A Novel Single Pass Thinning Algorithm[1]

G.S. Ng

R.W.Zhou and

C. Quek

Contact: Dr. Chai Quek

Submitted to: IEEE Transaction on System Man and Cybernetics

Date: 8th September 1994

Email: ashcquek@ntuix.ntu.ac.sg

Fax: (+65) 791 9414

Keywords: single pass, performance criteria, smoothing templates, performance and complexity analysis, boundary noise, noise immunity, thinning algorithm.

---

[1]Manuscript received:

Affiliation: Nanyang Technological University, School of Applied Science, Research Laboratory II, Nanyang Avenue, Singapore 2263.

Acknowledgment: The authors wish to acknowledge the generous financial support from the Nanyang Technological University, School of Applied Science for funding the AF project. AF is an intelligent toolkit for anti-forgery.

A Novel Single Pass Thinning Algorithm

G. S. Ng, R. W. Zhou, C. Quek

***Abstract**: A new sequential thinning algorithm, which uses both flag map and bitmap simultaneously to decide if a boundary pixel can be deleted, as well as the incorporation of smoothing templates to smooth the final skeleton, is proposed in this paper. Three performance measurements are proposed for an objective evaluation of this novel algorithm against a set of well established techniques. Extensive results comparison and analysis are presented in this paper for discussion.*

1 Introduction

Thinning is an important pre-processing step for many image analysis operations. The main objective of thinning in image processing and pattern recognition are: to reduce data storage while at the same time retaining its topological properties, to reduce transmission time as well as to facilitate the extraction of morphological features from digitized patterns. Thinning reduces the amount of data to be stored by transforming a binary image into a skeleton, or line drawing.

The result of the thinning process is skeleton. It was first introduced by Blum [Blum, 1964], who called it the medial axis. There are many thinning algorithms developed from then on [Arcelli, 1981][Dill & Levine, 1987][Pavlidis, 1982][Xu & Wang, 1987]. These algorithms can be classified into two classes: parallel and sequential thinning algorithms [Rosenfeld & Pfaltz, 1966][Stefanelli & Rosenfeld, 1971]. The main difference between these two types is that sequential thinning operates on one pixel at a time and the operation depends on preceding processed results, whereas parallel thinning operates on all the pixels simultaneously. Most of the thinning algorithms developed were parallel algorithms. Sequential thinning generally produces better skeletons and requires less memories, but parallel thinning is substantially faster. However, a sequential algorithm implemented on serial hardware will be faster than a parallel algorithm.

Ideally a skeleton must be topologically equivalent to the object. If the object is connected, then the skeleton must also be connected. In addition, it must run along the medial axis of the object and must be one pixel thick. Maintaining connectivity is a basic requirement of all thinning algorithms. The ability to handle boundary noise is another property desired in these algorithms. Any compromise in these two important aspects will lead to the generation of skeletons with unpredictable accuracy. As a result, a skeleton should possess the following characteristics:

1) Connectivity should be preserved. It is generally agreed that a skeleton should be 4-connected in background, 8-connected in foreground and is one pixel thick. A break point test is incorporated into many thinning algorithms to prevent disconnectivity.
2) Excessive erosion should be prevented. The end points of a skeleton should be detected as soon as possible so that the length of a line or curve that represents a true feature of the object is not shortened excessively.
3) The skeleton should be immune to small perturbations in the outline of the object. Noise, or small convexities, which do not belong to the skeleton, will often result in a tail after thinning. The length of these tails should be minimized. For this purpose, the deletion of loop points, which will introduce extra holes in the skeleton, should be avoided.

In this paper, a novel single pass thinning algorithm that uses both flag map and smoothing templates for boundary pixel deletion is proposed. The skeleton produced by this algorithm is not only one-pixel thick, perfectly connected, well-defined, but also has the desired property of handling boundary noise. The new technique will be thoroughly described in section 2. Section 3 provides a brief description of related work. Three performance measurements are proposed as objective criteria in section 4 for the evaluation of thinning algorithms. Detailed result analysis and performance evaluation of the proposed algorithm against related work based on these criteria are presented in section 5.

2.0 Problems existing in current algorithms and some solutions

In most existing thinning algorithms, when a pixel is decided as a boundary pixel, it is either deleted directly from the image or flagged and not deleted until the entire image has been scanned. There are deficiencies associated with each of these strategies; namely:

Deficiency of first strategy: lapse from medial axis

In the former case, the deletion of each boundary pixel will change the object in the image and hence affect the final skeleton. This kind of algorithms does not thin the object symmetrically. If the image is scanned in a row-wise manner from the left to the right, the thinned object lines are then located towards the south-eastern borders of the image, because the north-western border points are removed first.

