IMPROVED PARALLEL THINNING ALGORITHM TO OBTAIN UNIT-WIDTH SKELETON

Jun-Sik Kwon

Department of Electrical Engineering, Se-myung University, Jechon, Korea[1]
jskwon@semyung.ac.kr

ABSTRACT

To extract the creditable features in a fingerprint image, many people use a thinning algorithm that plays a very important role in preprocessing. In this paper, we propose a robust parallel thinning algorithm that can preserve the connectivity of the binarized fingerprint image, while making the thinnest skeleton of only 1-pixel wide, which gets extremely close to the medial axis. The proposed thinning method repeats three sub-iterations. The first sub-iteration takes off only the outermost boundary pixel using the inner points. To extract the one-sided skeletons, the second sub-iteration seeks the skeletons with a 2-pixel width. The third sub-iteration prunes the needless pixels with a 2-pixel width existing in the obtained skeletons. The proposed thinning algorithm shows robustness against rotation and noise and makes the balanced medial axis. To evaluate the performance of the proposed thinning algorithm, we compare it with and analyze previous algorithms.

Keywords

Fingerprint image, Parallel thinning, Connectivity preservation, Unit-width skeleton, Medial line approximation

1. INTRODUCTION

In recent times, the fast spread of information and communication equipments raises the needs for security. From this security point of view, the development of personal authentication technology which secures the reliance and stability is a very important issue. At present, personal authentication methods include inherent risk of leakage, loss, oblivion, and surreptitious use. To solve the risk of forgery, many researchers have recently vigorously studied the biometrics-based personal identification and authentication systems using bodily features. Bodily features are divided into two classes: 1) physical properties – including fingerprint, palm print, iris, hand geometry, retina, face, and vein. 2) behavioral properties –including signature, voice, typing and more. Among these features, fingerprint recognition technologies have the lowest risk of copy and spillage, highest stability, simple usage, low cost, and fit in with the realizing means [1-2].

It is recognized that the use of fingerprint is difficult because incorporated in the image is the hard-to-make-decision criterion coordinate. The flexibility of the fingerprint makes changed and damaged lower quality images that often use noise for identification[3]. To solve the problem, we employed the thinning algorithm that facilitates the extraction of the structural feature of a fingerprint. Since the fingerprint image is of regular width and the flow follows a gentle sloped ridge, it is very proper to indicate a one-pixel width skeleton[3-4]. As a result, the preservation of the fingerprint information greatly facilitates extraction of the fingerprint features for identification purpose.

DOI : 10.5121/ijma.2013.5201

Thinning is divided into iterative and non-iterative methods. The non-iterative method, like human intuition and perception, extracts the information in the geometric structure of the image signal from various circumstances. This method prioritizes the possibility of restoration rather than connectivity preservation of the object [5]. Furthermore iterative thinning is classified into sequential [6] and parallel methods [7-13]. In every iteration steps, the sequential thinning algorithm applies all previous computed results to a present computing step. The method has a lower cost of computing complexity than other methods. However the result is biased by the scanning direction, because the skeletons of objects are dependent on the scanning direction. In the parallel thinning algorithm, all pixels involving an iterative computing step are covered independently. The method has merit in that it presents the skeleton near the center of objects, but demerits in that the two-pixel width pattern is removed at one time.

There are many existing thinning algorithms, which are associated with the differences of their applications and performances. Thus, the definition and requirements of thinning are also different. In spite of the requirement of diversity, the following requirements are generally and common to all algorithms [8],[14-17].

Requirement 1: Connected image regions should thin to connected line structures (connectivity preservation);
Requirement 2: Approximate end-line locations should be maintained (no excessive erosion);
Requirement 3: The thinning results should at a minimum be 8-connected (unit-width skeleton);
Requirement 4: The thinning results should approximate the medial lines (medial line approximation); and
Requirement 5: Extraneous spurs caused by thinning should be minimized (boundary noise immunity).

However, every thinning algorithm did not satisfy the important thinning requirements perfectly. In particular, the local operation processed by the parallel 3×3 mask is least costly; but, the connectivity of the pattern is not guaranteed. Further, a local operation with extended masks with measurement such as 3×4, 4×3, 4×4, or 4×5, are processed occasionally [5],[7],[13]. In the thinned image, the objects achieve 8-neighbor connectivity and include extra pixels (not located at the end-point, existing in the center, and even if removed, does not break the lines).

In this paper, the proposed thinning algorithm searches the center line of the object and removes only the outermost boundary pixels, therefore it is robust to rotation and noise, while maintaining a balanced center line.

