1998 International Computer Symposium
Workshop on Image Processing and Character Recognition
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Corner Preserved One-Pass Parallel Thinning Algorithm with the Aids of Condensed Image Plane

Rei-Yao Wu
Information Management Department
Shih Hsin University, Taipei, Taiwan, R.O.C.
E-mail: rywu@cc.shu.edu.tw

Abstract

A* new one-pass parallel thinning algorithm is proposed in this paper. With the aids of the condensed image plane, the proposed algorithm obtains skeletons with the properties of perfectly 8-connected, noise-insensitive, and topologically equivalent to the original object shape without excessive erosion by performing template matching on 21 newly designed templates. Furthermore, the corners and T-junctions are perfectly preserved in the obtained skeletons. It is helpful for many image analysis and recognition tasks since both corner and T-junction are important structural features for these tasks. Experimental results show that this algorithm is indeed in good performance.

1. Introduction

Since it is very difficult to derive the useful features or important control points from a thick stroke image, many image recognition and analysis processes skeletonize the processed images before their real recognition or analysis tasks are performed. The raster-to-vector task that is useful for the processing of an engineering drawing or a digital map is an example. A raster-to-vector process usually skeletonizes the image first, and then derives vector lines from the skeletons. Optical character recognition is another typical example. In order to derive the stroke features or the cross point features for recognition, an optical character recognition task also performs skeletonization on the recognized image first.

The skeletonization process is called thinning. There are many approaches to perform this process. Most of the studies perform thinning by removing the border points layer by layer from the thinning objects. Each pixel in the image is inspected and is removed if it is determined to be a border point. Algorithms in this approach can be categorized into sequential algorithms [1-3] and parallel algorithms [4-19].

A sequential thinning algorithm inspects pixels one by one in its defined order. Since whether a pixel is a border point or not is usually determined by its neighbors, the removal of a border point will affect the removal of its neighbors immediately. On the other hand, a parallel thinning algorithm processes all the pixels of the image simultaneously. When the thinning process is performed on a parallel machine, all the border points are removed at the same time. Since the processing time is dependent only on the thickness of the thinning object, the thinning task can be finished very quickly when a large image is been processed. During the thinning process, the removal of a pixel is decided only by its neighbors obtained in the previous iteration. Since a removed pixel informs nothing to its neighbors for their removal decision in the same iteration, it is possible that pixels are removed excessively. For example, pixels on a two-pixel width line will be removed thoroughly [18] since all the pixels are border points. Several methods were used to avoid this problem. Many researchers [4-11, 19] use multiple passes(or subcycles) in each thinning iteration. For example, the two-pass parallel thinning algorithm proposed by Zhang and Suen [5] removes all border points from the south-east side of the object shape in the first pass, saving the intermediate result; and removes in the second pass all border points from the north-west side of the object shape based on the intermediate result. Multi-pass methods indeed avoid excessive removal of border points, but they also slow down the processing speed. To overcome this problem. several one-pass parallel thinning algorithms [12-18] have been proposed.

Skeleton obtained by a thinning algorithm may be different from that obtained by another algorithm. There is no unique definition for the obtained skeleton which a thinning algorithm should produce. However, a good thinning algorithm should have the following characteristics in general.

1. It should preserve the connectivity of the object shape. Here the connectivity may be 4-neighbor connectivity or 8-neighbor connectivity. In this study, 8-neighbor connectivity is assumed.
2. It should not produce excessive erosion. Removing important points to eliminate meaningful strokes is called excessive erosion. This elimination will also eliminate the important information for further image analysis.
3. It should be noise-insensitive. Noise appearing on the boundary of an object shape should not intensively affect the thinning result.
4. It should produce a good representation of the original object shape. A skeleton must be topologically equivalent to the original object shape.

Criteria 1 through 3 are easy to observe, while criterion 4 is quite difficult to measure. As described in the criterion, the thinned skeleton must be topologically equivalent to the original object shape. In the other words, the thinning algorithm can neither lose any stroke nor create any extra stroke. In addition, the topological structures between strokes should be preserved in the thinned results. For example, the cross points, the fork points, and the corner points are important topological structural points. Since these points all describe the important topological relationships between two strokes, they are usually treated as important features in a recognition task. If the thinning process which acts as the preprocess of the recognition task loses these important structural features, the recognition task will usually fail

* This work was supported by National Science Council, Republic of China on the contract number NSC87-2213-E-128-001

consequently. Unfortunately, almost every existing thinning algorithm processes these features quite deficiently. More specifically, a fork point is usually thinned to be a 'Y' shaped skeleton, while a corner point is usually thinned to be a slant line. The important features are therefore lost in the thinned result. This is a momentous problem of a recognition task. In this paper, we propose a new one pass parallel thinning algorithm that will preserve the fork points as well as the corner points. The thinning results are described and discussed also.

