Technical Note

Image Processing and Computer Vision

Robert M. Haralick Editor

A Comment on "A Fast Parallel Algorithm for Thinning Digital Patterns"

H. E. LÜ and P. S. P. WANG

ABSTRACT: A fast parallel thinning algorithm for digital patterns is presented. This algorithm is an improved version of the algorithms introduced by Zhang and Suen [5] *and Stefanelli and Rosenfeld* [3]. *An experiment using an Apple II and an Epson printer was conducted. The results show that the improved algorithm overcomes some of the disadvantages found in* [5] *by preserving necessary and essential structures for certain patterns which should not be deleted and maintains very fast speed, from about 1.5 to 2.3 times faster than the four-step and two-step methods described in* [3] *although the resulting skeletons look basically the same.*

1. INTRODUCTION

"Thinning" plays a very important role in the preprocessing stage of pattern recognition [1, 2, 4]. It deals with extracting the distinctive features known as "skeletons" from the patterns. On one hand, it gets rid of all redundant points: on the other hand, it preserves the basic structure and characteristics of the pattern.

In [5] a fast parallel algorithm for thinning digital patterns is proposed. Using this algorithm, each pattern can be thinned down to a skeleton of unitary thickness, and experimental results show that this algorithm is very good with respect to both connectivity and contour noise immunity. Further, it is from 1.5 to 2.3 times faster than the four-step and two-step methods introduced in the literature [3].

This research was partially supported by the National Science Council, Taipei, Taiwan.

Unfortunately, however, there are some errors, unclarity, and disadvantages in the algorithm. This note will point them out and propose improvements to some of them.

2. DEFINITIONS, TERMINOLOGIES, AND THE ALGORITHM

We adopt the terminologies and definitions of [5].

A binary digitalized picture is defined by a matrix IT where each pixel IT(i, j) is either 1 or 0. The pattern consists of those pixels that have the value 1. It is assumed that the neighbors of the point (i, j) are $(i-1, j)$, $(i-1, j+1)$, $(i, j+1)$, $(i+1, j+1)$, $(i+1, j)$, $(i+1, j-1)$, $(i, j-1)$, and $(i-1, j-1)$, as is shown in Figure 1. In parallel picture processing, the new value given to

P_1 $(i-1, j-1)$	P_2 $(i-1, j)$	P_3 $(i-1, j+1)$
P_1 $(i, j-1)$	P_1 (i, j)	P_4 $(i, j+1)$
P_1 $(i+1, j-1)$	P_6 $(i+1, j)$	P_5 $(i+1, j+1)$

FIGURE 1. Designations of the Nine Pixels in a 3 × 3 Window.

a point at the nth iteration depends on its own value as well as those of its eight neighbors at the $(n-1)$th iteration, so that all picture points can be processed simultaneously. It is assumed that a 3×3 window is used and that each element is connected to its eight neighboring elements.

In the algorithm, each iteration is divided into two subiterations. In the first subiteration, the contour point P_1 is deleted from the digital pattern if it safisfies the following conditions:

(a) $2 < B(P_1) < 6$ (c) $P_2^*P_4^*P_6 = 0$

(b) $A(P_1) = 1$ (d) $P_4^*P_6^*P_8 = 0$

where $A(P_1)$ is the number of 01 patterns in the ordered set $P_2, P_3, \ldots, P_8, P_9$ that are the eight neighbors of P_1 (Figure 1), and $B(P_1)$ is the number of nonzero neighbors of P_1, that is: $B(P_1) = P_2 + P_3 + P_4 + \cdots + P_8 + P_9$.

In the second subiteration, only conditions (c) and (d) are changed (Figure 3) as follows:

(c′) $P2^*P4^*P8 = 0$ (d′) $P2^*P6^*P8 = 0$

and the rest remain the same.