The proposed algorithm is described in chapter II. We compare the proposed algorithm with the proven good performance of the existing parallel thinning algorithms – ZS algorithm[7], CCS algorithm[8], GH algorithm[9], LW algorithm[10], MPS algorithm[11], KG algorithm[12], and AW algorithm[13] - and evaluate them.

2. PARALLEL THINNING ALGORITHM USING CONSIDERATION OF INNER PIXEL

In this paper, we propose a robust parallel thinning algorithm which can preserve the connectivity of the object, making the thinnest skeleton with a one-pixel width and gets very close to the medial axis. The proposed thinning method repeats three sub-iterations. All sub-iterations each have an aim, and the iteration is repeated until the aim is accomplished.

The aim of the first sub-iteration is searching the inner pixels with the boundary mask and removing only the outermost contour pixels, leaving a two-pixel width. Even if the object is rotated, the skeleton pattern caused by thinning is also rotated as same angle. The aim of the second sub-iteration is to find the two-pixel width skeleton formed from the first sub-iteration and to extract a one-pixel width skeleton, therefore resulting in the medial line being closer to object. The aim of third sub-iteration is the removal of the extra two-pixel width skeleton formed in the previous sub-iteration, resulting skeleton in a 1-pixel or unit-width skeleton. For that reason, the parallel thinning algorithm is processed sequentially.

The fundamental mask used in the proposed algorithm is a 3×3 mask. But, low efficiency causes the removal pattern decision mask uses the more neighbor pixels than the local 3×3 mask. The extended mask used in the proposed algorithm is 4×4 or 5×5 mask, but as the case, to make an exact decision of the pixel of interest(Pi), the mask is variable. And this algorithm is possible to perform the each sub-iteration step with tree structure due to the nature of mask pattern. This process improves the problem of speed decline due to many complex computing.

For example, the boundary mask conditions 1) and 2), suggested in next section, the mask with the number of neighbor pixels N(Pi)=1, N(Pi)=2, ..., N(Pi)=7 is combined with eight pure boundary masks respectively. Additionally, a boundary pixel of an object satisfied in a 3×3 mask is only matched with other suggested masks. So, by computing the condition leading to the decision of removal, originating from the satisfied boundary pattern mask, the result is applied to a pixel of interest (Pi) and moved on to the next pixel. The process is shown in Figure1 and C is any counter here.

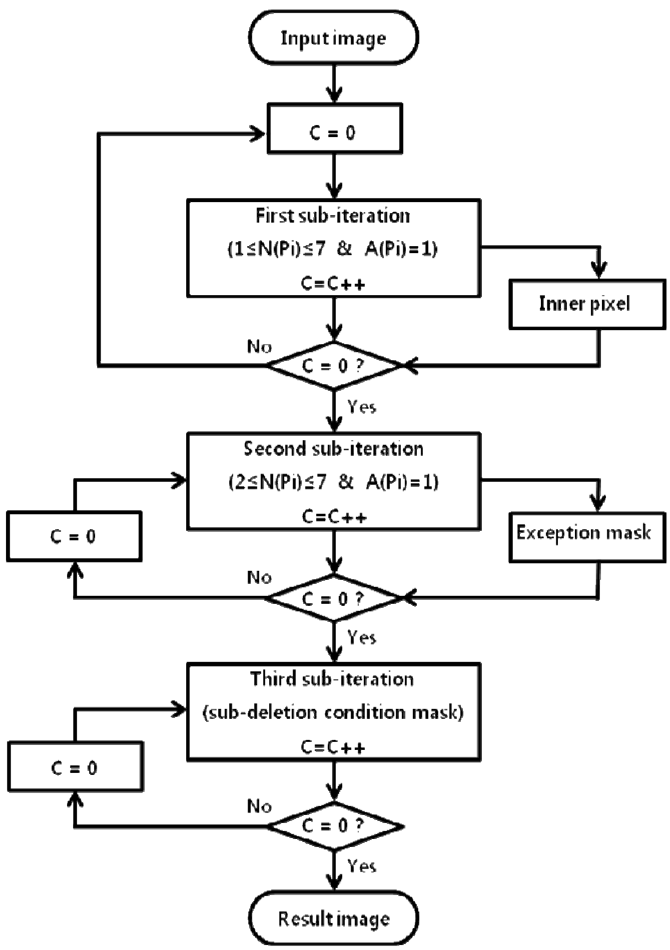

Figure 1. Flowchart of the sub-iteration modified by the tree structure

2.1. Outermost boundary pixel

The outermost boundary pixels of object are the set of boundary chain of background and foreground pixels. According to Figure 2(a), 255 boundary masks should be generated with a 3×3 mask, including the pixel of interest (Pi). Though we remove the pixel of interest (Pi) in the generated mask, only the pure boundary pixels that preserve the area of the object are applied in our algorithm. Thus, the sum [N(Pi)] of 8-neighbor pixels is 1 to 7. In Figure 2(b), only the pixel of interest (Pi) where the number [A(Pi)] of pixels with a change of 0→1 pattern (here, 0: background, 1: foreground) is 1 are defined as a boundary pixel.