The remainder of this paper is organized as follows. In Section 2, the proposed thinning algorithm is described in details. Section 3 presents some thinning results of the algorithm. And finally, Section 4 includes a conclusion.

2. Proposed Algorithm

Similar to most of the existed thinning algorithms, the proposed thinning algorithm removes the border points layer by layer. And, as a parallel one, the proposed algorithm removes all the border points at the same time if they are defined to be border points after the border points on the outer layer are removed. The process of deleting the border points in a snapshot is called a thinning iteration. During the thinning process is in progressing, an image plane, which we called the image plane, is used to represent the thinning states of the processed image, from the original input image to its skeletonized result. A pixel of the value 1 in the image plane represents that its positional corresponding pixel is black in the thinned image. While a pixel of the value 0 represents a white one. In addition to the image plane, the proposed algorithm needs another image plane, which we called the condensed image plane. In each iteration, the proposed algorithm derives the condensed image plane from the image plane. Template matching is then performed both on the image plane and on the condensed image plane to select the border points. After all the border points are simultaneously removed from the image plane, the next iteration proceeds consequently. Since a border point is determined by template matching, a set of templates is designed for the image plane, and another set of templates is designed for the condensed image plane.

1.1 Condensed Image Plane

The size of a condensed image plane is the same as that of the image plane from which it is derived. Suppose the image plane is in the size of MxN, i.e., M pixels height and N pixels width. We can label a pixel in the plane by P_{ij}, where $1 <= i <= M$ and $1 <= j <= N$. Because the condensed image plane has the same size as the image plane, a pixel in it can also be labeled in the same manner. Let's label a pixel in the condensed image plane by D_{ij}, where $1 <= i <= M$ and $1 <= j <= N$. And we can say that pixel D_{ij} in the condensed image plane is the positional corresponding pixel of pixel P_{ij} in the image plane.

As described previously, the condensed image plane is derived from the image plane in each thinning iteration. Each pixel in the condensed image plane must be reset in each iteration. More specifically, a pixel D_{ij} in the condensed image plane is defined as

$$D_{ij} = \begin{cases} 1 & \text{if } P_{kl} = 1, \forall\, i-1 \le k \le i+1 \text{ and } j-1 \le l \le j+1, \\ & \quad \text{where } 2 \le i \le M-1 \text{ and } 2 \le j \le N-1; \\ 0 & \text{otherwise.} \end{cases}$$

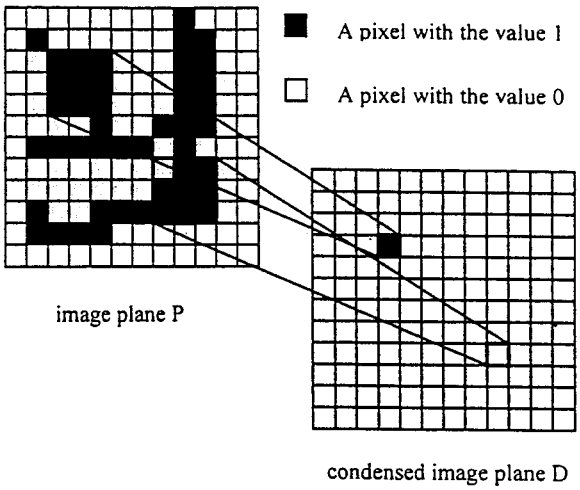


Fig. 1. Condensed image plane and its corresponding image plane in the size of 12x12. Pixel D_{44} is of the value 1 since P_{44} and its eight neighbors are all of the value 1. Another pixel D_{99} is of the value 0 since P_{88} is of the value 0.

From the definition, D_{ij} is defined to be 1 if and only if P_{ij} and the eight neighbors of P_{ij} are all of the value 1. Fig. 1 shows the condition to derive a D_{ij} of the value 1. The marginal pixels of the condensed image plane, i.e., the pixels D_{ij}, where $i = 1$ or $i = M$ or $j = 1$ or $j = N$, are always of the value 0, since there is not enough neighbors of P_{ij} to support D_{ij} to be the value 1.

1.2 Thinning Templates

To determine whether a pixel is a border point or not, the proposed algorithm performs template matching both on the image plane and on the condensed image plane. Thus, thinning templates are designed both for the image plane and for the condensed image plane. In the image plane, the pixel that is focused for the determination of removal is called the tested pixel. For each black pixel (a pixel of the value 1) P_{ij} in the image plane, template matching is performed on the neighborhood of P_{ij} in the image plane and on the neighborhood of D_{ij} in the condensed image plane. As shown in Fig. 2, eighteen templates(template (a) through template (r)) are designed for the image plane, and three templates(template (o') through (q') are designed for the condensed image plane. Each little square in the figure represents a placeholder with which a pixel should be matched. Symbol 'c' denotes the placeholder corresponding to the tested pixel. Symbol '1' and '0' denote the placeholder that should be matched with a pixel of the value 1 and 0 correspondingly. And, an empty square represents a don't care pixel.