To overcome this problem, some thinning algorithm use several passes or sub-cycles in one thinning iteration [Pavlidis, 1982][Feigin & Ben-Yosef, 1984]. Each pass is an operation to remove boundary pixels from a given direction. However, both the time complexity and memory requirement will be increased.

Deficiency of second strategy: excessive deletion

In the latter case, as the pixels are only flagged, the state of the bitmap at the end of the last iteration will be used in deciding which pixel to delete. However, if this flag map is not used to decide on whether to delete the current pixel then the information generated from processing the previous pixels in the current iteration will be lost. In certain situation, the final skeleton may be badly distorted; for example, a two-pixel-wide line will be completely deleted.

Some efforts have been made to handle this shortcoming, such as using restoring templates [Chin et al., 1987]. These methods to some degree are able to satisfy the requirements, though not completely, at the expense of higher computation time.

In the proposed algorithm, the image and the flag map are used together to decide on which pixel to delete. In this way, one can overcome the problems mentioned earlier as well as to increase the stability of the final skeleton with respect to the boundary noise. When examining a pixel and its neighbours to decide if it is to be flagged as a boundary pixel, both the flag map and the results from the last iteration are used in the decision-making process.

2.1 Semantics of symbols

In the subsequent sections the following symbols will be used to describe the various thinning algorithms. The symbols P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7 and P_8, shown in Figure 1(a), represent the pixel P[i][j] in the bitmap and its neighbours p[i-1][j], p[i-1][j+1], p[i][j+1], p[i+1][j+1], p[i+1][j], p[i+1][j-1], p[i][j-1] and p[i-1][j-1] respectively. The flag map is used for flagging those pixels that will eventually be deleted. The size of the flag map is the same as that of the image. That is to say, for an $n \times m$ image the size of the flag map will also be $n \times m$. In this algorithm, the symbols Q_0, Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, and Q_8, as shown in Figure 1(b), correspond to P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, and P_8 respectively. Initially, all pixels in the flag map are set to 1. The value will be changed to 0 as soon as that pixel is flagged.

P_1	P_2	P_3
P_8	P_0	P_4
P_7	P_6	P_5

(a)

Q_1	Q_2	Q_3
Q_8	Q_0	Q_4
Q_7	Q_6	Q_5

(b)

Figure 1. (a) Local pixel notation of bitmap used in boundary pixel check.

(b) Local pixel notation of flag map used in boundary pixel check.

2.2 Definitions of functions in the proposed thinning algorithm

Three functions are used in the proposed algorithm. They are the previous neighbourhood function *PN*, the current neighbourhood function *CN* and the "$0 \rightarrow 1$" transition function *Trans*. The previous neighbourhood function *PN* is defined as follows:

$$PN(P_0) = \sum_{i=1}^{8} P_i \qquad \text{(e.1)}$$

It counts the number of previous neighbours of pixel P_0, that is the number of its neighbours in the bitmap from the last iteration. Previous neighbourhood function *PN* is used to detect whether a foreground pixel P_0, whose pixel value in the bitmap is one, is a boundary point. If $PN(P_0)$ is equal to 8, then pixel P_0 is not a boundary point since there is no background pixel, whose pixel value is zero, in its neighbourhood. The current neighbourhood function, *CN*, is defined as follows,

$$CN(P_0) = \sum_{i=1}^{8}(P_i \times Q_i) \tag{e.2}$$

It counts the number of current neighbours of pixel P_0. If one foreground pixel in the neighbourhood of P_0 has been flagged then it can no longer be considered to exist. The conditions for matching the current neighbourhood function *CN* will be explained later. The operator "$\times$" in the function *CN* is the logical "and" and the operator "+" is the logical "or".

