A new one-pass parallel thinning algorithm for binary images

Rei-Yao Wu*

Institute of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan 30050, ROC

Wen-Hsiang Tsai

Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, ROC

Abstract

Wu, R.-Y. and W.-H. Tsai, A new one-pass parallel thinning algorithm for binary images, Pattern Recognition Letters 13 (1992) 715–723.

A new parallel thinning algorithm for binary images is proposed. This algorithm employs template matching to remove the edge points of an object shape in a binary image iteratively. Difficulty of terminating the thinning process without excessive erosion is solved without dividing an iteration into several passes. With a newly designed set of matching templates, 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.

1. Introduction

Thinning is a process to remove the outer points of an object shape in an image. The remaining points form the skeleton of the object and so the process is also called skeletonization. The thinning process occasionally removes a large volume of

Correspondence to: W.-H. Tsai, Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, ROC.

* Also with Institute for Information Industry, Taipei, Taiwan 10636, ROC.

data, and so improves the efficiency of many subsequent image analysis tasks.

Many thinning algorithms have been proposed up to now. Most of them remove the boundary points of an object shape layer by layer to obtain the skeleton of the object. These algorithms can be divided into two categories, namely, sequential and parallel algorithms. A sequential thinning algorithm performs the pixel-removal work pixel by pixel in the image plane, and the removal decision depends on the result obtained so far in the current iteration as well as those of the previous iterations. On the other hand, a parallel thinning algorithm processes all the pixels of the image simultaneously, and the removal decision depends only on the result of the previous iterations.

The skeleton of an object shape obtained from the thinning process does not have a unique definition. 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 points in important positions such as line segment ends will eliminate relevant 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 shape. A skeleton must be topologically equivalent to the original object shape (Chin et al. (1987)). A good result is the medial axis of the object shape.

Many of the existing parallel thinning algorithms (Stefanelli and Rosenfeld (1971), Zhang and Suen (1984)) use several passes (or subcycles) in each thinning iteration. For example, the two-pass parallel thinning algorithm proposed by Zhang and Suen (1984) removes all boundary pixels from the south-east side of an object shape in the first pass, saving the intermediate result; and removes in the second pass all boundary pixels from the north-west side of the object shape based on the intermediate result. Some problems, for example, removal of all lines with two-pixel width, that might occur in a parallel thinning algorithm, can be avoided with this kind of multi-pass method. However, thinning speed will be reduced by such sequential processing passes.

To overcome the time consuming problem of the multi-pass methods, some one-pass parallel thinning algorithms (Holt et al. (1987), Chin et al. (1987), Chen and Tsai (1990)) have been proposed. Holt et al. (1987) expanded the window for each pixel to include the edge information about its neighbors, and used a Boolean expression, which indicates whether a tested pixel should be removed, to construct a one-pass parallel thinning algorithm. Unfortunately, imperfectly 8-connected thinning results will be obtained after edge removal. An

extra clean-up phase was used further to eliminate undesirable staircase shapes in the final result.

Chin et al. (1987) used a set of 3×3 thinning templates as well as two restoring templates (one 1×4 and the other 4×1) to deal with the breakage and disappearance of horizontal and vertical limbs with two-pixel width. To avoid producing noisy skeletons, they used eight trimming templates to clear noise in each iteration. But the use of the trimming templates causes another problem, as indicated by the authors, that the end points of object shapes are no longer preserved. In the extreme case, if a limb with one-pixel width is thinned, the object shape will degenerate to a single pixel.

In addition to the use of the image plane, Chen and Tsai (1990) used a so-called dense image plane to get more edge information. Template matching is applied both to the image plane and to the dense image plane to decide the survival of each pixel. Like Chin's algorithm, a set of templates were used to clear noise in each iteration. Again, this noise cleaning process causes excessive erosion. Using the dense image plane to get more information seems a good idea to preserve the object shape, but the obtained result is not perfectly 8-connected.

Unlike the one-pass thinning algorithms mentioned above, a new one-pass parallel thinning algorithm is proposed in this paper which uses only a set of thinning templates for template matching. In addition to its simplicity, the algorithm produces perfectly 8-connected and noise-insensitive results without excessive erosion. The experimental result shows that this algorithm possesses all the characteristics of a good parallel thinning algorithm described previously.