Pure boundary mask: 1) $1 \leq N(Pi) \leq 7$
2) $A(Pi)=1$

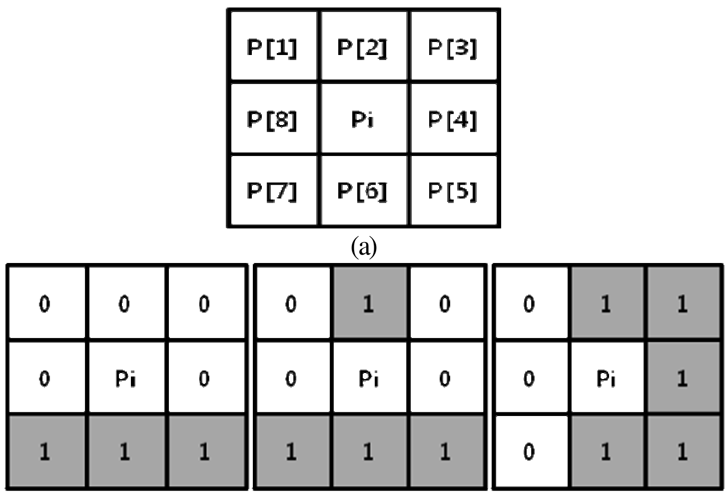

Figure 2. (a) Pixel P(i) and its mask and (b) Number of 0→1 Patterns

2.2. Probe into the inner point of neighbor pixels

This algorithm removes only the pixels located at the pure boundary of the object. If neighbors of the pixel of interest (Pi) are located at the pure boundary and had more than one inner pixel, the pixel of interest (Pi) does not fallen upon medial line and is the pure outermost boundary pixel. Thus, as depicted in Figure 3, the medial line of an object is located inside an object and neighbors of the pixel of interest (Pi) are located at the pure boundary and have one or more inner pixels. In Figure 3, the three dotted lines are inner pixels of object and are corresponding skeletons. As shown in Figure 4, the combination of two relevant masks breaks off the connectivity of skeletonized lines. When the number of background pixels from neighbor pixels is one and all other pixels belong to the domain of the object, the phenomenon where the 'ㅁ' and 'ㅂ' shapes are eroded into 'ㅇ', is prevented. But, if the corresponding mask is not verified as the pure boundary pixel, it is not supposed to be removed. This pixel of interest (Pi) is removed without any further operations. However the case of , , ,  is an exception; here white is a background pixel of the object whereas black is the foreground. In this case, with only one inner pixel, we cannot make the decision that Pi is the pure outermost boundary pixel of the object and the number of inner pixels is defined as more than five pixels, exceptionally. The probe into inner point is applied in the first sub-iteration of the proposed algorithm and only the pure boundary pixel of the object is removed.

0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0
0	0	1	1	1	0	1	0	0
0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0
0	0	1	0	1	1	1	0	0
0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3. The case where the inner-point is also the skeleton of the objects

0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0

Figure 4. The case of broken connectivity

2.3. Exception mask and removal decision mask.

In this algorithm, to characterize the skeletonized line with the intrinsic structure of object is insufficient and requires some sub-cycle to be supplemented to satisfy the purpose of each sub-iteration. The end point of the skeletonized line formed by thinning should not be shrunk infinitely or not preserve the connectivity. The pure boundary masks as , , , will be excluded from operation. But, by supplementing the following condition, the diagonal end point patterns as , , , , remove the pixel regarded as noise.

Diagonal end point pattern mask condition : if following logic operation condition is contended, the pixel of interest (Pi) is removed.

ABC’D’E’ + A’B’E’(C + D) + CDE=1

A, B, C, D, E of Figure 5 symbolizes the corresponding pixel shown in Figure 4. When the pure boundary masks are the boundary pattern masks , , , , , , , , the pixel of interest (Pi) is removed, because it lies on the boundary of the object. However if it is the end point of the zigzag stair pattern, or the horizontal or vertical 2 pixel-width line, it should not be removed, because it is located at the center of the skeleton. Thus only the outermost boundary pixel could be removed.