The "$0 \rightarrow 1$" transition function, *Trans*, is defined as follows,

$$Trans(P_o) = \sum_{i=1}^{8} count(P_i) \tag{e.3}$$

where

$$count(P_i) = \begin{cases} 1, & if \quad ((P_i \times Q_i = 0) \,\&\&\, (P_{i+1} \times Q_{i+1}) = 1) \\ 0, & otherwise \end{cases}$$

and

$$P_9 = P_1, \quad Q_9 = Q_1$$

It gives the number of "$0 \rightarrow 1$" transitions when traversing across the 8-neighbours $P_1, P_2, \ldots, P_8$. The function *Trans* is used to measure the connectivity within the immediate neighbourhood of the pixel. If the condition ($Trans(P_0)=1$) and ($CN(P_0)>1$) are true, then there exists only one connected component within the perimeter of the 3×3 sub-image. In this case, as can be seen in Figure 2, it is safe to remove the central pixel that has the value 1, since its removal will not affect the connectivity of the rest of the pixels in the local window. If the condition ($Trans(P_0)=\min(CN(P_0), 8 - CN(P_0))$) is true, that means the central pixel P is a break point. The removal of the break point will damage the connectivity of the image and hence it must be preserved, see Figure 3. However, there are four special cases which satisfy the second condition but should not be included. They are represented by the templates depicted in Figure 4(a) to Figure 4(h) in which X is displaced by 0. In these four cases, the central point P could be deleted in order to keep 8-connectivity.

0	0	0
0	1	1
1	1	1

Figure 2. Trans(P_0) = 1.

1	1	1
0	1	0
1	0	1

Figure 3. Trans(P_0) = 3 = min(5, 8 - 5).

In equation (e.3), the flag map Q_i is used in the calculation of the value of $Trans(P_0)$. Here the bitmap and the flag map are used to count the "$0 \rightarrow 1$" transitions. The statement "P_i and Q_i is zero" in equation (e.3) means that a foreground pixel is to be considered as background as soon as it has been flagged. In the second statement, a foreground pixel is considered as part of the foreground only if it is not flagged.

2.3 The proposed thinning algorithm

The proposed thinning algorithm is described as follows:

Step 1: for each pixel P in the bitmap do the following:

i) count PN(P), CN(P) and $Trans$(P);

ii) if P is a black boundary pixel that satisfies condition 1 and (condition 2 or condition 3), flag it. The three conditions are given as follows:

condition 1: (CN(P)>1) && (CN(P)<6);

condition 2: ($Trans$(P)==1); and

condition 3: P and its neighbours match one of the smoothing templates in Figure 4(a)-4(h);

Step 2: delete the flagged pixels;

Step 3: repeat step 1 and step 2 until no pixel can be deleted.

x	1	0
0	1	1
0	0	0

(a)

0	1	0
0	1	1
0	0	x

(b)

0	0	x
0	1	1
0	1	0

(c)

0	0	0
0	1	1
x	1	0

(d)

0	0	0
1	1	0
0	1	x

(e)

x	0	0
1	1	0
0	1	0

(f)

0	1	0
1	1	0
x	0	0

(g)

0	1	x
1	1	0
0	0	0

(h)

Figure 4. Smoothing templates used in boundary pixel check.

2.3.1 Explanation of the conditions used in the algorithm.

The following is a detailed discussion of the three conditions used in the algorithm, namely :

Condition 1: *(CN(P)>1) and (CN(P)<6).*

The first component *(CN(P)>1)* in condition 1 is used to prevent the deletion of end points. If the pixel, P, has only one foreground neighbour in the current bitmap, then it must be an end point. The second component *(CN(*P*)<6*) is used to prevent excessive deletion and to enhance the skeleton's shape. The deletion of pixels, which have more than five neighbours in the current bitmap, will generate unwanted loops and branches, and hence produces some distortion and instability in the final skeleton. Many other algorithms which delete such pixels will also generate more noise-sensitive results.

Condition 2: *(Trans(P)==1).*

This predicate checks if the pixels within the perimeter of the 3×3 neighbourhood form one connected component. It ensures that break point will not be deleted. Otherwise, the object in the bitmap will become disconnected.

Condition 3: *matching of smoothing templates 5(a) ... 5(h).*

It deletes those pixels which matches the smoothing templates. The “X” in the templates can be either 1 or 0.