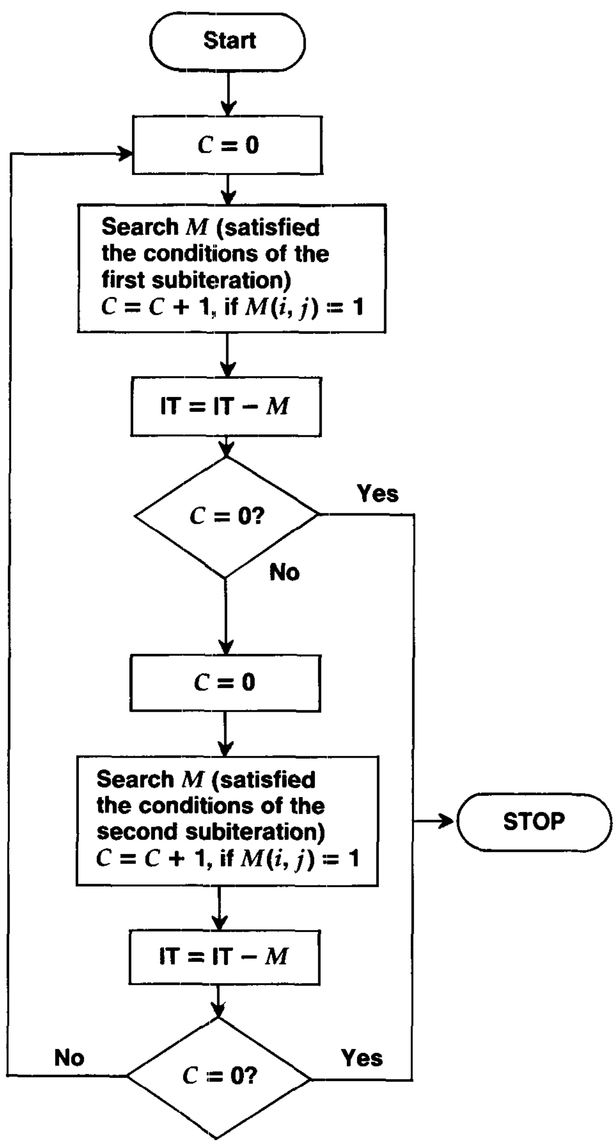


FIGURE 2. Flowchart of the Thinning Algorithm.

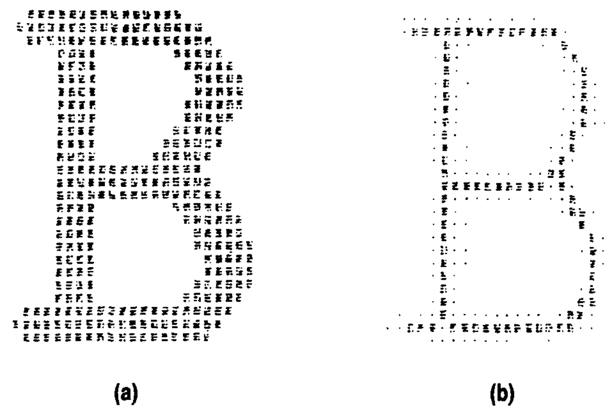


FIGURE 3. An Example. (a) Original pattern. (b) Correct thinned solution.

3. ERRORS AND UNCLARITY: DISADVANTAGES AND PARTIAL SOLUTIONS

1. The first comment is that there is an error in the flowchart of the algorithm [5, Figure 6]. The loop arrow from the bottom box C = 0 should return to the top box C = 0 rather than the box containing search M, that is, the correct algorithm should be as in Figure 2.

2. The second comment is a matter of clarity. According to the thinning algorithm [5, Figure 6]. The first subiteration comes before the second subiteration. Therefore the output of [5, Figure 7b] should be as in Figure 3 rather than in Figure 4. Figure 4 is actually the output of the algorithm in which the first subiteration

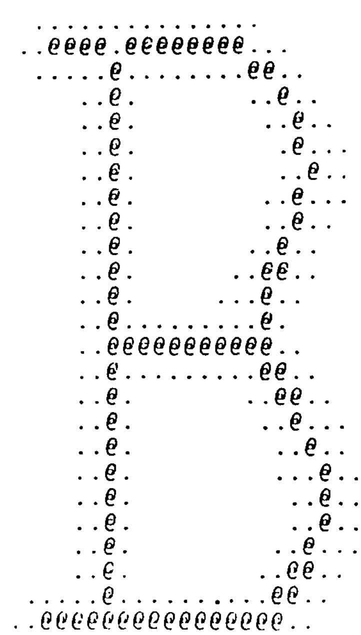


FIGURE 4. The Output of [4].

and the second subiteration are swapped. It is a matter of the different orders used in the subiterations, which should be clarified.