1.3 Border points

In each thinning iteration, border point removals is determined by template matching. To describe the template matching process, we define the neighborhood of a tested pixel P_{ij} to be the set of the pixels around it in the image plane. Since the neighborhood should be matched with the templates shown in Fig. 2, it consists of the twelve neighbors shaped in diamond. In the condensed image plane, we also define the neighborhood of the

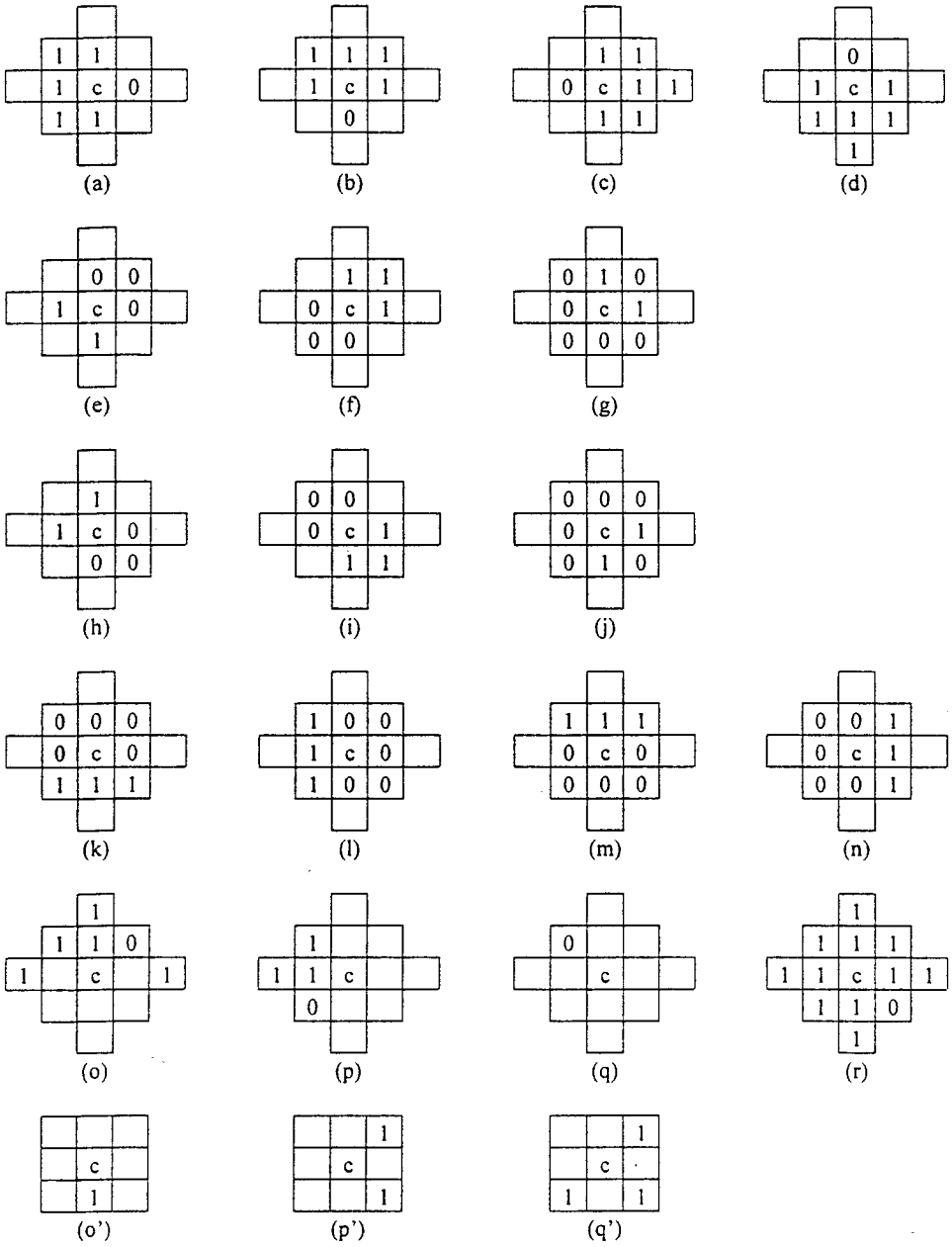


Fig. 2. Thinning templates. Templates (a)-(r) are templates for the image plane, and templates (o')-(q') are the associated templates of templates (o)-(q) for the condensed image plane.

pixel D_{ij}. This neighborhood is defined to be the set of the eight pixels adjacent to the pixel D_{ij} since the templates designed for the condensed image plane is shaped in 3x3 square. For each black tested pixel P_{ij}, if its neighborhood is matched with any of the template (a) through (n) or template (r), then it is determined as a border point by the proposed algorithm. Unlike this kind of simple matching, templates (o) through (q) and templates (o') through (q') are matched in a shortly different way. Template (o) and template (o') form a pair and should be matched simultaneously with the neighborhood of pixel P_{ij} and the neighborhood of pixel D_{ij} correspondingly. When both of these two matches are successful, pixel P_{ij} is determined to be a border pixel. Otherwise, it is not. The fact is also true for the pair of template (p) and template (p') as well as the pair of template (q) and template (q').