3.0 Related works

Many thinning algorithms have been proposed to date. Their applications often produce different skeleton shapes. The following five algorithms in this section are common in literature. The are briefly outlined and discussed here. In the next section, the results of these algorithms will be compared against that of the proposed one. The first algorithm used is a simple one pass algorithm without using connected templates and flag map [Tamura, 1978]. Pixels are raster scanned throughout the image and deleted against two constraints: connective constraint and end-point constraint. The basic disadvantage of this algorithm is that it does not thin the image object symmetrically. The second is actually a sequential implementation of Zhang and Suen's algorithm, which may be considered as the most widely used algorithm for thinning images. Compared with the first one, it is a more elaborate thinning algorithm that requires two successive iterative passes, six templates and a flag map. The third is Jang and Chin's algorithm [Jang & chin, 1990]. Four sub-cycles and eight templates are employed to remove border points from all four directions as well as unnecessary points at junctions. Jang and Chin's algorithm converges to a one-pixel wide and properly connected skeleton in a finite number of iterations. The fourth thinning algorithm is a one-pass two-operation process proposed by Arcelli and Baja [Arcelli & Baja, 1989]. This process is based on the notion of multiple pixel as well as on the use of the 4-distance transform rather than on the original object. It has been shown to be profitable for both reducing the whole computation time and getting a labeled skeleton. The last one is O'Gorman's K*K Thinning [O'Gorman, 1990]. O'Gorman generalized the commonly used 3*3 thinning into K*K thinning in this paper. That is, instead of examining windows of 3*3 pixels and erasing the center pixel, K*K sized windows are examined throughout the image and a center core of (K-2)*(K-2) pixels are erased if the criteria are met.

The five algorithms mentioned above as well as the one proposed here have been implemented and optimized in C. The test images are 512×128 binary patterns, which have been pre-processed by thresholding the grey level. Samples of these are given in Figure 7.

4.0 Performance measurements

The performance evaluations of the algorithms generally includes execution time (or time complexity) and memory requirement. These two criteria are easy to evaluate but do not answer the question on "Which thinning algorithm is better?". To do this, it is necessary also to evaluate the thinned outputs.

There exists a number of comparative studies examining various thinning algorithms [Davies & Plummer, 1981][Hilditch, 1983][Tamura, 1978]. Although no precise standard has yet been established. It has generally been accepted that a good skeleton produced by a thinning algorithm must possess the necessary properties covering topology, shape, connectivity, thinness, and sensitivity to boundary noise.

The assessment of thinning algorithms can be rather subjective. In order to draw some comprehensive conclusions over the performance of the algorithm, some objective criteria are proposed in this paper. In this section three performance measurements are proposed. They are the thinness measurement, *TM*, the connectivity measurement, *CM*, and the sensitivity measurement, *SM*.

4.1 The thinness measurement TM

A thinned skeleton must be thinned as named. But to what extent can a object in an image be said to be completely thinned or thinned to some degree? The thinness measurement, *TM,* measures the degree to which an object in an image is thinned. This is given in equation (e.4).

$$
\begin{aligned}
&TM = 1 - TM1 \div TM2; \\
&TM1 = \sum_{i=0}^{n} \sum_{i=0}^{m} triangle\_count(P[i][j]); \qquad \text{(e.4)} \\
&TM2 = 4 \times [\max(m,n) - 1]^2;
\end{aligned}
$$

where,

triangle_count(P[i][j])= P[i][j]*P[i][j-1]*P[i-1][j-1] + P[i][j]*P[i-1][j-1]*P[i-1][j] + P[i][j]*P[i-1][j]*P[i-1][j+1] + P[i][j]*P[i-1][j+1]*P[i][j+1]

the operators: "*" and "+" are arithmetic operations, and

n, m are the dimensions of the image.

If P[i][j] is a black pixel, then the value of *triangle_count*(P[i][j]) is equal to the number of triangles whose three vertices are all black pixels as shown in Figure 5. It is easy to prove that, if anywhere in the image that is not one-pixel wide (not thinned), then there exists a triangle composed of three black pixels, see detailed illustrations in Figure 6. Such triangles are referred to as *black triangles*. The total number of black triangles can be used to measure the thinness of the final skeleton. This is given by *TM1* in equation (e.4). *TM1* is normalized with respect to the parameter *TM2*. *TM2* represents the largest number of black triangles that an image could have. Hence, the normalized value of the thinness measurement *TM* lies between 0 and 1. When it is 1, the image is completely thinned and when its value is 0, the image is not thinned at all. The larger the normalized value of TM value the more likely the object in the image is thinned. From the illustrations in Figure 6, one can see that when an image is a black square, it has the largest number of black pixels, and hence its thinness measurement *TM* will be 0.

