, 1992)

The explicit definitions of two types of line patterns are introduced using the connectivity number and the number of 1-pixels in the 8-neighbor, and then parallel and sequential thinning algorithms for binary and grey-level images are proposed. Thinned images are one pixel thick and topologically equivalent to the original images.

By a character recognition experiment, these thinning algorithms obtain the better result than other two representative algorithms in the form of the result images, and the processing time and the recognition rate.

Key word: Thinning, Image Processing, Pattern Recognition, Character Recognition, OCR, Skeleton, Erosion, Computer Vision

1. Introduction

Thinning algorithm in the digital picture has played important roles both theoretically and practically in image processing and computer vision system. It have been widely used for feature extraction, data compression, shape description and so on in character recognition, image processing, 3-Dimentional recognition on computer vision.

The thinning algorithm of a continuous binary image was first introduced by Blum[1] in 1964. Rosenfeld[2,3,4] pointed out the relation between the thinning algorithm and the connectivity in digital pictures. Yokoi[5] proposed the connectivity numbers in digital pictures and the sequential thinning algorithms based on them. Tamura[6] proposed the parallel thinning algorithms using local operation masks.

But, thinning algorithm have some problems in both the theory and the application on character recognition, image processing, computer vision and so on. The first problem is that

* Department of Information Engineering, Faculty of Engineering, Nagoya University

the definition of the result picture (the line pattern) is ambiguous in the set theory. The second problem is the form of the result picture. There are serious three following problems in the character recognition.

(1) The shrink and the disappearance of the character line in the result picture occur frequently.

(2) The noise branch is appeared by the little noise in the border of the input picture.

(3) One cross section in the input picture becomes two branch points in the result picture.

In this paper, we define the line pattern, which is the result picture of the thinning algorithm, using the set theory explicitly. We propose the new parallel and sequential thinning algorithms in binary and grey level pictures.

2. Basic Definitions and Fundamental Properties

We start with some elementary definitions.

[Definition 1] (Picture) A digital picture is a finite rectangular array of "points". Any "point" in the array is specified by a pair of numbers i, j ($1 \le i \le m$, $1 \le j \le n$), and each point is called a pixel (or picture element). The pixel (i,j) has one of the "grey-values". A picture having the value fij at the pixel (i,j) = x is denoted by F = { fij }. Especially, the picture in which grey values of pixels are 0 or 1 is called a binary picture.

[Definition 2] (Neighbor) The neighbors of a pixel {xk} are identified by the eight directions shown in Fig.1. The four pixels {x1,x3,x5,x7} in the odd directions is called 4-neighbors and the eight pixels {x1,x2,...,x8} is called 8-neighbor.

[Definition 3](Connectivity Number [5]) Let B ={bij} = {b(x)} be binary picture. Two-types of connectivity number are considered here.

(1)Connectivity Number for 4-connectivity

$$N_C^{(4)}(x) = \sum_{k=1}^{4} b(x_{2k-1})\{1 - b(x_{2k})b(x_{2k+1})\}$$

(2)Connectivity Number for 8-connectivity

$$N_C^{(8)}(x) = \sum_{k=1}^{4} \overline{b(x_{2k-1})}\{1 - \overline{b(x_{2k})}\;\overline{b(x_{2k+1})}\}$$

	$j-1$	j	$j+1$
$i-1$	x_4	x_3	x_2
i	x_5	x	x_1
$i+1$	x_6	x_7	x_8

Fig.1 Neighborhood xk of x.

[Property] If the pixel where the connectivity number is 1 would be deleted, connectivity of the picture does not change.

[Definition 4](Operation Form) Two types of the operation form are introduced. Let $F^{(m)}=\{f_{ij}^{(m)}\}$ be the m-th processed picture, and let $F^{(0)} = \{ f_{ij}^{(0)} \}$ be the initial picture. The parallel operator uses $F^{(m-1)}$, but the sequential operator uses both $F^{(m-1)}$ on the unprocessed neighbor and $F^{(m)}$ on the processed neighbor.

3. Definitions of Line Pattern and Thinning Algorithms

3.1 Definitions of line pattern

We propose the definitions of line pattern based on the set theory. Intuitively we can think that the line pattern is the picture that the width of any line is one dot and that it does not include the 2x2 dots subpicture (Fig.2). But this definition have the serious problem for the thinning algorithm. That is, the windmill-like picture for 4-connectivity

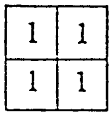


Fig.2 2×2 block of 1-pixels.