1.4 The proposed thinning algorithm

For convenience, let's denote the image plane at the k-th iteration as P^k and the condensed image plane at the k-th iteration as D^k. The proposed algorithm is described precisely as follows.

Input. A binary image.
Output. The image of the thinning result.
Step 1. Initialize the image plane P^0 according to the input image.

set $k := 0$.

Step 2. Derive the condensed image plane D^k from the image plane P^k. D^k_{ij} is set to 1 if and only if P^k_{ij} and the eight neighbors of P^k_{ij} are all black pixels.

Step 3. $k := k + 1$; flag := FALSE.

Step 4. Check each pixel P^k_{ij} of P^k. If its value is 1 and either one of the following two conditions is true,

Condition 1: its neighborhood is matched with any of the templates (a) through (n) or template (r); or

Condition 2: its neighborhood and the neighborhood of D^k_{ij} are matched with any of the template pairs (o-o') through (q-q'),

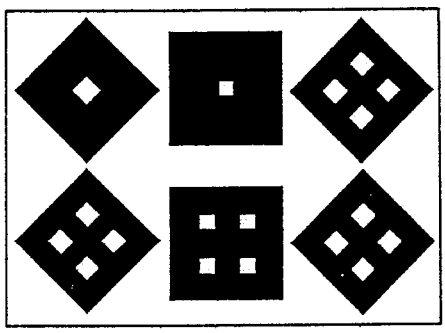
then change its value to 0 and set flag := TRUE.

Step 5. If flag = FALSE, which means the image has been thinned, then go to Step 6. Otherwise, go to Step 2 to perform the next iteration.

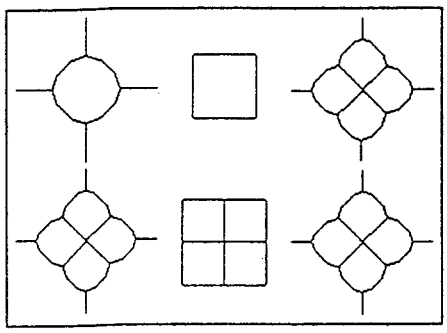
Step 6. Output the thinned image according to P^k.

3. Experimental Results

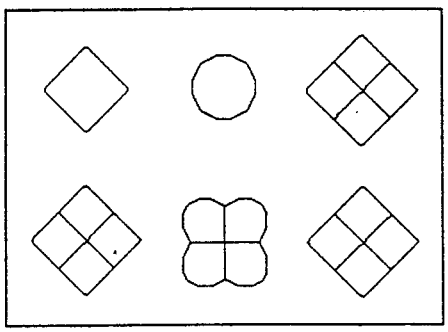
Fig. 3. shows the thinning results of some drawings shaped like the Chinese character '□' and '田'. In this figure, the thinning results of four parallel thinning algorithms are shown for comparison. These thinning algorithms are Datta's algorithm [19], Wu's algorithm [18], Chin's algorithm [15], and the proposed algorithm. Experiments discussed below are also performed by these four algorithms. In fact, unlike the other three one-pass parallel thinning algorithms, Datta's algorithm is a multi-pass thinning algorithm. Due to its good thinning result, we include it here for discussion. Fig. 3(b) through Fig. 3(e) show the results derived by Datta's algorithm, Wu's algorithm, Chin's algorithm, and the proposed algorithm, correspondingly. Observing the