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

2. Proposed thinning algorithm

Figure 1 shows the templates for use in the proposed thinning algorithm. There are twelve 3×3 templates (templates (a), (b), and (e) through (n)), one 3×4 template (template (c)), and one 4×3

```
1 1 y        1 1 1        y 1 1 x      y 0 y
1 c 0        1 c 1        0 c 1 1      1 c 1
1 1 y        y 0 y        y 1 1 x      1 1 1
                                       x 1 x

 (a)          (b)          (c)          (d)
```

```
x 0 0        x 1 1        0 1 0        x 1 x        0 0 x        0 0 0
1 c 0        0 c 1        0 c 1        1 c 0        0 c 1        0 c 1
x 1 x        0 0 x        0 0 0        x 0 0        x 1 1        0 1 0

 (e)          (f)          (g)          (h)          (i)          (j)
```

```
0 0 0        1 0 0        1 1 1        0 0 1
0 c 0        1 c 0        0 c 0        0 c 1
1 1 1        1 0 0        0 0 0        0 0 1

 (k)          (l)          (m)          (n)
```
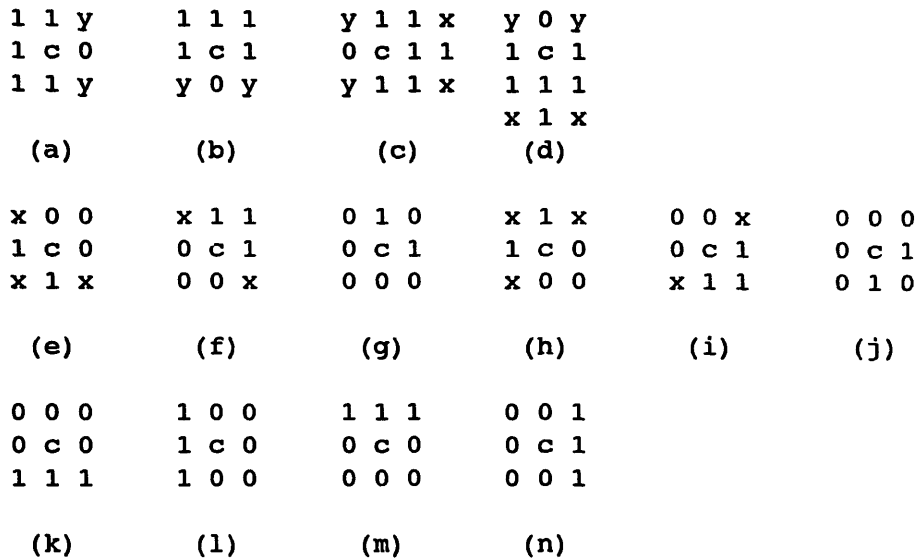
Figure 1. Proposed thinning templates.

template (template (d)) in this template set. The symbols 'c', '0', '1', and 'x' used in these templates denote the currently tested pixel, a white pixel, a black pixel, and a don't-care condition, respectively. These symbols follow the conventional notations, while the symbol 'y' also appearing in the templates is a special one. It does not appear singly in a thinning template, and at least one of the pixels represented by the set of symbols 'y' should be a white pixel. The proposed thinning algorithm uses these templates to eliminate the edge points of a thick object shape layer by layer in parallel. More specifically, the proposed algorithm iteratively applies a matching test to each pixel of an image in parallel to determine whether the tested pixel should be removed (i.e., to decide whether it should be changed to a white pixel if it is a black pixel). A pixel is removed if its neighboring pixels match any of the above templates (a)-(n). In each iteration of this process, the information about each tested pixel and its neighboring pixels are obtained from the result of the previous iteration.
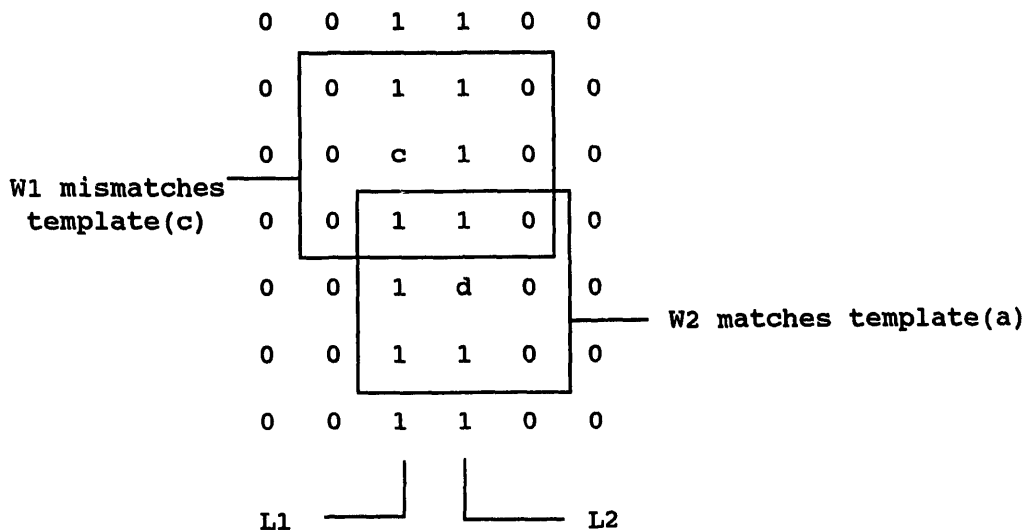