4.2 The connectivity measurement CM

The second measurement to be defined here is the connectivity measurement, *CM*. This is given in equation (e.5).

$$CM = \sum_{i=0}^{n} \sum_{i=0}^{m} S(P[i][j]) \qquad \text{(e.5)}$$

where

$$S(P[i][j]) = \begin{cases} 1, & if \quad CN(P[i][j]) < 2 \\ 0, & otherwise \end{cases}$$

It is used to measure the connectivity of the output skeleton. An object in an image is said to be disconnected, if it has broken pieces where they should not. Furthermore, wherever there are disconnected lines, there are also end points and discrete points. The number of end points and discrete points in an image is a constant value for a particular image. Hence, when an image is disconnected, it will have more end points and discrete points than it should. From this observation, one could make use of the number of end points and discrete points in the image as a connectivity measurement to measure connectivity of the object in an image. This can be used as an effective criteria for comparing the results of thinning algorithms.

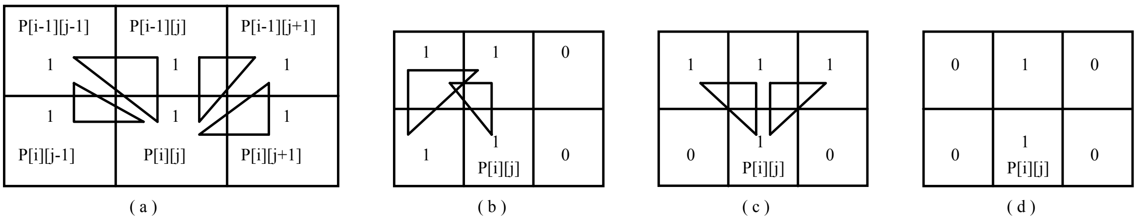


Figure 5. (a) Vertices of black triangles are all black pixels (having value one);

(b) count(P [i] [j]) = 2; (c) count(P [i] [j]) = 2; (d) count(P [i] [j]) = 0.

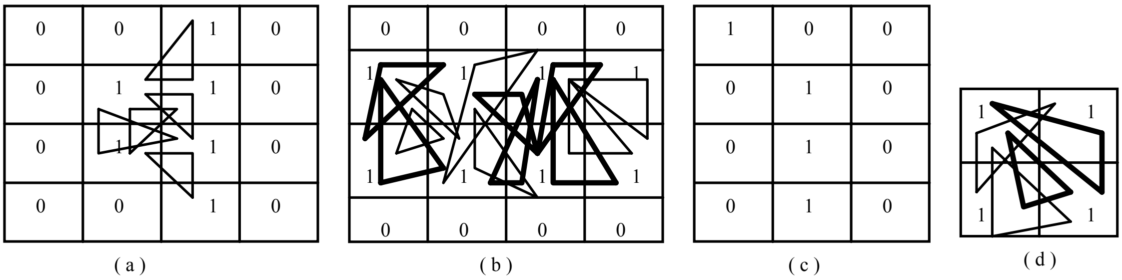


Figure 6. *(a) Black triangles exist wherever the image is not thinned.*

(b) This is a two-pixel wide bar. TM1 = 12, $TM2 = 4\times[\max(4,4)-1]^2$ *= 36, TM = 0.666667.*

The bar can be considered slightly thinned, though not completely.

(c) This is a curved line with TM1=0. Hence, TM=1. The line is completely thinned.

(d) This is a square consisting of 4 black pixels. TM1=4 and TM2=4, hence, TM = 0.

The square is not thinned at all.

It is obvious that a thinned skeleton would have more end points and discrete points than the original image. Thinning algorithms can neither rejoin the broken lines nor eliminate the discrete points. They can only increase end points and discrete points by erroneous deletions. The closer the number of these points in the final skeleton to that in the original image, the higher the degree of connectivity in the skeleton with respect to the original image.

4.3 Sensitivity Measurement

The third measurement is the sensitivity measurement, *SM*. This is given in equation (e.6).

$$SM = \sum_{i=0}^{n} \sum_{j=0}^{m} S(P[i][j]) \tag{e.6}$$

where

$$S(P[i][j]) = \begin{cases} 1, & if \quad Trans(P[i][j]) > 2 \\ 0, & otherwise \end{cases}$$