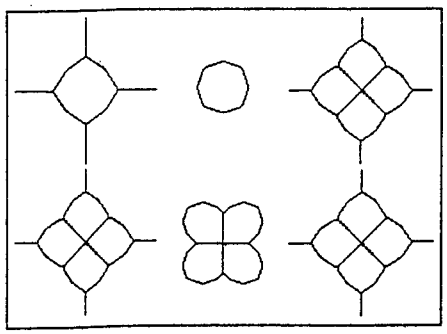
(a) the original image



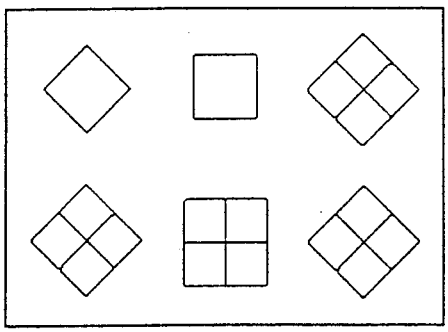
(b) Datta



(c) Wu



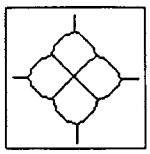
(d) Chin



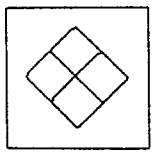
(e) proposed algorithm

Fig. 3. An image and its thinning results. (a) is the original image. (b) through (e) are the results thinned by Datta's algorithm, Wu's algorithm, Chin's algorithm, and the proposed algorithm, correspondingly.