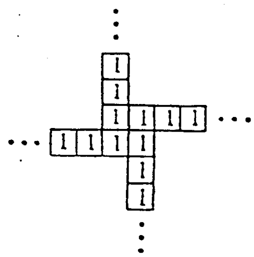


Fig.3 A windmill-like picture.

(Fig.3) is the line pattern. Because if you would delete any pixel, the connectivity of the picture would be changed and you could not transform it into the line pattern.

[Definition 1](Line pattern of 4-connectivity)

We define the line pattern of 4-connectivity as the pattern where the all pixels satisfy one of the following conditions.

(1) $N_C^{(4)}(x) \neq 1$

(2) $S^{(8)}(x) \equiv \sum_{k=1}^{8} b(x_k) \leq 2$

This definition includes the Fig.3 and we think that this definition agrees with the intuitive line pattern.

[Definition 2](Line pattern of 8-connectivity)

We define the line pattern of 8-connectivity as the pattern where the all pixels satisfy one of the following conditions.

(1) $N_C^{(8)}(x) \neq 1$

(2) $S^{(8)}(x) \leq 1$

Where condition (1) of these definitions shows the twice border points, and this condition preserves the connectivity of the input pattern. Condition (2) of these definitions shows the preservation of the end points. We illustrate the examples of the line pattern in Fig.4,5.

3.2 Thinning algorithms for binary pattern

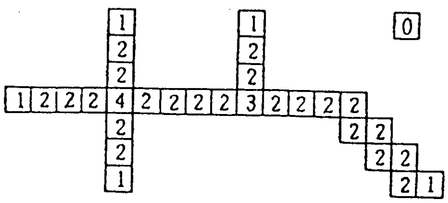


Fig.4 An example of 4-connected line pattern. (Number is the Connectivity Number)

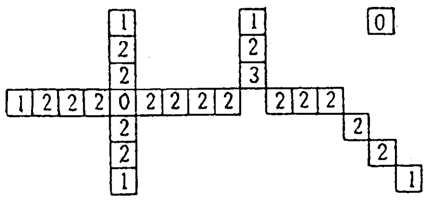


Fig.5 A example of 8-connected line pattern. (Number is the Connectivity Number)

We proposed the thinning algorithms based on the previous definition of line pattern.

[Algorithm 1] (Sequential Thinning Algorithm for Binary Pattern - STA)

If no 1-pixel delete over the frame scan, this algorithm will stop. The scan of this algorithm starts at the top left corner like a TV scan, it ends at the bottom right corner (fig.6). If 1-pixel satisfy the following every conditions, the pixel will be deleted (be changed to 0-pixel).

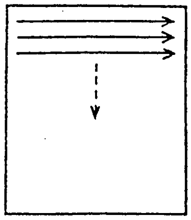


Fig.6 Way of scanning the picture.

[1] Deletable conditions for 4-connectivity

(Condition 1) Condition for the border points

$$S^{(4)}(x) \equiv \sum_{k=1}^{4} b(x_{2k-1})^{(m-1)} < 4$$

(Condition 2) Condition for the preservation of the connectivity

$$N_C^{(4)}(x) = \sum_{k=1}^{4} b(x_{2k-1})^{(n)}\{1 - b(x_{2k})^{(n)} b(x_{2k+1})^{(n)}\} = 1$$

where,

$$n = \begin{cases} m & (x_2, x_3, x_4, x_5 : \text{processed neighbor}) \\ m-1 & (x_1, x_6, x_7, x_8 : \text{unprocessed neighbor}) \end{cases}$$

(Condition 3) Condition for the preservation of the end points

The 1-pixel satisfies the following condition (a) or (b)

(a) $S^{(8)}(x) \equiv \sum_{k \in S} b(x_k)^{(n)} > 2$

(b) The 1-pixel satisfies one of the masks (a) ~ (d) in Fig. 7.