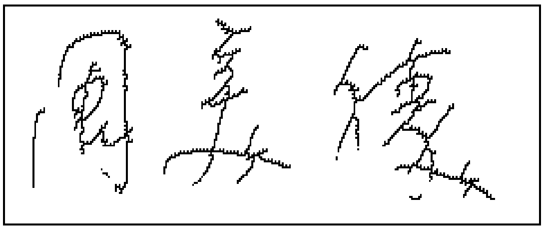
A good thinning algorithm should be immune to slight noise in the image. The noise mentioned here are mainly the perturbations in the outline of an object, not unwanted extra parts in the image. Those unwanted extra parts such as an extra black dot, are called avoidable noise. They can be handled by other image preprocessing techniques such as thresholding, cutting, etc. The main concern here is only the small perturbations along the outline of the object which usually cause deformation, offshoots, and tabs in the final skeletons. Many algorithms have attempted to improve their immunity to the noise [Chin et al., 1987][Jang & Chin, 1990]. However the evaluations of their efforts is still unclear. The sensitivity measurement is a simple but effective criterion for addressing this issue.

It is obvious that a cross point exists wherever there is an offshoot or a tab. For a given image, the total number of its cross points is fixed. The excess cross points are caused by the vulnerability to noise and the inadequacy of the thinning algorithm. Hence, the total number of cross points in an image can be used as a kind of sensitivity measurement. The fewer the cross points, the higher the immunity of the algorithm to noise.

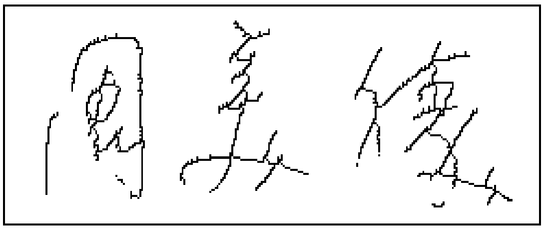
5.0 Results analysis and comparison

In this section, the performance of the various thinning algorithms are analyzed according to the criteria proposed in the previous section. The real running times, and the measurements, *TM*, *CM* and *SM* of the different thinning algorithms on Chinese signatures, are listed in Table 1.

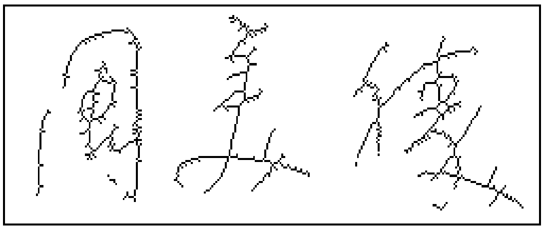
(a) The original signature



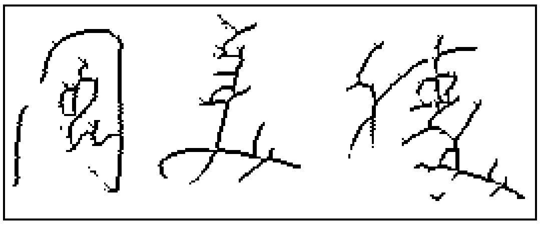
(b) The final skeleton of Tamura’s algorithm



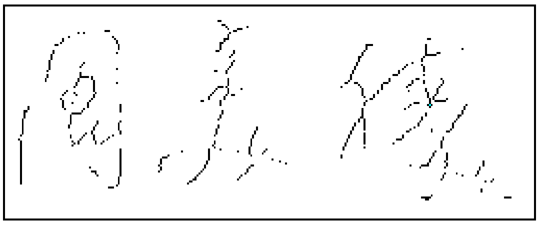
(c) The final skeleton of Zhang and Suen's algorithm



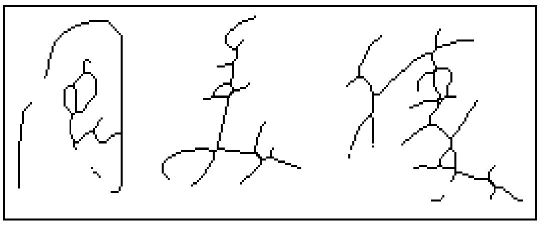
(d) The final skeleton of Jang and Chin’s algorithm



(e) The final skeleton of Arcelli and Baja's algorithm



(f) The final skeleton of O'Gorman’s algorithm.



(g) The final skeleton of the proposed algorithm.

Figure 7. The final skeletons of Chinese signature using different thinning algorithms.

A Chinese signature after binarisation is used as the test image. This is shown in Figure 7(a). It is still quite noisy along the boundary. Figure 7(b) to 7(e) are the final skeletons of the three algorithms introduced in section 3. Figure 7(e) shows the result of the proposed algorithm. From the skeletons in Figure 7, it is easily observed that the final skeletons derived from both Tamura's algorithm and Jang & Chin's algorithm are the most noisy, and the results obtained from the proposed algorithm is smooth, clear and connected when compared with the other three. Of course, this observation is totally subjective. The objective measurements based on the criteria suggested in section 4 are listed in the Table 1.