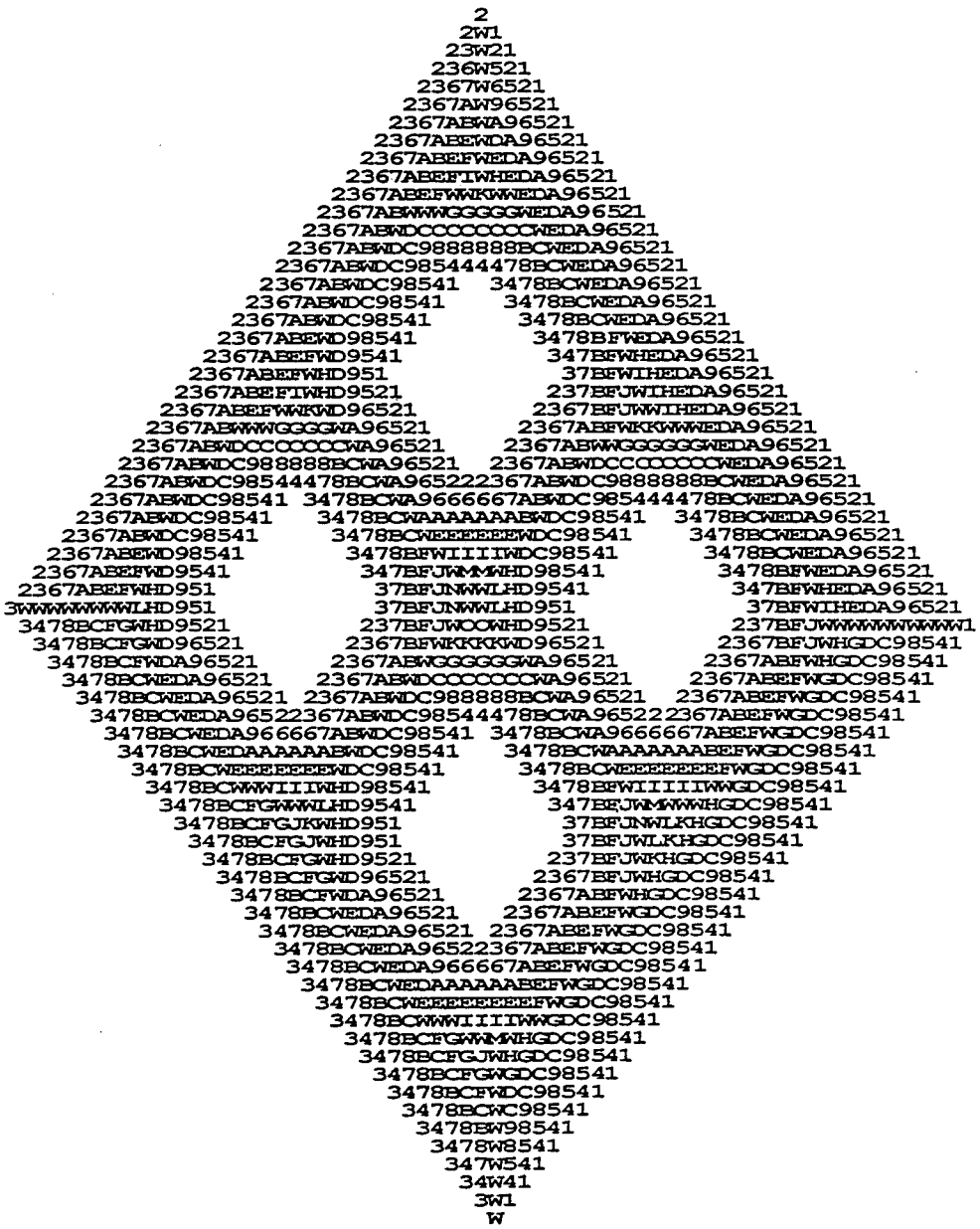
(a) the original image



(b) Datta

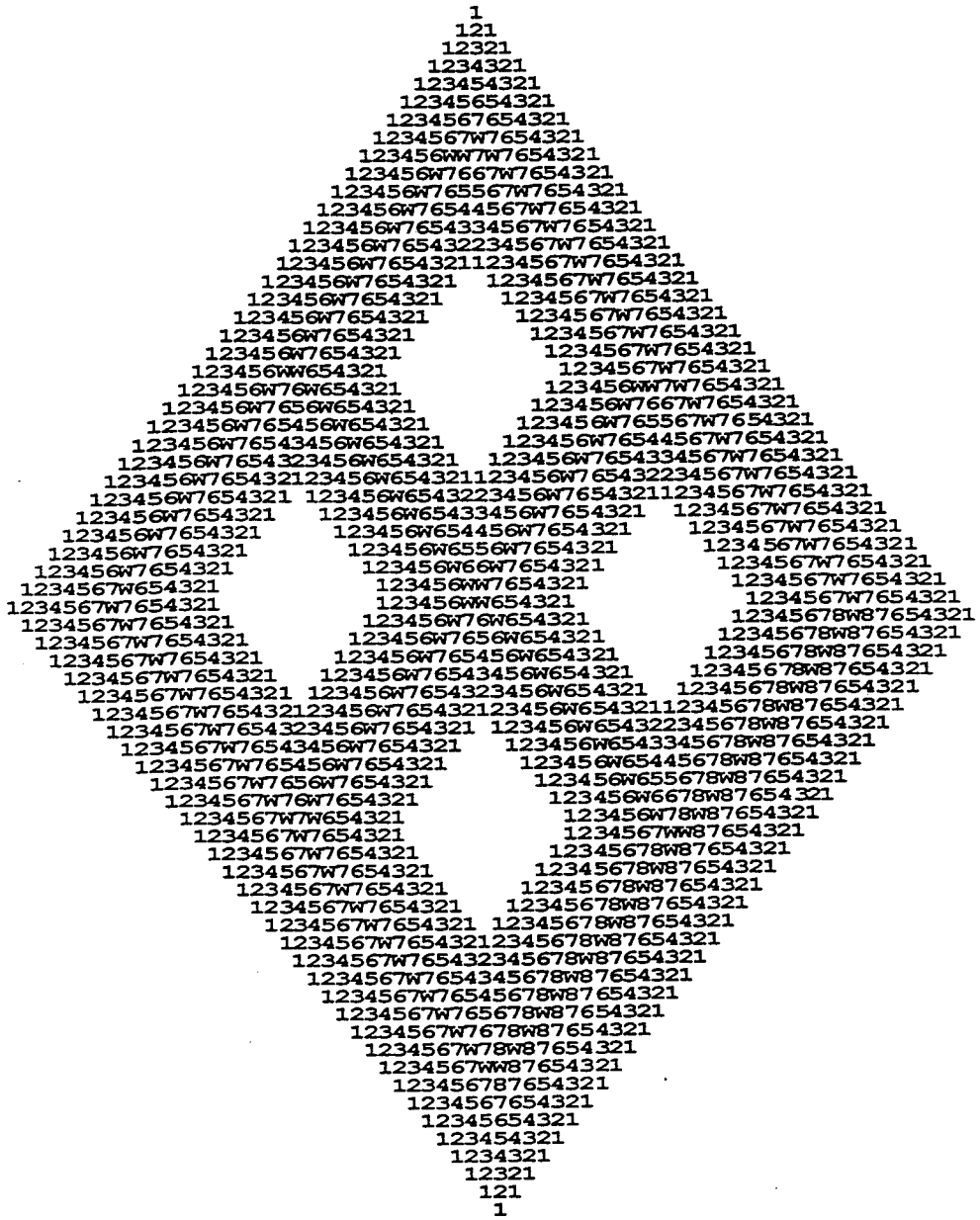


(c) proposed algorithm



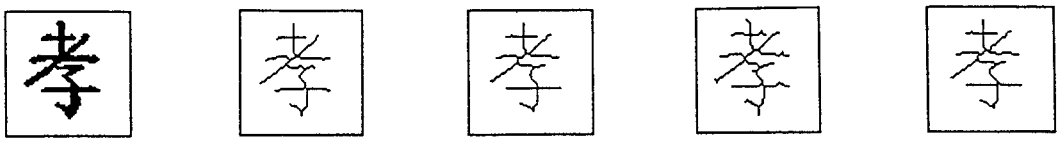
(d) procedure by Datta's algorithm

Fig. 4. Thinning results and procedures. (a) is the original image; (b) and (c) are the thinning results of Datta's algorithm and the proposed algorithm, correspondingly; (d) and (e) are their corresponding thinning procedures.