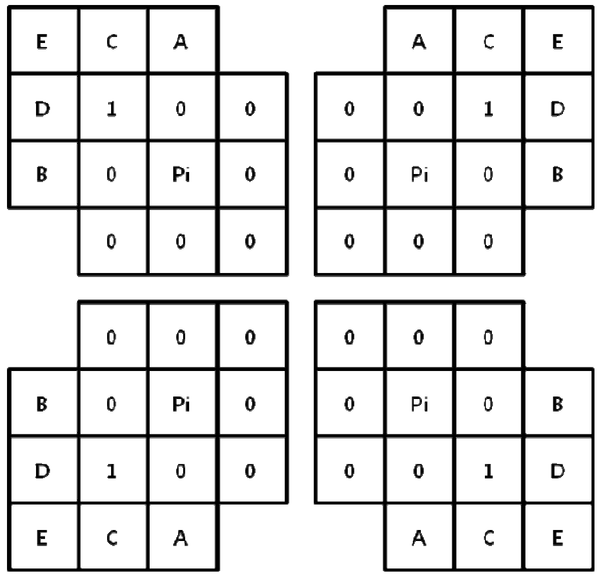

Figure 5. End-point masks of diagonal direction

The mask considered in the pure boundary mask is the horizontal or vertical 2 pixel-width straight line. In this case, the pixel of interest (Pi) located consecutively on the line of one side, similarly established consecutive removal to make the center line a one-pixel width. Further consideration of the successively horizontal or vertical length builds the balanced center line from all directions. Thus, , , , is applied to the mask by calculating the successive length, and , , , is removed by considering the neighboring pixels. The former belongs to the second sub-iteration condition, and the latter is performed in all sub-iteration conditions. In the case of the square pattern and two-pixel width, the latter makes a one-pixel width according to the final sub-iteration condition.

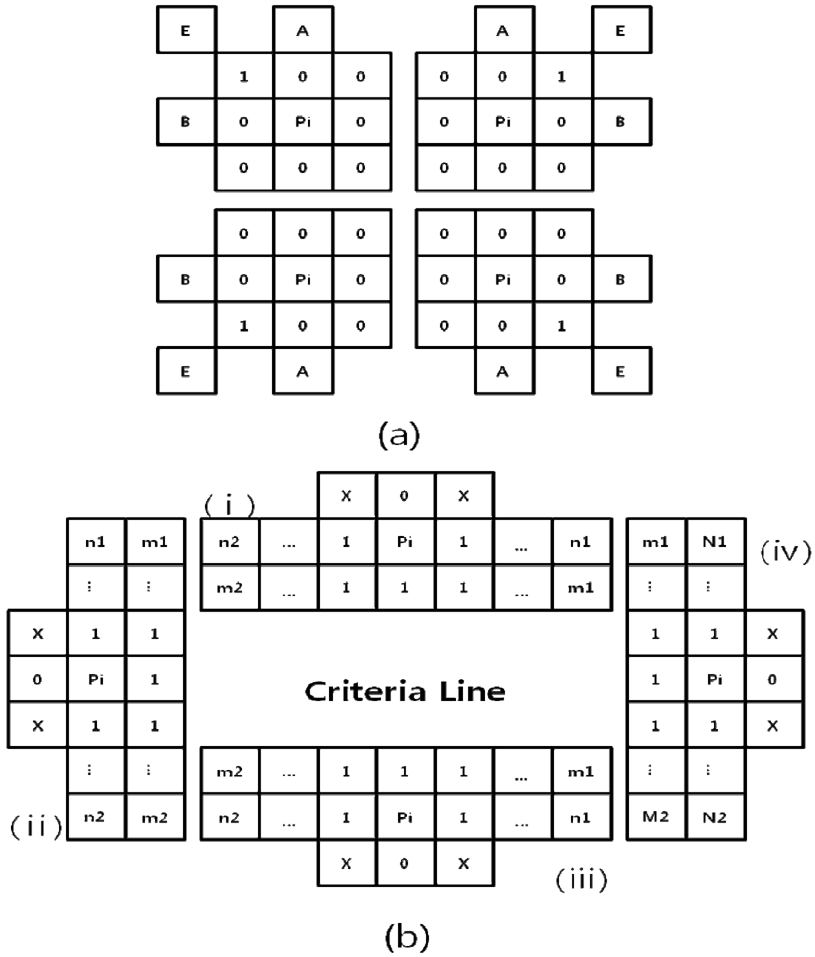

Figure 6. (a) Square pattern masks and (b) Horizontal and vertical pattern masks, × denotes "don't care"

Square pattern condition: pixel (Pi), satisfying the following logic operation, is removed.
AB + AE + BE=1
Here, A, B, E is the corresponding pixel in Figure 6(a).