Performance Comparison	*TM* (thinness measurement)	*CM* (connectivity measurement)	*SM* (sensitivity measurement)	real running time on SUN SPARC 10
the original image after thresholding	0.987293	13	30	
Tumura's algorithm	0.998437	37	80	1.103995 secs
Zhang and Suen's algorithm	0.999010	43	56	2.018892 secs
Jang and Chin's algorithm	0.999925	145	89	4.065187 secs
Arcelli and Baja's algorithm	0.996533	44	21	1.289685 secs
O'Gorman's algorithm	0.999975	206	6	1.475501 secs
our proposed algorithm	0.999896	27	14	1.371793 secs

Table 1.Measurement comparison of final skeletons in Figure 7.

The first column of Table 1 gives the comparison of the thinness measurement, *TM*, for the various algorithms. The value of *TM* of the original image is 0.987293, not 0. This is because, as was mentioned earlier, only the *TM* value of a black square is zero. Although the original image is not thinned at all, its strokes are still quite slim. The thinness of Jang and Chin's result is the best, while the skeleton produced by the proposed algorithm is the next best. Next, the second column illustrates the comparison of the 4 thinning algorithms on the basis of the connectivity measurement. As anticipated, the connectivity of the proposed algorithm is the best when compared with the others, especially when compared with

O'Gorman's algorithm, which has a connectivity measurement of 206. Thirdly, the third column shows the comparison of the 4 thinning algorithms on the basis of the sensitivity measurement. From this column, it is clear that the proposed algorithm produces an amazing result which has 14 cross points, even less than that of the original image. The total number of cross points in the final skeleton obtained by the proposed algorithm is 17.5% of Tamura's algorithm, 25% of Zhang and Suen's algorithm, 15.7% of Jang and Chin's algorithm, 66.7% of Arcelli and Baja's algorithm, and 46.7% of the original image. But it is still not the best, the final skeleton obtained by O'Gorman's algorithm has only 6 cross points at the expense of connectivity performance. This coincides with the visual observations. Lastly, the fourth column tabulates the comparison of actual execution time of the 4 algorithms. Tamura's algorithm is the fastest followed by Arcelli and Baja's algorithm, while the proposed algorithm comes in third.

The above analysis shows that the proposed algorithm has managed to produce several desired improvements. Some of its attractive features are low computation time, high reliability and stability, particularly with respect to noise. Its advantages are obvious when considering thinness, connectivity, noise sensitivity and time complexity. But most importantly, it is the high immunity to boundary noise that accounts for the greatest visual impact.

6.0 Conclusion

This paper describes a novel single pass thinning algorithm, its properties and advantages. Detailed discussion of each step, conditions and templates used in the algorithm is given. Three objective performance criteria for thinning algorithms are proposed in this paper to effectively compare the proposed algorithm against a set of well established techniques. The results generated by this algorithm are generally better than those produced by the other algorithms. The comparison shows that the algorithm is fast, reliable, immune to noise, and hence will be an effective technique in image processing.

7.0 Acknowledgment

The authors wish to acknowledge the financial support from the Nanyang Technological University, School of Applied Science, for the work undertaken within the research project AF: An Intelligent Toolkit for Anti-Forgery.

References

[Arcelli, 1981] C. Arcelli, "Pattern thinning by contour tracing", Computer Graphics and Image Processing 17, 2, pp. 130-144, Oct., 1981.

[Arcelli & Baja, 1989] C. Arcelli and G. S. Baja, "A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 11, No. 4, April 1989.

[Blum, 1964] H. Blum, "A transformation for extracting new descriptors of shape", in Proc. Symp. Models for the Perception of Speech and Visual Form, Cambridge, MA: MIT Press, 1964.

[Chin et al., 1987] R. T. Chin, H. K. Wan, D. L. Stover and R. D. Iverson, "A one-pass thinning algorithm and its parallel implementation", Comp. Vision, Graphics, and Image Processing, 40, pp. 30-40, 1987.

[Davies & Plummer, 1981] E. R. Davies and A. P. N. Plummer, "Thinning algorithms: A critique and a new methodology", Pattern Recognition, 14, pp. 53-63, 1981.