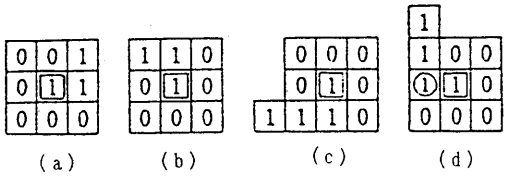


Fig.7 Masks for skeletonizing binary patterns.
(In the case (d), the neighbor pixel ① is also eliminated.)

[2] Deletable conditions for 8-connectivity

(Condition 1) Condition for the border points

$S^{(4)}(x) < 4$

(Condition 2) Condition for the preservation of the connectivity

$$N_C^{(8)}(x) \equiv \sum_{k=1}^{4} \overline{b(x_{2k-1})^{(n)}}\{1 - \overline{b(x_{2k})^{(n)}}\,\overline{b(x_{2k+1})^{(n)}}\} = 1$$

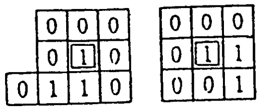
(Condition 3) Condition for the preservation of the end points

The 1-pixel satisfy the following condition (a) and (b)

(a) $S^{(8)}(x) > 1$

(b) The 1-pixel doesn't satisfy the masks (a) and (b) in Fig. 8.

[Condition 1] is the condition for the location of the result picture. [Condition 2] is the condition for the preservation of the connectivity of the input picture. [Condition 3] is the condition for the preservation of the end points , the masks aren't symmetry.



(a) (b)

Fig. 8 Masks for preventing exessive eliminations of 1-pixels.

[Algorithm 2] (Parallel Thinning Algorithm for Binary Pattern - PTA)

Parallel Algorithm processes all pixel on the picture concurrently. To prevent the disappearance of the 2 bits width line, the thinning process are divided into 4-subcycles. First subcycle (m1) processes the 1-pixel with the neighbor x_3 which is 0-pixel. Second, third, fourth subcycle (m2,m3,m4) processes the 1-pixel with the neighbor x_7, x_5, x_1, respectively. If 1-pixels satisfy the following every conditions, the pixels will be deleted (be changed to 0-pixel) concurrently.

[1] Deletable conditions for 4-connectivity

(Condition 1) Condition for the border points

$$S^{(4)}(x) \equiv \sum_{k=1}^{4} b(x_{2k-1})^{(m-1)} < 4$$

(Condition 2) Condition for the preservation of the connectivity

$$N_C^{(4p)}(x) = \sum_{k=1}^{4} b(x_{2k-1})^{(mc-1)}\{1 - b(x_{2k})^{(mc-1)} b(x_{2k+1})^{(mc-1)}\} = 1$$

where,

$$c \in \{1,2,3,4\}$$

(Condition 3) Condition for the preservation of the end points

$$S^{(8p)}(x) \equiv \sum_{k=1}^{8} b(x_k)^{(mc-1)} > 2$$

[2] Deletable conditions for 8-connectivity

(Condition 1) Condition for the border points

$$S^{(4)}(x) < 4$$

(Condition 2) Condition for the preservation of the connectivity

$$N_C^{(8p)}(x) \equiv \sum_{k=1}^{4} \overline{b(x_{2k-1})^{(mc-1)}}\{1 - \overline{b(x_{2k})^{(mc-1)}}\ \overline{b(x_{2k+1})^{(mc-1)}}\} = 1$$

(Condition 3) Condition for the preservation of the end points

$$S^{(8p)}(x) > 1$$

This algorithm doesn't have the asymmetry, so it doesn't use the asymmetric logical masks.

3.3 Thinning algorithms for grey-level pattern

In the binary thinning algorithm, the distortions of the result picture at the branch points or cross points appear. To prevent these distortion, we propose the algorithm using the grey-level information at the pixel in this section.

[Algorithm 3] (Sequential Thinning Algorithm for Grey-level Pattern - STAG)

(STEP 1)[Preparation] Binary picture B={b(x)} is made from the input grey-level picture using the threshold level. This step separates the " Object" from "Background" in the input picture. This thinning algorithm processes the area of "Object".

(STEP 2)[Thinning process] Thinning algorithm processes the following (sub step) about the grey-level $T = T_1, T_2, \ldots, T_n$ ($T_1 < T_2 < \ldots < T_n$, T_n is the maximum of the grey-level in the input picture), sequentially.

(Sub step) Thinning algorithm processes the following operation about the number of the scan $m = 1,2,\ldots$, sequentially. The algorithm stops if the processed picture equals the preprocessed picture.