(e) procedure by the proposed algorithm

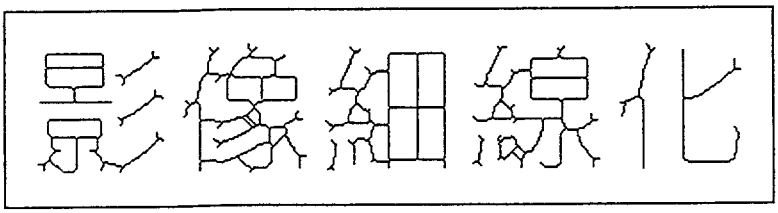
Fig. 4. Thinning results and procedures. (a) is the original image; (b) and (c) are the thinning results of Datta's algorithm and the proposed algorithm, correspondingly; (d) and (e) are their corresponding thinning procedures. (Continued)



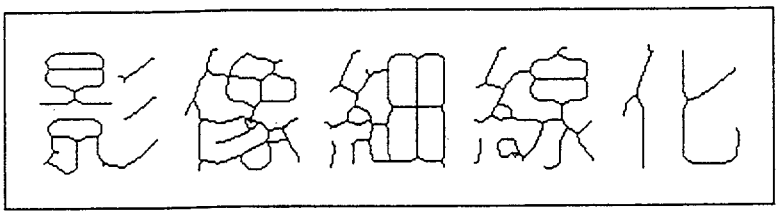
(a) (b) (c) (d) (e)

Fig. 5. Thinning results of a handprinted Chinese character "孝". (a) is the original image; (b) through (e) are the results thinned by Datta's algorithm, Wu's algorithm, Chin's algorithm, and the proposed algorithm, correspondingly.

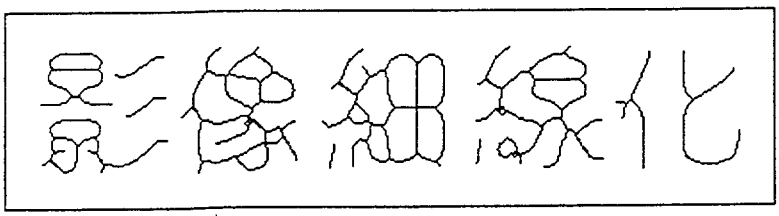
(a) the original image



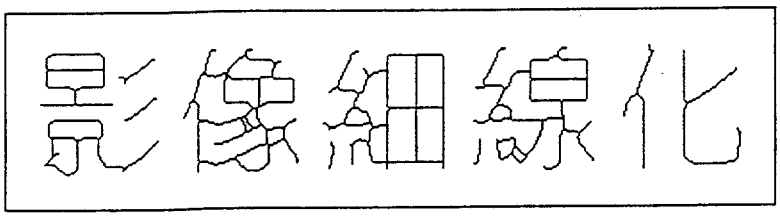
(b) the result of Datta's algorithm



(c) the result of Wu's algorithm



(d) the result of Chin's algorithm



(e) the result of the proposed algorithm

Fig. 6. Thinning results of printed Chinese characters "影像細線化". (a) is the original image; (b) through (e) are the results thinned by Datta's algorithm, Wu's algorithm, Chin's algorithm, and the proposed algorithm, correspondingly.

results, we can find Datta's algorithm is good for right corners and right T-junctions, but it is bad for slant corners and slant T-junctions. In contrast, Wu's algorithm is good for slant features, but bad for right features. Chin's algorithm performs worst in this experiment. On the other hand, the proposed algorithm performs well at all the corners and T-junctions.

Fig. 4 shows the results of another experiment. Not only the thinning results, the thinning procedures are also shown. In order to observe the procedural difference of the removals of border points between a one-pass parallel thinning algorithm and a multi-pass one, the figure shows the thinning procedure of the proposed algorithm and that of Datta's algorithm in Fig. 4(e) and Fig. 4(d), correspondingly. In the figure, symbol 'W' denotes the skeleton points. The other alphanumerical characters denote the removed points. Symbol '1' denotes points removed in the first pass, and symbols '2' through '9' denote points removed in its corresponding pass. Alphabetical characters denotes points removed in passes later than the 9-th pass in accordance with their alphabetical order. As shown in Fig. 4(d), the largest symbol is 'O', which represents points removed in the 24-th pass. That is, Datta's algorithm performs this thinning task with 24 passes, while the proposed algorithm performs the same task only with 8 passes. Since the processing time of a parallel thinning algorithm implemented in a parallel logic circuit is dependent on the number of passes it performs, the proposed algorithm is quite faster than Datta's algorithm.

Fig. 5 and Fig. 6 show the thinning results of a handprinted Chinese character '孝' and a printed Chinese clause "影像細線化", respectively. Experiments are also performed by the four algorithms that are tested in the first experiment, and the results are shown in the same order as that in Fig. 3. As shown in the figures, the proposed algorithm derives good results in both of these two experiments. More specifically, it is noise insensitive and without excessive erosion. The derived skeletons are also topologically equivalent to the original object shapes. Especially, at the corners and T-junctions, the thinning results of the proposed algorithm are better than that of the other parallel algorithms.