[Dill & Levine, 1987] A. R. Dill and M. D. Levine, "Multiple resolution skeletons", IEEE Trans. Pattern Analysis and Machine Intelligence PAMI-9, 4, pp. 495-504, July, 1987.

[Feigin & Ben-Yosef, 1984] G. Feigin and N. Ben-Yosef, "Line thinning algorithm", Proc. SPIE, vol. 397, pp. 108-112, 1984.

[Fischler & Bolles, 1983] M. A. Fischler and R. C. Bolles, "Perceptual organization and the curve partitioning problem", in Proc. 8th IJCAI, Karlsruhe, pp. 1014-1018, 1983.

[Hilditch, 1983] C. J. Hilditch, "Comparison of thinning algorithms on a parallel processor", Image Vision Comput. 1, pp. 115-132, 1983.

[Jang & Chin, 1990] B. K. Jang and R. T. Chin, "Analysis of thinning algorithms using mathematical morphology", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 12, No. 6, pp. 541-551, June, 1990.

[O'Gorman, 1990] L. O'Gorman, "K*K thinning", Comput. Vision Graphics and Image Processing, vol. 51, pp. 195-215, 1990.

[Levine, 1985] M. D. Levine, "Vision in man and machine", New York: McGraw-Hill, 1985.

[Pavlidis, 1980] T. Pavlidis, "A thinning algorithm for discrete binary images", Computer Graphics and Image Processing 13, pp. 142-157, 1980.

[Pavlidis, 1982] T. Pavlidis, "Algorithms for graphics and image processing", Computer Graphics and Image Processing, vol. 20, pp. 133-157, 1982.

[Persoom & Fu, 1977] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors", IEEE Trans. Syst., Man. Cybern., vol. SMC-7, no. 3, pp. 170-179, 1977.

[Pavlidis, 1982] T. Pavlidis, "Algorithms for graphics and image processing", Computer Science Press, Rockville, Md, 1982.

[Rosenfeld & Pfaltz, 1966] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing", J. ACM 13, 4, pp. 471-494, Oct., 1966.

[Stefanelli & Rosenfeld, 1971] R. Stefanelli and A. Rosenfeld, "Some parallel thinning algorithms for digital pictures", J. ACM 18, 2, pp. 255-264, April, 1971.

[Tamura, 1978] H. Tamura, "A comparison of line thinning algorithms from digital geometry viewpoint", Proc. Fourth Intern. Joint Conf. on Pattern Recognition, Kyoto, pp. 715-719, November 1978.

[Xu & Wang, 1987] W. Xu and C. Wang,, "CGT : A fast thinning algorithm implemented on a sequential computer", IEEE Trans. Systems, Man and Cybernetics SMC-17, (5), pp 847-851, 1987.

[Zhang & Suen, 1984] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns", Communications of ACM 27, (3), pp. 236-239, 1984.

List of Figure Captions

Figure 1. (a) Local pixel notation of bitmap used in boundary pixel check.

(b) Local pixel notation of flag map used in boundary pixel check.

Figure 2. Trans(P_0) = 1

Figure 3. Trans(P_0) = 3 = min(5, 8 - 5).

Figure 4. Smoothing templates used in boundary pixel check.

Figure 5. (a) Vertices of black triangles are all black pixels (having value one).

(b) count(P [i] [j]) = 2.

(c) count(P [i] [j]) = 2.

(d) count(P [i] [j]) = 0.

Figure 6. (a) Black triangles exist wherever the image is not thinned;

(b) This is a two-pixel wide bar. *TM1* = 12, $TM2 = 4 \times [\max(2,4) - 1]^2 = 36$, *TM* = 0.666667.

The bar can be considered slightly thinned, though not completely.

(c) This is a curved line with *TM1*=0. Hence, *TM=1*. The line is completely thinned.

(d) This is a square consisting of 4 black pixels. *TM1*=4 and TM2=4, hence, *TM* = 0.

The square is not thinned at all.

Figure 7. The final skeletons of Chinese signature using different thinning algorithms.

(a) The original signature

(b) The final skeleton of Tamura's algorithm

(c) The final skeleton of Zhang and Suen's algorithm.

(d) The final skeleton of Jang and Chin's algorithm.

(e) The final skeleton of Arcelli and Baja's algorithm.

(f) The final skeleton of O'Gorman's algorithm.

(g) The final skeleton of the proposed algorithm.

Table 1.Measurement comparison of final skeletons in Figure 7.