An Improved and Efficient Rotation Invariant Thinning Algorithm for Binary Document Images

V.N. Manjunath Aradhya,[1] G. Hemantha Kumar,[1] P. Shivakumara[2]

[1]*Department of Studies in Computer Science, University of Mysore, Mysore, INDIA;* [2]*Department of Computer Science, School of Computing, National University of Singapore (NUS), 3 Science Drive 2, Singapore 117543*

ABSTRACT

Skeletonization has been a part of image processing for a wide variety of applications. Thinning is the process of reducing thickness of each line of pattern to just a single pixel width. Thinning is usually used as the first step in application such as optical character recognition to improve the recognition rate. In this paper, we present an improved and efficient rotation invariant algorithm for binary document images. The proposed algorithm preserves topology of the characters and it is rotation invariant. The algorithm is improved version of the 2-steps method by Gonzales and Woods. This is a generalized algorithm that can be used to thin symbols, digits, characters, letters, irrespective of the script in which they are written. The results show that the system is very efficient in preserving the topology of symbols and letters written in any language.

KEYWORDS

optical character recognition, thinning, skeletonization, rotation invariant, document image analysis

---

Correspondence to: V.N. Manjunath Aradhya, Department of Studies in Computer Science, University of Mysore, Mysore, INDIA; email: V.N.M.A. - mukesh_mysore@rediffmail.com; G.H.K. - hemanth_kumar@lycos.com; P.S. - hudempsk@yahoo.com

1. INTRODUCTION

Binary image thinning plays an important role in image processing such as handwriting recognition (Abuhaiba et al , 1996), text recognition (Fitz & Green, 1995), fingerprint classification (Fitz & Green, 1995; Luk et al , 1991), and data compression. The results of binary image thinning are skeletons that represent the structural shape of original images by using less data. A good thinning method should produce skeletons including the shape information of original object with the accuracy so that these skeletons are suit for application (Xiaodong et al , 2004). Many thinning algorithms have been introduced for decades that have advantages and disadvantages, respectively. Some algorithms can obtain good quality skeletons but run slowly (Lam et al , 1992). Some algorithms run fast, but resulting skeletons are not good. Previously developed thinning algorithms do not address the problem of rotation invariance property.

A comprehensive survey of thinning algorithms is described in (Lam et al , 1992). Templates of (3 x 3) and (3 x 4), and (4 x 3) windows are usually used to perform thinning and trimming of extraneous pixels (Kim et al, 1992; Ahmed & Ward, 1998; Han et al , 1997; Arcelli et al, 1992). The algorithm described in Ahmed and Ward has the advantage that its rules are applied in one sequential pass only. In Arcelli et al (1992), an iterative parallel thinning algorithm that uses four passes per iteration is proposed. A modified algorithm with sub-iterations is proposed in (Arcelli et al , 1993). The latter method, however, may produce extraneous branches. A two-step algorithm that improves thinning results of Chinese characters is described in (He & Yan, 2000). In Altuwaijri and Bayoumi (1995), an artificial neural network, based on adaptive resonance theory, is proposed for thinning Arabic characters. The ZS algorithm described in Zhang and Suen (1984) has the advantage of speed and preserves the topology of the image in the majority of cases. However, all the pixels in the case of 2 x 2 image pixels will be deleted. In addition, a 45^0, 135^0 long or short diagonal line of 2 pixels width is reduced to a dot of 1 or 2 pixels. A solution for this case is suggested in Raju and Xu (1991). The algorithm proposed in Zhang and Wang (1996) produces thinner results (fewer pixels) than the above ZS algorithm, maintains high speed, and also generates one pixel wide skeletons. The HSCP algorithm described in Holt et al (1987) uses the first two conditions as in the

ZS method to determine which pixel can be deleted. The HSCP algorithm is compared with the ZS algorithm (Hall, 1989), and suggests a solution for solving the diagonal line problems in the ZS and HSCP algorithms.

Many thinning algorithms, their performance, and evaluation have been studied in Lam and Suen (1995). None of the previous methods addresses rotation invariant thinning. Many are specific to digits, characters, or letters, written in English, Chinese, Arabic, or any other scripts. However, to solve aforementioned problems, the rotation invariant rule-based thinning algorithm for character recognition is proposed by Ahmed and Ward (2002). This generalized algorithm is used to thin the symbols irrespective of their scripts. The advantage of this method is it is invariant to rotation. The algorithm fails on two-pixel wide lines. To overcome this drawback, an improved rotation-invariant thinning algorithm is proposed by Peter. I. Rockett (2005). The method proposed by Hemantha et al (2004) presents a thinning algorithm with 14 rules and reduces the time and computations compared with the method described in Ahmed and Ward.

From the above literature, we realize that there is a scope for developing a fast and efficient thinning algorithm that preserves shape even after rotation. Hence, in this paper, we present a new improved and efficient thinning algorithm that can be used to thin symbols, digits, characters, or letters, irrespective of the script in which they are written. The obtained thinned symbols will be the central lines of the symbols and, thus, the system is invariant to rotations in the original pattern. The advantage of the proposed systems is that the thinned pattern is one pixel wide and preserves the topology and does not produce any discontinuity and takes less time compared with the methods proposed in Ahmed and Ward (2002) and Hemantha et al (2004). The rest of the paper is organized as follows: In section 2, we describe our proposed thinning methodology. Experimentation and comparative study are given in section 3. Finally, a conclusion is given in section 4.