Figure 2. Vertical line with two-pixel width can be thinned by template pair (a) and (c). Symbols 'c' and 'd' denote the tested black pixels in the image. Pixel 'c' will be left since the pattern of its neighbors does not match template (c), while pixel 'd' will be removed since the pattern of its neighbors matches template (a).
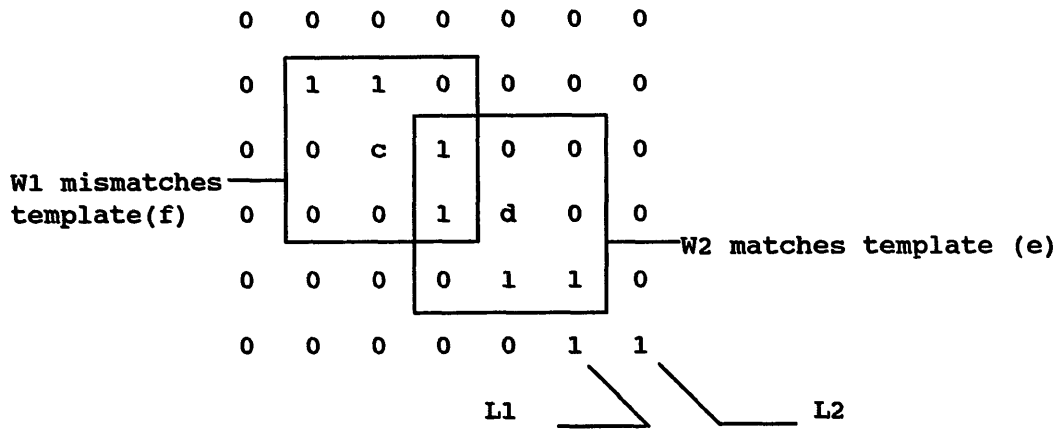
```
        O    O    O    O    O    O    O

        O    1    1    0    0    0    0

        O    0    c    1    0    0    0
W1 mismatches
template(f)  O    0    0    1    d    0    0

        O    0    0    0    1    1    0

        O    0    0    0    0    1    1
```

W2 matches template (e)

L1   L2

Figure 3. A diagonal line with two-pixel width can be thinned by template pair (e) and (i). Symbols 'c' and 'd' denote the tested black pixels in the image. Pixel 'c' will be left since the pattern of its neighbors does not match template (f), while pixel 'd' will be removed since the pattern of its neighbors matches template (e).

Accordingly, the proposed algorithm is a one-pass parallel thinning algorithm. For convenience, the image to be processed at the i-th iteration is denoted as f^i. The proposed algorithm is described precisely as follows.

Algorithm (proposed one-pass parallel thinning)
Input. A binary image f^0.
Output. The image of the thinning result.
Step 1. $i := 0$.
Step 2. $i := i + 1$; flag := false.
Step 3. Check each pixel of f^i. If it is a black pixel and its neighbors match any of the templates (a)–(n), then change it to a white pixel and set flag := true.
Step 4. If flag = false, which means the image is thinned, then go to Step 5 with f^i as the thinning result. Otherwise, go to Step 2 to perform the next iteration.
Step 5. Output the thinning result.