The directional length information of the horizontal and vertical pattern is the sum of the number of object pixels and is expressed in formula (1)

$$CS1 = \sum_{k=1}^{n1} k,\ CS2 = \sum_{k=1}^{n2} k,\ CS1' = \sum_{k=1}^{m1} k,\ CS2' = \sum_{k=1}^{m2} k \quad (1)$$

In Figure 6(b), CS1, CS2, CS1´ and CS2´ is the sum of number of object pixels appearing until it meets the background pixel in the direction n1, n2, m1 and m2 around the base line. By using the directional length information CS1, CS2, CS1´ and CS2´ of the horizontal and vertical pattern obtained in formula (1), the center line is extracted from the following two conditions.

Horizontal and vertical pattern condition:

① in (i), (iii) of Figure 4 (b) the pixel (Pi) satisfying the condition is removed.
$CS1 \le CS1' \;\&\; CS2 \le CS2' = 1$

② in (ii), (iv) of Figure 4 (b) the pixel (Pi) satisfying the condition is removed.
$CS1 \ge CS1' \;\&\; CS2 \ge CS2' = 0$

2.4. Center pixel to be removed

The result image by thinning should consisted of the skeletonized line with a one-pixel or unit width and be contented with the 8-neighbor connectivity. Thus, the condition that preserves the connectivity of objects and the unit-width, in spite of removal of the pixel (Pi) with 2-pixel width, is added to thinning algorithm. This algorithm suggests the following sub-deletion condition masks. It operates repeatedly by third sub-iteration condition and removes the pixel (Pi) satisfying the condition. The sub-deletion condition mask is shown in Figure 7 and the definition of mask may be summarized as follows.

Definition of sub-deletion condition masks: if anyone of following conditions is satisfied, the pixel (Pi) should be removed.

① P[2]*P[4]=1 & P[7]=0 .

② P[4]*P[6]=1 & P[1]=0.

③ P[6]*P[8]=1 & P[3]=0.

④ P[2]*P[8]=1 & P[5]=0.

X	1	X	0	X	X
X	Pi	1	X	Pi	1
0	X	X	X	1	X
X	X	0	X	1	X
1	Pi	X	1	Pi	X
X	1	X	X	X	0

Figure 7. Masks of the sub-deletion condition (× denotes "don't care")

The sub-deletion condition masks should not be applied to the parallel processing because the completion of two iterations of the thinning stage for this algorithm, preserves the connectivity of the end point and the image result. Thus, the sub-deletion condition masks, combined with the stair-type end point preservation masks, should be applied to the object sequentially. In Fig 8 (a), if the pixels of the left or right diagonal lines are removed, connectivity should be maintained. In Fig 8 (b) and (c), even if the pixel (Pi) is removed, connectivity should not be disrupted and in Figure 8 (d), even if one (Pi) of the two pixels is removed, connectivity should by no means be disrupted.

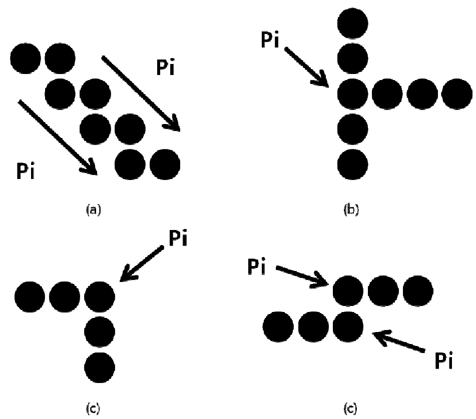

Figure 8. Center pixels connected to a two-pixel width

3. Comparison And Evaluation of Experimental Results

Our new algorithm (NEW) described in chapter II is compared with the previous algorithms (ZS, LW, MPS, KG, GH, CCS, AW) and evaluated. The various image patterns, as an alphabetic letter, numeric digit, symbol, figure, and fingerprint are used for the performance evaluation of each algorithm. A fingerprint taken is applied to the experimental image of the new thinning algorithm after going through two preprocessing phases – smoothing and 9×9 to block the binarization of the original image.

The performance of existing parallel thinning algorithms is evaluated by considering the satisfaction of the general requirements suggested in the introduction.

Requirement 1) connectivity preservation.