The scan starts at the top left corner like a TV scan, and it ends at the bottom right corner. It finds the pixel where f(x)=T, and check the deletable conditions in [Algorithm 1]. If the pixel satisfies the every deletable conditions, the pixel will be deleted, that is, the pixel will be changed into 0-pixel.

(STEP 3)[postprocess] The picture with the concave in the grey level (for example, Fig. 9) doesn't change into the line picture. So that, this step changes the inner points where the all neighbors are 1-pixels ($S^{(8)}(x) = 8$) or the local minimum points where the grey levels of all neighbors are greater or equal than the grey level of the pixel ($f(x_k) > f(x)$) into 0-pixels.

[Algorithm 4] (Parallel Thinning Algorithm for Grey-level Pattern - PTAG)

This algorithm is the same as the previous [Algorithm 3] except for the deletable conditions. The conditions is the same as the [Algorithm 2].

3.4 High speed process using the coordinates table

[Algorithm 3] and [Algorithm 4] take a lot of time for scanning the picture. To prevent this problem, we propose "the double table method of the coordinates". "The first table" store the x-y coordinates of the pixel where the grey-level is greater than the threshold from the input picture. "The second table" store the x-y coordinates of the pixel where the grey-level is T from "the first table". These algorithms use "the second table" in the thinning process, and don't scan the picture. This method is useful for the picture where the rate of 1-pixel is little, for example, the character picture.

```
3 3 3 3 3
3 2 2 2 3        1 1 1
3 2 2 2 3        1 1 1
3 2 2 2 3        1 1 1
3 3 3 3 3
```

(a):Input (b)Result in (STEP 2)

Fig.9 An example of thinned pattern.

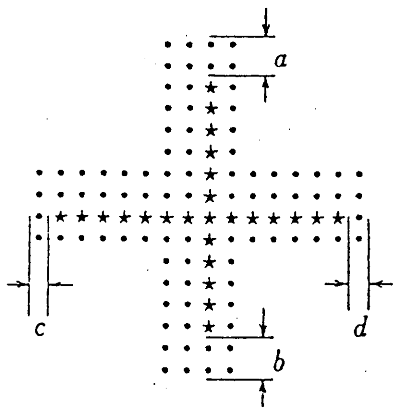


Fig.10 Cross figure of 4 bits width.

4. Some Experimental Results

4.1 Comparison with other binary thinning algorithm

In this section, we compare our algorithms (STA) with other representative thinning algorithms. We select two binary thinning algorithm using the study on the comparison of the line thinning algorithms[6]. First algorithm is the Yokoi's algorithm (YTA) [5] of the sequential process. Second algorithm is the Tamura's algorithm (TTA) [6] of the parallel process.

We compare the following items for each thinning algorithm (Table 1).

(1) The shrinkage of the line : we examine the sum of the 4-direction shrinkage from the border using the artificial cross picture (Fig.10,11). STA get the best result.

(2) The number of the superfluous branch & processing time : We examine the number of the superfluous branch and the processing time per character (FACOM230-38s, FORTRAN) for the 5 sets of Japanese character "Katakana" (46 categories ; total 230 characters) in the free handwritten database (ETL1) made by the Electrotechnical Laboratory in Japan and Fujitsu Ltd. TTA takes the best result on the number and STA takes the best result on the processing time. And using high speed process in the section 3.4 can reduce the processing time of STA to about 500ms.

(3) Evaluation of the form : In the thinning algorithm, we desire both the shrinkage and the number of the superfluous branch little. We propose the product of them as a new criterion. The quantity of YTA nearly equals to the STA and is half of the TTA.

4.2 Comparison with other grey level thinning algorithm

Table 1 Comparison of binary thinning algorithms

	Y T A	T T A	S T A
Number of shrinkage	26	66	22
Number of branch	43	40	58
The product of them	1118	2640	1276
Type of process	sequential	parallel	sequential
No. of subcycle	2	4	1
Process time [ms]	2056	1857	769
No. of image array	3	3	2

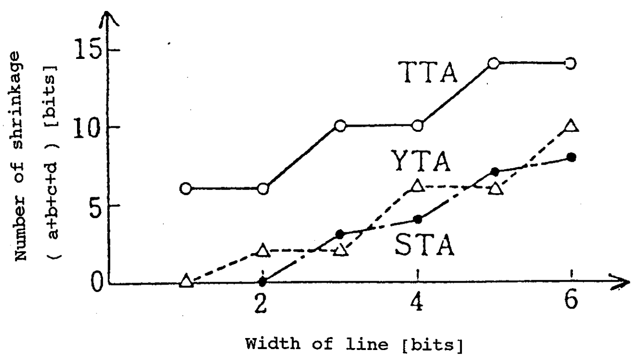


Fig.11 Shrinkages of skeleton in the three thinning algorithms.