The proposed algorithm is basically a template matching algorithm. It can be seen that templates (a)–(d) shown in Figure 1 are used to remove the edge points of horizontal and vertical lines. Templates (e)–(j) are used to remove the edge points of diagonal lines in the directions of north-west to south-east as well as north-east to south-west. Finally, templates (k)–(n) are used to eliminate noise points appearing beside the edge of a straight line. The idea of employing these templates is described in the following.

Template (a) is used for the removal of the points on the east edge. On the other hand, it seems that we can simply use the template

$$
\begin{array}{ccc}
y & 1 & 1 \\
0 & c & 1 \\
y & 1 & 1
\end{array}
\tag{A}
$$

to deal with the removal of west edge points. But,

```
                        *A              *B
-----------------*-------------*------------------
-------------**-**--------**-**---------------
--****-****-----******-----*****-*****--
------*------------------------------*--------
------*-----------------------------*--------
------*----------------------------*--------
      *C                            *D
```
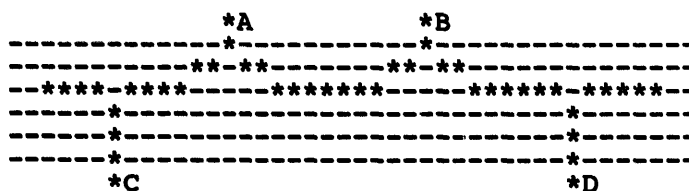
Figure 4. Noise points A, B, C, and D cause heavily shape variations. Symbol '*' denotes the obtained skeleton, while symbol '-' denotes the removed points.

for a parallel thinning algorithm, this is not true.

In the case of using a one-pass parallel thinning algorithm, the use of the pair of templates (a) in Figure 1 and (A) above will clear a vertical line with two-pixel width (or with width of an even number of pixels) thoroughly since each pair of an east edge point and a west edge point, which are adjacent to each other, of a vertical line with two-pixel width will be tested and removed simultaneously. Two methods can be used to resolve this problem in a parallel thinning algorithm. One is to divide each thinning iteration into two passes, pass one for east edge point removal and pass two for west edge point removal, or the reverse. The other