In only the AW and MPS algorithm of the suggested algorithm, the connectivity of specific patterns is injured. In an experimental image, it is shown that it is the AW algorithm in the square pattern with even-numbered pattern-width cannot preserve the connectivity. (Figure 9, Figure 10, and Figure 11)

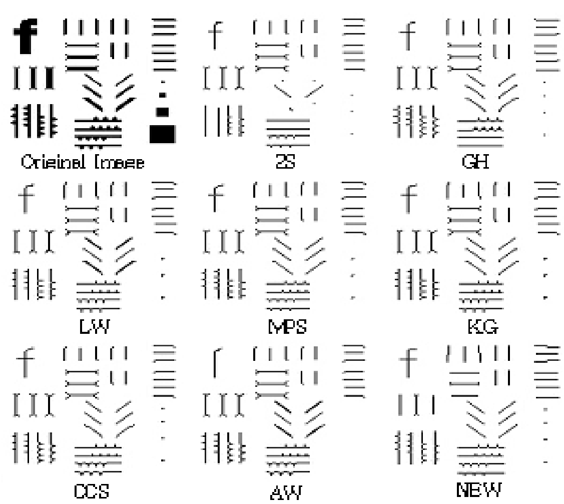

Figure 9. Thinned results of the horizontal, vertical, diagonal lines and squares

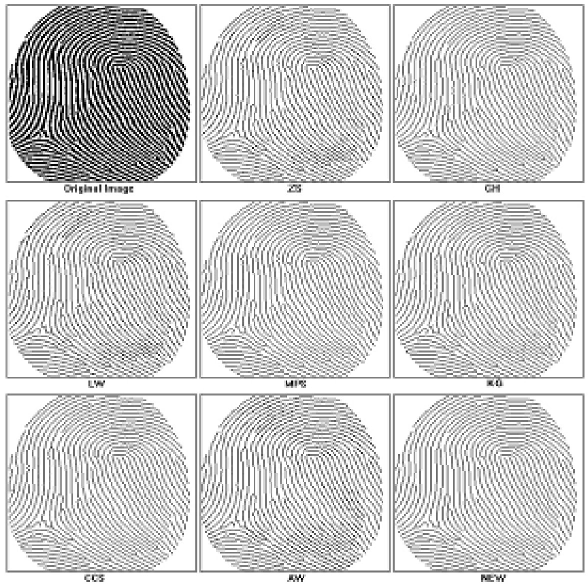

Figure 10. Experimental results of fingerprint image 1

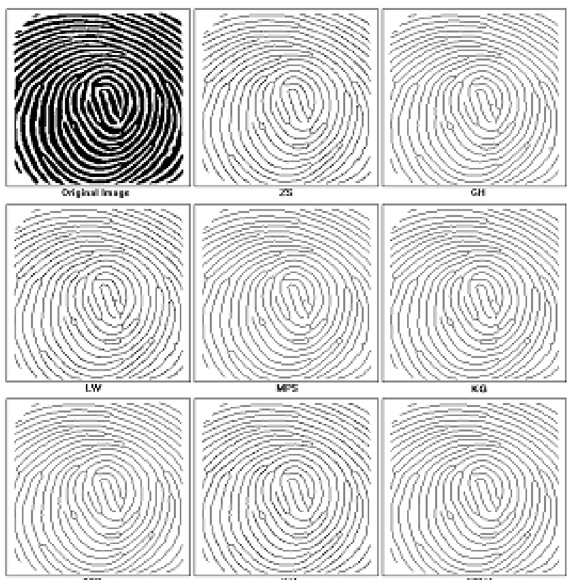

Figure 11. Experimental results of fingerprint image 2

Requirement 2) no excessive erosion

The images applied in our experiments show different degrees of erosion in all thinning algorithms. In particular, the ZS algorithm excessively erodes the diagonal pattern, while the GH, CCS and AW algorithms show excessive erosion of specific patterns. (Figure 9 and Figure 12)

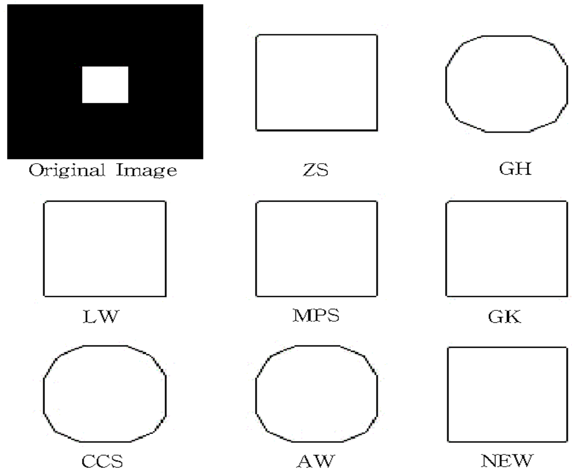

Figure 12. Thinned results of the'□' shaped image

Requirement 3) unit-width skeleton
To satisfy requirement 3) generally, the parallel thinning algorithms accompany the appurtenant sub-iteration conditions. The suggested new algorithm uses three phased sub-iteration conditions and shows perfect 8-connectivity with unit-width skeleton under any image. In particular, for previous algorithms such as the LW algorithm, the many redundant pixels remain on the skeletonized line, and to remove it, the MPS and KG algorithms are developed to have the appurtenant conditional clause. However, experimental results show that these algorithms cannot produce a perfect unit-width skeleton because the skeletonized lines made in the iteration phases under the main condition that already has a 2-pixel width which does not conform to the appurtenant conditions. But by using the appurtenant conditions in most thinning algorithms, many redundant pixels are removed, especially the skeletonized line of fingerprint image, which has certainly improved.