We compare the our thinning algorithm in [Algorithm 3] with other thinning algorithms for the grey-level image. We select Hilditch's method (HIL)[7] and Shikano's method (WPM)[8], Yokoi's method (YTAG) [9],our thinning algorithm for binary image (STA). we don't use the postprocessing for every method.

(1) The location of the result picture : we examine the infinite picture toward up and down.

(a) The picture with the concave area (Fig.12) : The locations of the result picture for each algorithms are shown in Fig.13. STAG and YTAG extract the concave parts, and these methods can extract the blackout area in the character.

(b) The picture with the flat area (Fig.14) : The locations of the result picture for each algorithms are shown in Fig.15. YTAG extract the large area from the flat point to the maximum point, HIL and STAG extract the maximum point.

(c) The application to the character picture : The result pictures for a Japanese character "Katakana" in Handwritten character database (ETL1) are shown in Fig.16. WPM extract the noise branches and the mixed line of 4-connectivity and 8-connectivity. YTAG extract the 2-bits width line and the noise branches. STA extract the disordered branch and cross points. STAG extract the best result picture by using the grey-level of the pixel. We think that the grey-level information reflect the writing trace by the writer.

The processing times are 1454ms(WPM), 1301ms(YTAG), 758ms(STA), 690ms(STAG),respectively. STAG is faster than STA by using high speed process in the section 3.4.

		⋮			
3	5	3	3	5	2
3	5	3	3	5	2
3	5	3	3	5	2
3	5	3	3	5	2
3	5	3	3	5	2
3	5	3	3	5	2
3	5	3	3	5	2
3	5	3	3	5	2
		⋮			

Fig.12 Input picture with concave points.

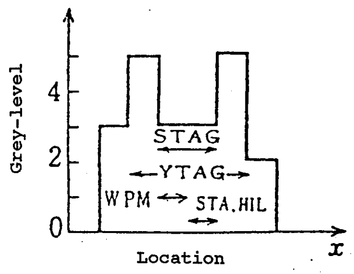


Fig.13 Grey-level and the locations of extracted line in Fig.12.

			⋮			
3	6	5	4	3	3	3
3	6	5	4	3	3	3
3	6	5	4	3	3	3
3	6	5	4	3	3	3
3	6	5	4	3	3	3
3	6	5	4	3	3	3
3	6	5	4	3	3	3
3	6	5	4	3	3	3
			⋮			

Fig.14 Input picture with flat points.

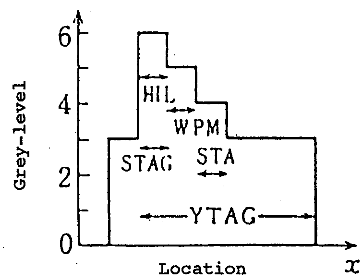


Fig.15 Grey-level and the locations of extracted line in Fig.14.