4. Conclusions

As discussed in the previous section, we have proposed an efficient corner preserved one-pass parallel thinning algorithm. Experimental results have shown that this algorithm can derive good thinning results. Especially, for the preservation of corners and T-junctions, the proposed algorithm performs much better than the other parallel thinning algorithms. As a one-pass parallel thinning algorithm, its processing speed is surely faster than a multi-pass one. An experiment has also shown this truth. Further more, in spite of a one-pass algorithm, its derived skeletons are better than that of a multi-pass one. It is believed that it is helpful for an image analysis task to preprocess its target image by the proposed algorithm before the real analysis task is performed.

References

[1] C. J. Hilditch, Linear Skeletons from Square Cupboards, in B. Meltzer and D. Michie (Eds.), *Machine Intelligence IV*, American Elsevier, New York, pp. 403-420, 1969.

[2] E. R. Davies, and A. P. N. Plummer, Thinning Algorithms: A Critique and A New Methodology, *Pattern Recognition*, Vol. 14, No. 1-6, pp. 53-56, 1981.

[3] N. J. Naccache, and R. Shinghal, An Investigation into The Skeletonization Approach of Hilditch. *Pattern Recognition*, Vol. 17, No. 3, pp. 279-284, 1984.

[4] R. Stefanelli and A. Rosenfeld, Some Parallel Thinning Algorithms for Digital Pictures, *J. ACM*, Vol. 18, No. 2, pp. 255-264, 1971.

[5] T. Y. Zhang and C. Y. Suen, A Fast Parallel Algorithm for Thinning Digital Patterns, *Commun. ACM*, Vol. 27, No. 3, pp. 236-239, 1984.

[6] H. E. Lu, and P. S. P. Wang, A Comment on "A Fast Parallel Algorithm for Thinning Digital Patterns", *Commun. ACM*, Vol. 29, No. 3, pp. 239-242, 1986.

[7] Y. S. Chen, and W. H. Hsu, A Modified Fast Parallel Algorithm for Thinning Digital Patterns, *Pattern Recognition Letters* 7, pp. 99-106, 1988.

[8] J. H. Sossa, An Improved Parallel Algorithm for Thinning Digital Patterns, *Pattern Recognition Letters* 10, pp. 77-80, 1989.

[9] R. W. Hall, Fast Parallel Thinnings Parallel Speed and Connectivity Preservation, *Commun. ACM*, Vol. 32, No. 1, pp. 124-131, 1989.

[10] Z. Guo and R. W. Hall, Parallel Thinning with Two-Subiteration Algorithms, *Commun. ACM*, Vol. 32, No. 3, pp. 359-373, 1989.

[11] L. Hayat, A. Naqvi, and M. B. Sandler, Comparative Evaluation of Fast Thinning Algorithms on A Multiprocessor Architecture, *Image and Vision Computing*, Vol. 10, No. 4, pp. 210-218, 1992.

[12] Y. S. Chen, and W. H. Hsu, A Systematic Approach for Designing 2-Subcycle and Pseudo 1-Subcycle Parallel Thinning Algorithms, *Pattern Recognition*, Vol. 22, No. 3, pp. 267-282, 1989.

[13] P. Kumar, D. Bhatnagar, and P. S. U. Rao, Pseudo One Pass Thinning Algorithm, *Pattern Recognition Letters* 12, pp. 543-555, 1991.

[14] C. M. Holt, A. Stewart, M. Clint, and R. H. Perrott, An Improved Parallel Thinning Algorithm, *Commun. ACM*, Vol. 30, No. 2, pp. 156-160, 1987.

[15] R. T. Chin, H. K. Wan, D. L. Stover, and R. D. Iverson, A One-Pass Thinning Algorithm and Its Parallel Implementation, *Computer Vision, Graphics, and Image Processing* 40, pp. 30-40, 1987.

[16] C. S. Chen and W. H. Tsai, A New Fast One-Pass Thinning Algorithm and Its Parallel Hardware Implementation, *Pattern Recognition Letters* 11, pp. 471-477, 1990.

[17] Y. S. Chen, and W. H. Hsu, A Comparison of Some One-pass Parallel Thinnings, Pattern Recognition Letters 11, pp. 35-41, 1990.

[18] R. Y. Wu, and W. H. Tsai, A New One-Pass Parallel Thinning Algorithm for Binary Images, Pattern Recognition Letters 13, pp. 715-723, 1992.

[19] A. Datta, and S. K.Parui, A Robust Parallel Thinning Algorithm for Binary Images, *Pattern Recognition*, Vol. 27, No. 9, pp.1181-1192, 1994.