Requirement 4) medial line approximation.
So far, there is no clear definition of a good skeleton and thus it is not easy to evaluate. But when the performance of thinning algorithms is generally evaluated, a similar reference skeleton is obtained by non-specialists, since it is compared and evaluated. Our experimental images show the GH algorithm, which sometimes got out of the medial line of the object.

Requirement 5) boundary noise immunity.
It is shown that eight algorithms are all robust in the image with one-pixel noise existing at the boundary. However the previous algorithms cannot process to remove one-pixel noise existing at the corner, so the final skeletonized line has many distortions. The suggested new algorithm produces the improved result image.

Generally, as Figure 10 and Figure 11 show, the fingerprint image is preprocessed by thinning, whereas the singular-points are analyzed by core and delta, and lastly, the minutiae as an ending and bifurcation-point, should be extracted. Thus, in the process of thinning, the many unintentional pseudo minutiae should not be produced. As shown in Figure 11 and Figure 12, GH, LW, MPS, KG and CCS algorithms produce the many unintentional pseudo minutiae.

Table 1 shows the experimental results of a fingerprint image by all algorithms and Table 2 summarizes whether the general requirements of thinning are satisfied or not. The number of iterations in all algorithms is shown in Table 1. The suggested algorithm requires many more conditions than in previous algorithms, however because of the operation with a tree structure, the number of iterations is at a rate comparable with that in other algorithms. As a result, in the image result of the suggested algorithm, the redundant pixels do not remain, and the number of medial line pixels is small, As a result image of other algorithms, the redundant pixels always remain, even though the number of medial line pixels is large. In GH, MPS and CCS algorithm, even though very few redundant pixels remained, the perfect unit-width skeletonized line was not produced.

Table 1. Experimental data of the 8 parallel thinning Algorithms

		algorithm							
		ZS	GH	LW	MPS	KG	CCS	AW	NEW
fingerprint image 1	S_O	29389	29389	29389	29389	29389	29389	29389	29389
	N_I	7	6	6	7	8	5	4	8
	S_M	8821	7906	8930	7843	8020	7873	8849	7840
	S_R	1909	8	1970	1	154	5	1963	0
fingerprint image 2	S_O	34137	34137	34137	34137	34137	34137	34137	34137
	N_I	9	14	9	10	11	8	8	11
	S_M	6342	5550	6405	5546	5623	5533	6162	5048
	S_R	1592	2	1636	0	68	12	1273	0

Note. S_O : number of pixels in original pattern
N_I : number of iterations
S_M : number of pixels remaining in the medial line
S_R : number of redundant pixels

Table 2. Performance comparison of the 8 parallel thinning algorithms

Algorithm	Connectivity preservation	Excessive erosion	Unit-width skeleton	Medial line approximation	Boundary noise immunity
ZS	Yes	Yes	Many redundant pixels	Yes	+
GH	Yes	Yes	Few redundant pixels	No	+
LW	Yes	No	Many redundant pixels	Yes	+
MPS	No	No	Tiny redundant pixels	Yes	+
KG	Yes	No	Few redundant pixels	Yes	+
CCS	Yes	Yes	Few redundant pixels	Yes	+
AW	No	Yes	Many redundant pixels	Yes	+
NEW	Yes	No	No redundant pixels	Yes	*

¤ + remarks removal of horizontal and vertical boundary noise, and * remarks removal of horizontal, vertical and diagonal boundary noise.

4. CONCLUSIONS

The parallel thinning algorithm suggested in this paper obtains the outermost boundary pixel of an object using the local mask. Under removal decision using the inner point, the extended local masks, and variable masks, only the pure boundary pixels are removed to produce the image result.