In addition, there are some disadvantages that cannot be neglected:

1. Noise that should be eliminated is instead enlarged. Suppose there is a very small noise to the northeast end of an "H", shown in Figure 5. Applying the algorithm of [4] will result in Figure 6. The noise not only is not eliminated but is worse after thinning.

2. Structures that should be preserved are destroyed. This can be seen in Figure 7. In general, diagonal segments with thickness 2 will eventually be reduced to only one or two points as shown in Figure 8.

3. Even worse, some digital patterns will totally disappear as shown in Figure 9. In general, digital patterns that can be reduced to a square consisting of four dots : : will be totally gone.

One solution to Problem 2, discussed above, is to modify the condition $2 \le B(P_1) \le 6$ to $3 \le B(P_1) \le 6$. This takes care of the endpoints which should not be eliminated as shown in Figure 10 (p. 242).

An experiment using an Apple II and an Epson printer has shown that Figure 8a becomes Figure 11a (p. 242) and Figure 9a is reduced to Figure 11b, c

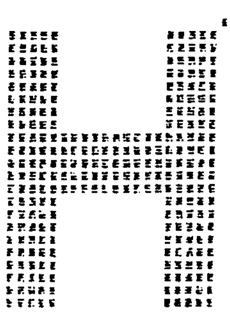
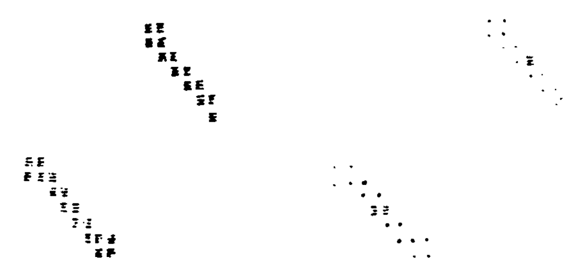


FIGURE 5. An "H" With a Tiny Noise.

(a) (b)

FIGURE 7. Structure That Should Be Preserved Is Destroyed. (a) Original digital pattern. (b) Skeleton after six iterations.



(a) (b)

FIGURE 8. A Diagonal Line Segment With Thickness 2. (a) Original digital patterns. (b) Skeleton after five iterations.

(a) (b) (c)

FIGURE 9. A Pattern Disappeared In Two Iterations.

although the results of other digital patterns such as "H", "B", "體", and the moving body are essentially the same as [5].

4. SUMMARY

An improved parallel algorithm for thinning digital patterns is presented in this note. It overcomes some disadvantages and difficulties of the method described in [5] while maintaining essentially the same fast speed,

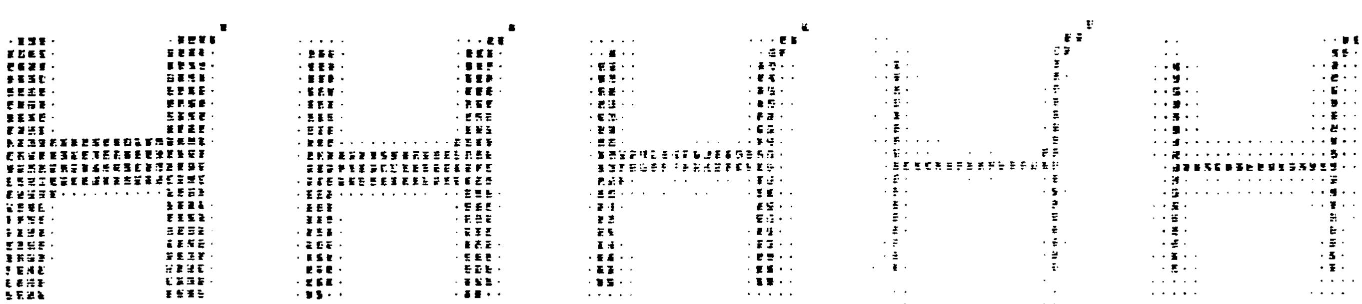


FIGURE 6. "H" After Five Iterations.