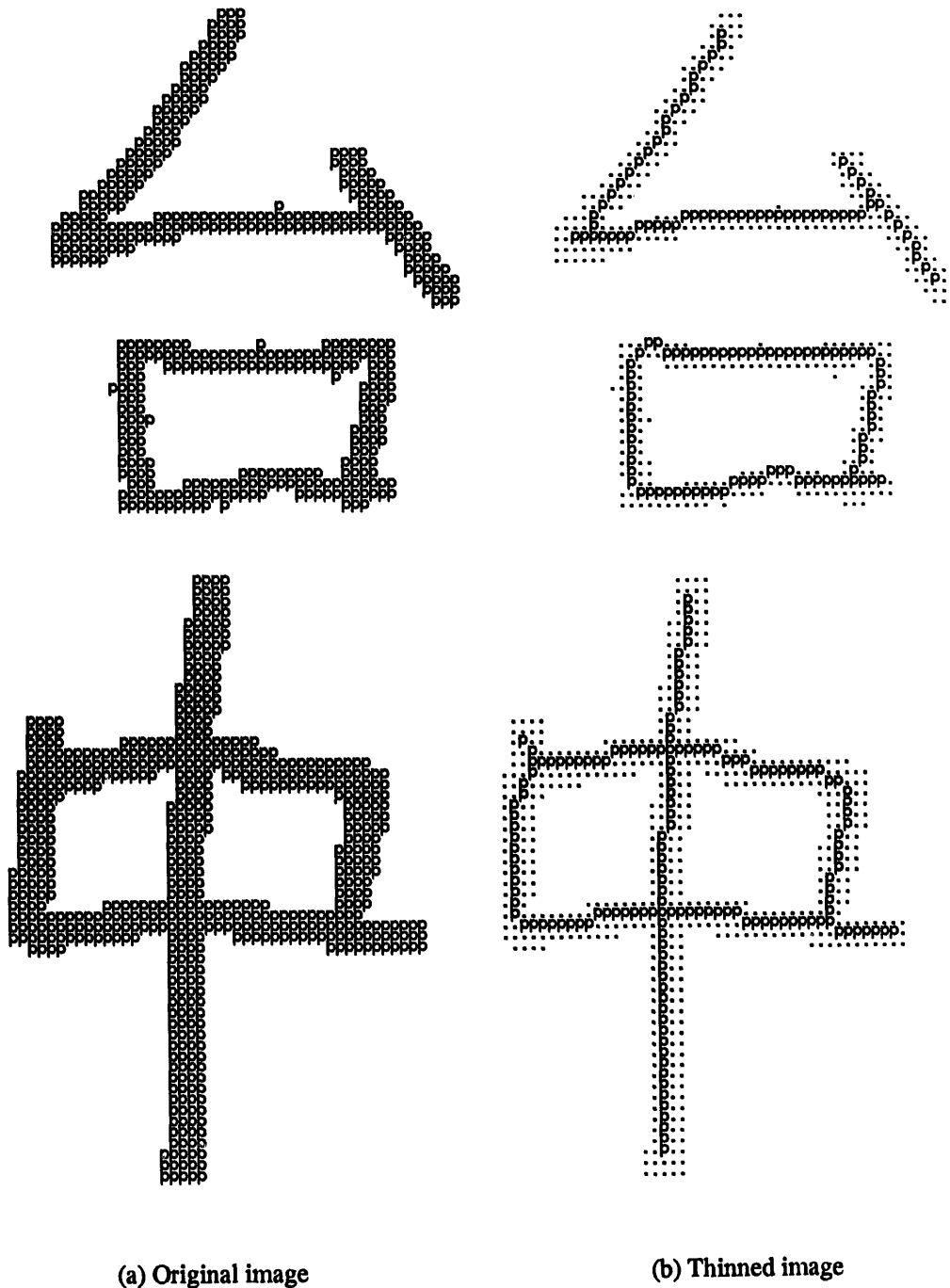(a) Original image   (b) Thinned image

Figure 5. Two thinning results of the proposed algorithm.

method is to set different priorities to the removal of the edge points of the east side and to that of the west side, such that when the edge points in each side of the vertical line with two-pixel width are tested for removal, only the edge points of one side will be removed, and the other side will be left as