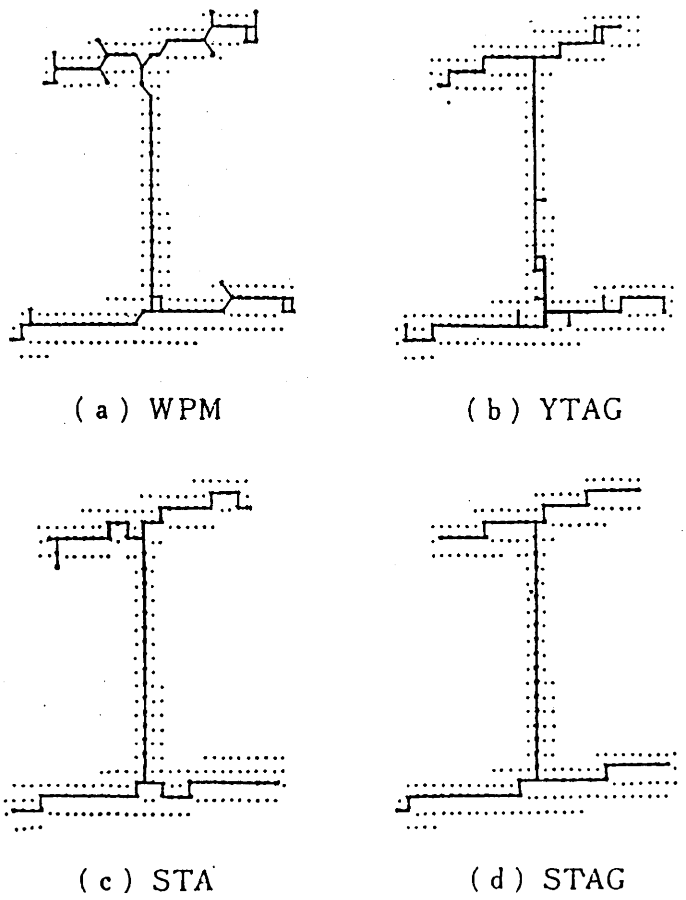


Fig.16 Results of handwritten character.

4.3 Application to character recognition experiment

We describe the experimental result of the character recognition method " Undeterministic stroke structure analysis [10]". The flowchart of this method is shown in Fig.17. This method include the binarization by the threshold and thinning, linear approximation, underterministic stroke extraction, Mahalanobis' distance for discriminant function. The recognition results of TTA and STA for the unknown 11,500 characters of Japanese character "Katakana" (46 categories, 250 characters/category) are shown in Table 2.

The examples for TTA and STA are shown in Fig.19. Fig.19 (a) and (b) show the correct examples for STA. Fig.18 (c) and (d) show the reject examples for STA. They are caused by

Table 2 Recognition rates for free handwritten "Katakane" characters (%)

	T T A	S T A
Correct	94.6	95.6
Reject	2.7	2.2
Error	2.7	2.2

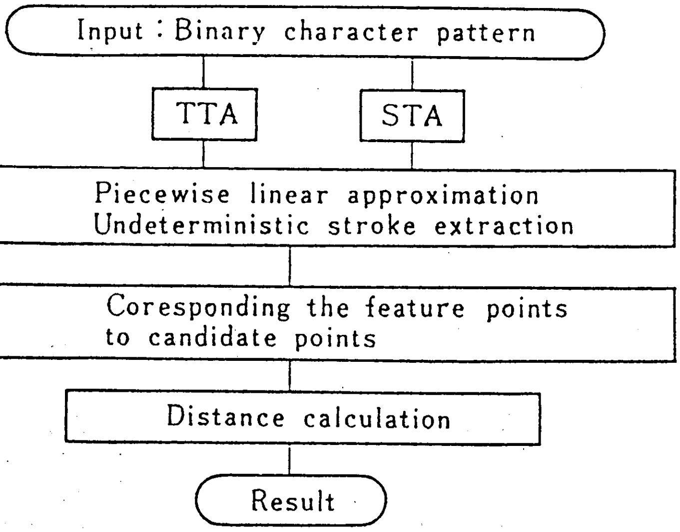


Fig.17 Flow chart of experiment for the character recognition.

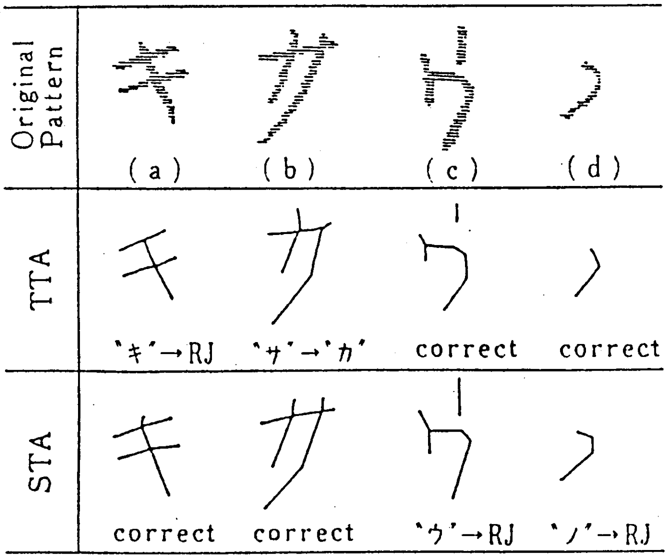


Fig.18 Examples of correct and reject characters.

less shrinkage.