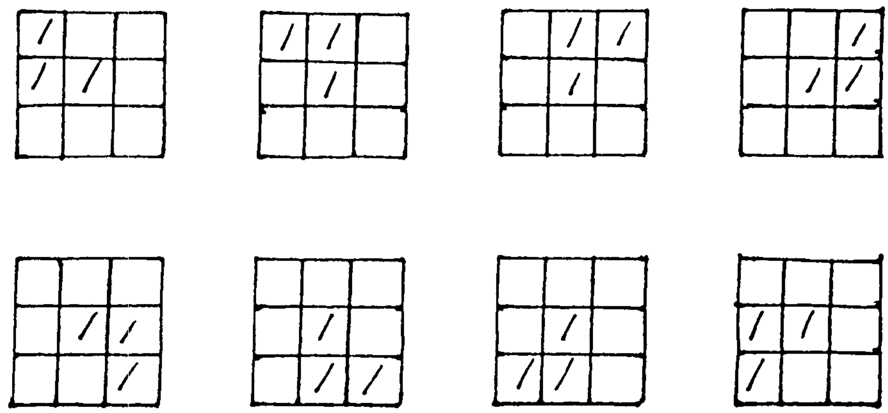
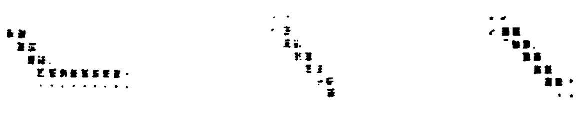


FIGURE 10. Endpoints Which *Should Not* Be Deleted.



(a) (b) (c)

FIGURE 11. Structure Is Preserved.

about 1.5 to 2.3 times faster than other methods introduced in [3]. However, the noise immunity and total disappearance of patterns remain as an open problem.

Acknowledgments. This work was done when the second author, P.S.P. Wang, was a visiting professor at both Departments of Electrical Engineering and Information Engineering, National Taiwan University. The support from both department chairmen C.H. Chen and L.S. Lee is greatly appreciated.

REFERENCES

1. Fu, K.S., and Rosenfeld, A. Pattern recognition and image processing. *IEEE Trans. Comput. C-25*, 12 (Dec. 1976), 1336–1346.
2. Pavlidis, T. A flexible parallel thinning algorithm. In *Proceedings of IEEE Computer Society Conference on Pattern Recognition and Image Processing*. Aug. 1981, 162–167.
3. Stefanelli, S., and Rosenfeld, A. Some parallel thinning algorithms for digital picture. *J. ACM 18*, (Apr. 1971), 255–264.
4. Wang, P.S.P., Ed. *Intelligent Systems, Imaging Technology and Software Engineering*. Sungkang Computer Book Co., Taipei, Taiwan, 1984.
5. Zhang, T.Y., and Suen, C.Y. A fast parallel algorithm for thinning digital patterns. *Commun. ACM 27*, 3 (Mar. 1984), 236–239.

CR Categories and Subject Descriptors: I.4 [**Image Processing**]; I.5.2 [**Pattern Recognition**]: Design Methodologies—*pattern analysis*, I.5.4 [**Pattern Recognition**]: Applications—*computer vision*
General Terms: Algorithms, Theory
Additional Key Words and Phrases: applications, parallel algorithm, thinning digital patterns, skeletonization

Received 1/85; revised 10/85; accepted 11/85

Authors' Present Addresses: H. E. Lü, Department of Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. P.S.P. Wang, College of Computer Science, Northeastern University, 360 Huntington Avenue, Boston, MA 02115.