the skeleton. Chin et al. (1987) used thinning templates as well as restoring templates to perform this directional bias. Their algorithm so needs two types of templates which have lower uniformity. The implementation also becomes more complicated. On the other hand, the algorithm proposed
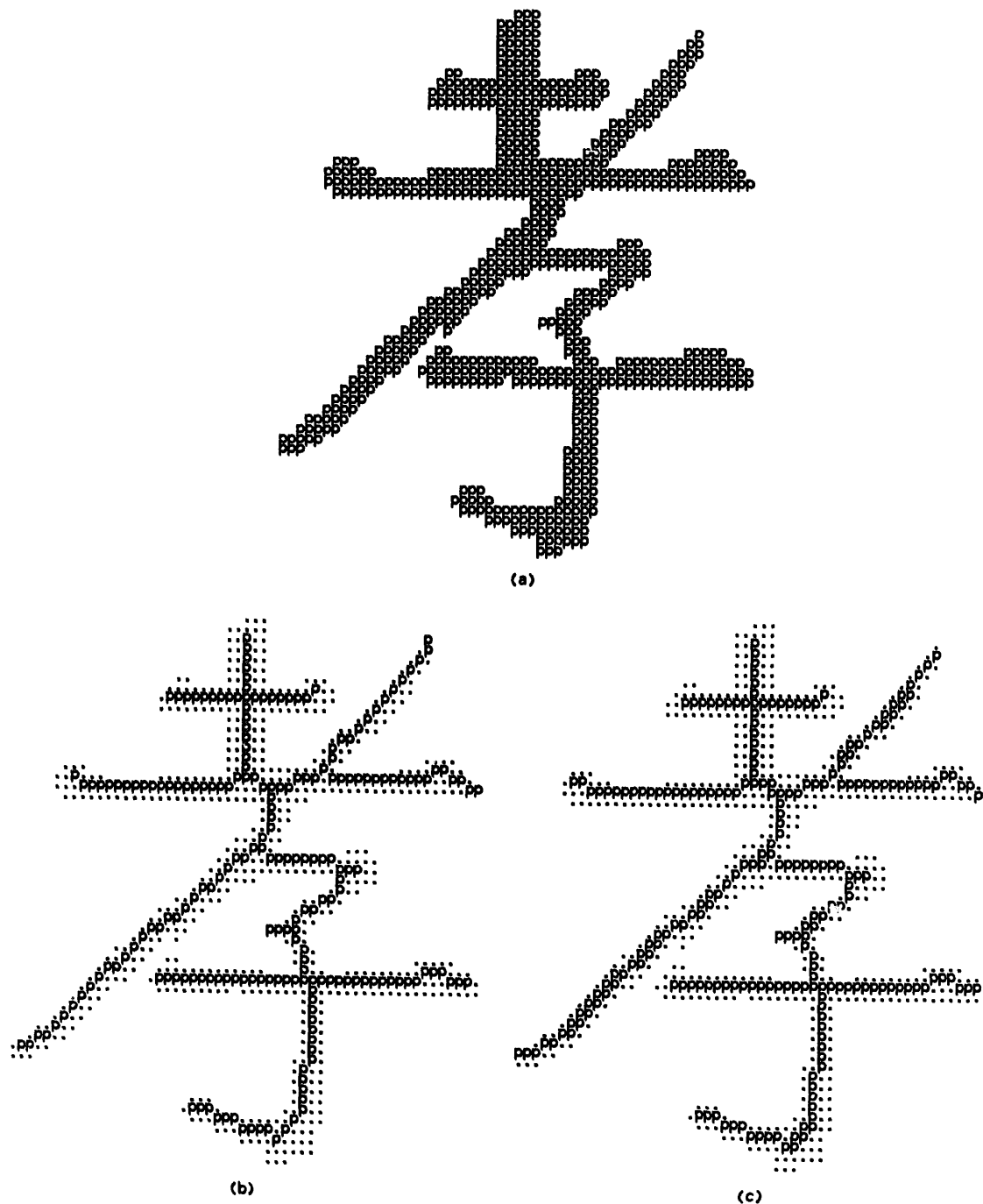

(a)

(b)

(c)

Figure 6. Comparison of one-pass parallel thinning algorithms. (a) The original image. (b) The result of the proposed algorithm. (c) The result of Holt's algorithm.
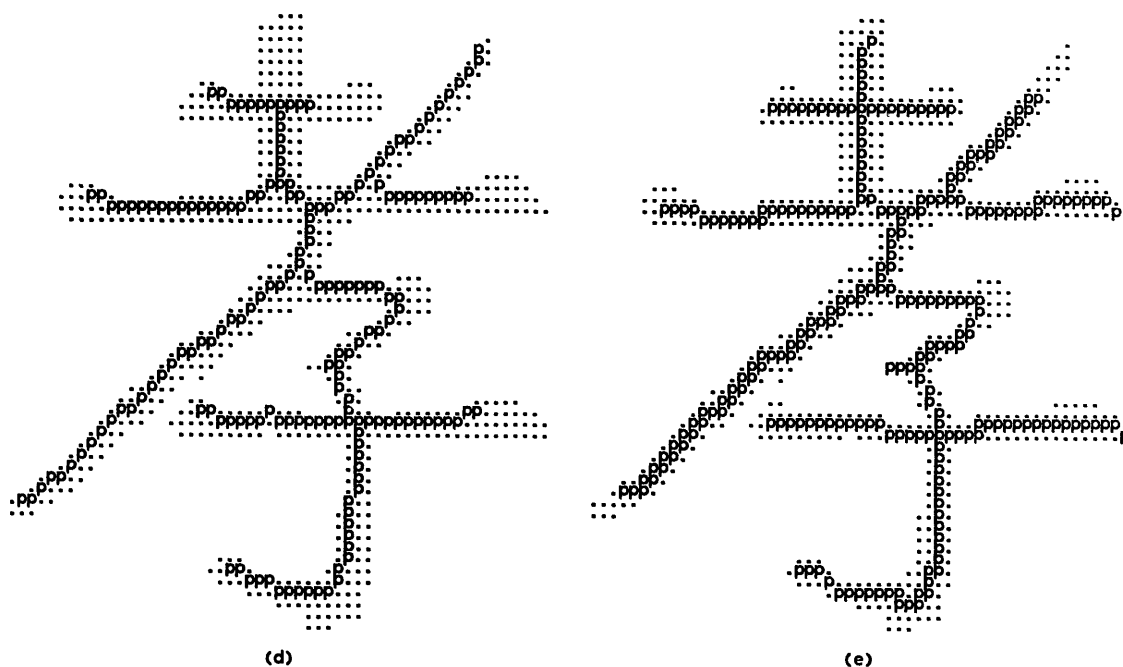
Figure 6 (contd.). (d) The result of Chin's algorithm. (e) The result of Chen and Tsai's algorithm.