Usage of the variable mask is unsuitable for measuring the computation complexity of the algorithm, but it reveals important information regarding the formation of the most suitable skeleton for approaching the medial line. Additionally, the suggested algorithm has more intricate demerits than other algorithms, and for this reason, it can be used for comparison of the weight of the mask with each of the other masks or condition contributing to the decision of removal. However it is the repetition of the simple operation, since the computational complexity or final number of repetitions is not increased significantly. By using the operation of the algorithm with the tree structure, the processing speed is improved despite the number of the mask. Moreover the number of the mask pertaining to one pixel is decreased than previous thinning algorithms. The pixel removed by an operation is also increased than previous thinning algorithms. By using an appropriate mask and removal conditions, we can remarkably improve the quality of the skeletonized line

.

The parallel thinning algorithm suggested in this paper performs well in the extraction of the skeletionized line compared to previous thinning algorithms, since it will be very helpful in fingerprint recognition. In further work related to parallel thinning, we will prove the effectiveness and utilization of the parallel thinning algorithm in various fields. Futher, it will be widely available in image analysis, pattern recognition, and so on.

ACKNOWLEDGEMENTS

The author would like to thank the 2010 sabbatical support of Semyung University.

REFERENCES

[1] A.K. Jain, R. Bolle, S. Pankantil, Biometrics-Personal Identification in Networked Society, Kluwer Academic Publishers, 1999, pp. 1-41.

[2] Gizem A.K. Jain, L. Hong, R. Bolle, On-line Fingerprint Verification, IEEE Trans. Pattern Analysis Machine Intelligence, vol. 19, no. 4 (Apr. 1997) 302-313.

[3] H. Kim, H.I. Kim, Rotation-Scale-Translation-Intensity Invariant Algorithm for Fingerprint Identification, J. IEEK-S, vol. 35, no. 6 (Jun. 1998) 751-913,.

[4] D.S. Ahn, H.I. Kim, Automatic Real-time Identification of Fingerprint Using Block-FFT, J. IEEK-B, vol. 32, no. 6 (Jun. 1995) 909-921.

[5] S.W. Lee, Performance Evaluation of Image Skeletonization Algorithms, J. KIISE, vol. 18, no. 6 (Nov. 1991) 661-671.

[6] S. Suzuki, K. Abe, Sequential Thinning of Binary Pictures Using Distance Transformation, Proc. 8th Int. Conf. on Pattern Recognition, Paris, France, 1986, pp. 289-292.

[7] Y.T. Zhang, C. Suen, A Fast Parallel Algorithm for Thinning Digital Patterns, Comm. ACM, vol. 27, no.3 (1984) 236-239.

[8] S.S.O. Choy, C.S.T. Choy, W.C. Siu, Note New Single-Pass Algorithm for Parallel Thinning, Computer Vision and Image Understanding, vol. 62, no. 1 (Jul. 1995) 69-77.

[9] Z. Gao, W.R. Hall, Parallel Thinning with Two-Subiteration Algorithm, Communications of the Association for Computing Machinery, vol. 32, no. 3 (1989) 359-373.

[10] H.E. Lu, P.S.P. Wang, An Improved Fast Parallel Thinning algorithm for Digital Patterns, Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (1985) 364-367.

[11] A.D. Mandalia, A.S. Pandya, R. Sudhaker, Modified Fast Parallel Thinning Algorithm for Noisy Handprinted Characters, IEEE Proc. of 2nd International Conference on Image Processing, Singapore, Sep. 1992, pp. 7-11.

[12] J.S. Kwon, J.W. Gi, and E.K. Kang, An Enhanced Thinning Algorithm Using Parallel Processing, Proc. of 11th IEEE International Conference on Image Processing, vol. III, Thessaloniki Greece, Oct. 2001, pp. 752-755.

[13] M. Ahmed, R. Ward, A Rotation Invariant Rule-Based Thinning Algorithm for Character Recognition, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 12 (Dec., 2002) 1672-1678.

[14] A. Rosenfeld, J.L. Pfaltz, Distance Functions on Digital Pictures, Pattern Recognition, vol. 1 (1968) 33-61.

[15] C. Arcelli, G.S.D. Baja, A Width-Independent Fast Thinning Algorithm, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 7, (Jul. 1985) 463-474.

[16] H. Tamura, A Comparison of Line Thinning Algorithms from Digital Geometry Viewpoint, IEEE Proc. of 4th International Conference on Pattern Recognition, Kyoto, Japan, 1978, pp. 715-719.

[17] L.O. Gorman, k×k Thinning, Comput. Vision Graphics Image Process., vol. 51 (1990) 195-215.

Authors

Jun-Sik Kwon received the B.S., the M.S. and the Ph.D. degrees from the Chung-Ang University, Korea, all in electronics engineering, in 1984, 1986, and 1995, respectively. He joined the faculty at Se-Myung University in 1995, where he is now a Professor in the Department of Electrical Engineering. His research interests include Pattern Recognition, Computer Vision, and Image Processing