The effect using the grey-level show in Fig.19. The distortion of the line pattern is less than the binary picture. We think that the grey level is useful for the character recognition.

5. Conclusion

The explicit definitions of two types of line patterns are introduced using the connectivity number and the number of 1-pixels in the 8-neighbor, and then parallel and sequential thinning algorithms for binary and grey-level images are proposed. Thinned images are one pixel thick and topologically equivalent to the original images.

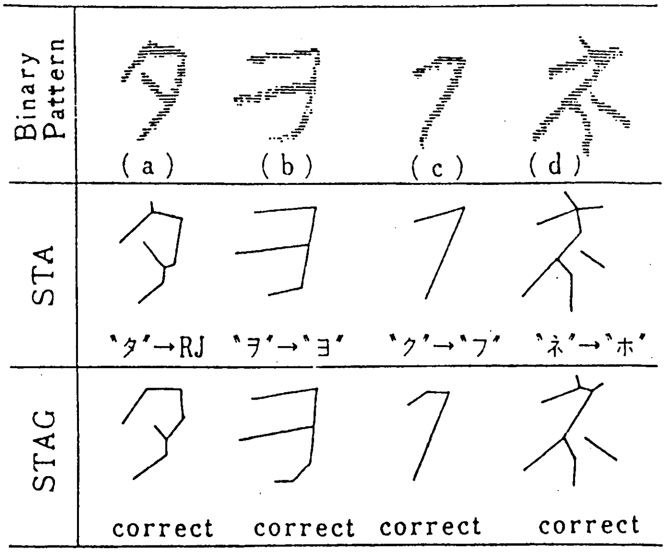


Fig.19 Examples of improved characters.

By a character recognition experiment, these thinning algorithms obtain the better result than other two representative algorithms in the form of the result images, and the processing time and the recognition rate.

Acknowledgement

We thank Prof. Mitsu Yoshimura (Chubu University, Japan) for helpful discussion and Prof. Malayappan Shridhar (The University of Michigan - Dearborn, U.S.A) for useful suggestion.

References

1) H.Blum : A Transformation for Extracting New Descriptiors of Shape, in Proc. Symp. Models for the Perception of Speech and Visual Form. MIT Press Cambrige, MA (1964).

2) A.Rosenfeld and J.L.Pfaltz : Sequential Operations in Digital Picture Processing, J.Assoc. Compt. Mach., 13,4, pp.471-494 (1966).

3) A.Rosenfeld : Connectivity in digital pictures, J. Assoc. Compt. Mach.,17,1, pp.146-160 (1970).

4) R.Steffanelli and A.Rosenfeld : Some parallel thinning algorithms for digital pictures, J. Asoc. Compt. Mach.,18,2, pp.255-264 (1971).

5) S.Yokoi, J. Toriwaki and T.Fukumura : An analysis of topological properties of digitized binary pictures using local features, Computer Graphics and Image Processing, 4,5, pp.63-73 (1975).

6) H.Tamura : A comparison of the line thinning algorithms from digital geometry viewpoint, in Proc. 4th Int. Joint Conf. Pattern Recogn., pp.715-719 (1978).

7) C.J.Hilditch : Linear skeletons from square cupboards, (B.Meltzer and D.Michie ed) : Machine Intelligence 4, Edinburgh Univ. Press, PP403-420(1969).

8) K.Shikano, J.Toriwaki, T.Fukumura : Wave propagation method - a method for conversion of grey pictures into line figures, Trans. IECE(D), J55-D, 10, pp.668-

675(1972).

9) S.Yokoi, J.Toriwaki, T.Fukumura : A method for extracting feature points and line figures from grey pictures, Trans.IECE(D), J58-D, 10, pp.601-608(1975).

10) F.Kimura, M.Yoshimura, Y.Miyake, M.Ichikawa : Unconstrainedly handprinted "Katakana" character recognition by a stroke structure analysis method, Trans. IECE(D), J62-D, 1, pp.16-23(1979).