here uses only a single type of templates to perform the required bias. Only a 3×4 template (template (c)) is included to detect whether the west edge point appears beside a vertical line wider than two pixels. That is, the pixels on the west edge will be removed only when the width of the vertical line is greater than two. In the case of treating a vertical line with two-pixels width, the edge points in the east side will be removed since the patterns of their neighbors match template (a), while the points in the west side will be left since the patterns of their neighbors mismatch template (c). As shown in Figure 2, line L2 will be removed, while line L1 will be retained as part of the skeleton. From similar discussions for horizontal line processing, templates (b) and (d) can be derived.

Besides vertical lines and horizontal lines, diagonal lines also need be considered. Templates (e)–(j) are designed for the removal of diagonal edge points. A perfectly thinned 8-connected diagonal line will be derived from applying these patterns in the matching test. Templates (e) and (f) are the pair for the removal of the edge points of the north-west to south-east diagonal line. Here, template (e) is responsible for the removal of the north-east-side edge points and template (e) is for

that of the opposite side. Similar to the design idea of the pair (a) and (c) for vertical lines, this pair (e) and (f) are not exactly co-reflective. The south-west point in template (e) is represented by symbol 'x', while the north-east point in template (f) by '1'. This design achieves the bias requirement for parallelly processing diagonal lines with two-pixel width; a diagonal line with two-pixel width will not be completely eliminated. As shown in Figure 3, line L2 will be removed, while line L1 will be left as part of the skeleton. Unfortunately, setting the north-east point of template (f) to '1' will miss the removal of a point with its neighbor pattern which matches template (g). Template (g) is therefore added to perform this removal. In the same manner, templates (h)–(j) are derived for processing diagonal lines in the other direction.

The last subset of the proposed templates is designed for noise removal. Noise appearing beside an object shape occasionally causes unexpected forks in many thinning algorithms. An example is shown in Figure 4, in which the noise points A, B, C, and D appearing beside the boundary of a horizontal line will cause unexpected results. This problem can be avoided by applying templates (k)–(n) to remove the noise points.

3. Experimental results

In this section, some experimental results are given to illustrate the effects of the proposed algorithm. The thinning results of some other one-pass parallel thinning algorithms are also given for comparison. Figures 5–7 show the results in which skeleton points are marked by 'p', and all those points that have been removed are marked by ' · '.

Figure 5 shows two thinning results of the proposed algorithm. The input images used here are two Chinese characters in dimension of 64×64. The results show that the proposed algorithm obtains perfectly 8-connected and noise-insensitive skeletons without excessive erosion.

Figure 6 shows another experimental result. In this experiment a third Chinese character also in dimension of 64 × 64 was thinned using the proposed algorithm and several other one-pass parallel thinning algorithms. And the results were compared. In Figure 6, (a) is the original shape, (b) is

the thinning result of the proposed algorithm, (c) is the result of Holt's algorithm (1987), (d) is the result of Chin's algorithm (1987), and (e) is the result of Chen and Tsai's algorithm (1990). For a comparison of the one-pass parallel thinning effects of these algorithms, the postprocessing step for staircase elimination in Holt's algorithm is not performed in this experiment. Since Chin's algorithm and Chen and Tsai's algorithm did not converge in a small amount of time when the noise cleaning templates were applied, the thinning process was terminated after 8 iterations. As indicated in the results, Holt's algorithm obtains an imperfectly 8-connected skeleton. Due to the use of some trimming templates, Chin's algorithm produces excessive erosion, as indicated in the upper part of the result. Chen and Tsai's algorithm obtains an imperfectly 8-connected thinning result. On the other hand, as shown in (b), the thinning result of the proposed algorithm is again perfectly 8-connected, noise-insensitive, and topologically
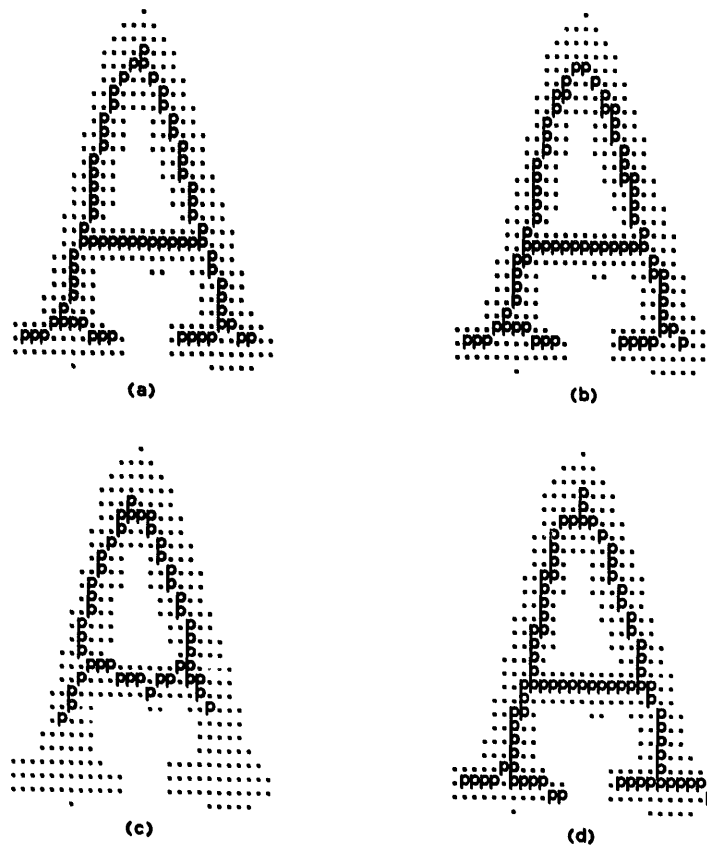

Figure 7. Comparison of one-pass parallel thinning algorithms. (a) The results of the proposed algorithm. (b) The result of Holt's algorithm. (c) The result of Chin's algorithm. (d) The result of Chen and Tsai's algorithm.

equivalent to the original object shape without extensive erosion. Similar results can also be observed in an experiment of thinning an English character 'A' in dimension of 32×32. Figure 7 shows the results of this experiment.

4. Conclusion

A parallel thinning algorithm with good performance has been proposed in this paper. The proposed algorithm uses neither multiple passes nor restoring templates to preserve lines with two-pixel width. With a new set of matching templates, perfectly 8-connected skeletons can be obtained after the thinning process. The problems of excessive erosion and noise sensitivity faced by many other algorithms are also solved in the proposed algorithm. Furthermore, because the thinning operation removes boundary points symmetrically in all directions, the obtained skeleton is isotropic.

That is, the skeleton runs along the medial axis of the original object. This symmetrically removing scheme also eliminates more points in each iteration so that the thinning process can be completed in fewer iterations.

References

Chen, C.-S. and W.-H. Tsai (1990). A new fast one-pass thinning algorithm and its parallel hardware implementation. *Pattern Recognition Lett.* 11, 471–477.

Chin, R.T., H.-K. Wan, D.L. Stover and R.D. Iverson (1987). A one-pass thinning algorithm and its parallel implementation. *Computer Vlision, Graphics, and Image Processing* 40, 30–40.

Holt, C.M., A. Stewart, M. Clint and R.H. Perrott (1987). An improved parallel thinning algorithm. *Comm. ACM* 30 (2), 156–160.

Stefanelli, R. and A. Rosenfeld (1971). Some parallel thinning algorithms for digital pictures, *J. ACM* 18 (2), 255–264.

Zhang, T.Y. and C.Y. Suen (1984). A fast parallel algorithm for thinning digital patterns. *Comm. ACM* 27 (3), 236–239.