2. PROPOSED METHODOLGY

Our algorithm is iterative. At each iteration, our algorithm deletes every point that lies on the outer boundaries of the symbol, as long as the width of the symbol is more than one pixel wide. Region points are assumed as 1 and background points are assumed to have 0 value. The proposed method consists of successive passes of four basic steps applied to the contour points of the given region, where contour point is any pixel with value 1 and having at least one 8-neighbor valued 0. In two steps method (Gonzales and Woods, 2002), 8-neighborhood of a pixel is used to thin a character image. Instead of using 8-neighbor, the proposed method uses only four neighbors with four basic steps to thin a character image. 8-neighborhood notations and 4-neighborhood notation are shown in figures 1 and 2, respectively.

P_1	P_2	P_3
P_8	P_0	P_4
P_7	P_6	P_5

Fig. 1: 8-neighborhood of the pixel

	P_2	
P_8	P_0	P_4
	P_6	

Fig. 2: 4-neighborhood of the pixel

First, the number of nonzero neighbors of P_0 is calculated as N (P_0) = P_2 + P_4 + P_6 + P_8. Transition say, T (P_0) is calculated as the number of 0-1 transition in the order sequence P_2, P_4, P_6, and P_8 respectively. Finding the number of nonzero neighbors and 0-1 transition using 4-neighborhood, the proposed method uses four basic steps to complete one iteration. In each iteration, the proposed method deletes the pixels in all 4-directions i.e. pixels P_2, P_4, P_6, and P_8 if satisfied. The four steps of the proposed method are shown in the following algorithm.

Algorithm for thinning
Input: Any Symbols
Output: Thinned Image
Method:
Step 1:

(a) If $2 <= N(P_0) <= 4$
(b) If $T(P_0) = 1$
(c) If $P_2 = 0$
(d) If $P_1 = 0$
If the above condition is satisfied, make $P_0 = 0$.

Step 2:

(a)If $2 <= N(P_0) <= 4$
(b) If $T(P_0) = 1$
(c) If $P_4 = 0$
(d) If $P_3 = 0$
If the above condition is satisfied, make $P_0 = 0$.

Step 3:

(a) If $2 <= N(P_0) <= 4$
(b) If $T(P_0) = 1$
(c) If $P_6 = 0$
(d) If $P_5 = 0$
If the above condition is satisfied, make $P_0 = 0$.

Step 4:

(a) If $2 <= N(P_0) <= 4$
(b) If $T(P_0) = 1$
(c) If $P_8 = 0$
(d) If $P_7 = 0$
If the above condition is satisfied, make $P_0 = 0$.

Step 5: Repeat the above procedure for all the region points.

Step 6: Stop.

Method ends

Step 1 is applied to every border pixel in the binary region under consideration. If one or more of conditions (a)-(d) are violated, then the value of the point in question is not changed. If all conditions are satisfied the point is flagged for deletion. However, the point is not deleted until all border points have been processed. This delay prevents changing the structure of the data during execution of the algorithm. After step 1 has been applied to all border points, those that are flagged are deleted (changed to 0). Then, step 2, step 3, and step 4 are applied to the resulting data in exactly the same manner as step 1. Finally, at each iteration of the thinning algorithm consists of applying step 1 to flag border points for deletion; deleting the flagged points; applying step 2, 3 and 4 to flag the remaining border points for deletion; and deleting the flagged points. These four basic procedures is applied iteratively until no further points are deleted, at which time the algorithm terminates, yielding the thinned symbols will be the central lines of the symbols and, thus the system is invariant to rotations in the original pattern.

3. EXPERIMENTAL RESULTS AND COMPARATIVE STUDY

In this section, we present some experimental results that were carried out by the proposed method. All experiments were carried out on a PC with P4 1.6GHz CPU and 256 MB RAM memory under "C" Language. We conducted many experiments on different datasets that are synthetically created. Results of thinning for some sample binary images are shown in figure 3, 4, 5 and 6. The proposed thinning system is compared with well known existing systems (Gonzales & Woods, 2002; Ahmed & Ward, 2002; Hemantha et al , 2004). Results of thinning for different methods are shown in figures 3–6. The results obtained from method described by Gonzales and Woods do not preserve topology of the symbols and it is not invariant to rotations.

The method described by Ahmed and Ward preserves shape of the symbols and is invariant to rotations. The method fails on two-pixel wide lines and is computationally expensive because it uses more rules to compute the thinning process. The method by Hemantha et al preserves the shape of the symbols, even after rotations. The advantage of the method proposed by Hemantha et al is that it takes less time compared with the Ahmed and Ward method because the rules are reduced from 20 to 14. The results obtained from the proposed system are very efficient and preserve the topology of the symbols and is invariant to rotations and takes less time compared to the aforementioned methods.

To establish the performance of the proposed system with existing methods, we considered parameters such as Time taken to execute the system and Computations involved during the process. Results obtained using these parameters for binary images of figure 1, 2, 3 and 4 are tabulated in Table 1, 2, 3 and 4 respectively.

From Table 1, 2, 3, and 4 it is clear that the proposed method takes less time and computations compare with the methods by Ahmed and Ward (2002) and Hemantha et al (2004). The proposed method takes quite more time and computations compared with Gonzales and Woods because their method has two basic steps for computing the thinning process, whereas in our proposed system, four basic steps are used to thin a given image. The methods described in Ahmed and Ward and Hemantha et require more time and computations because of more rules that have been adopted to thin a given character image. Tables 5–8 reports the number of pixels deleted at each iteration with respect to figures 3–6.

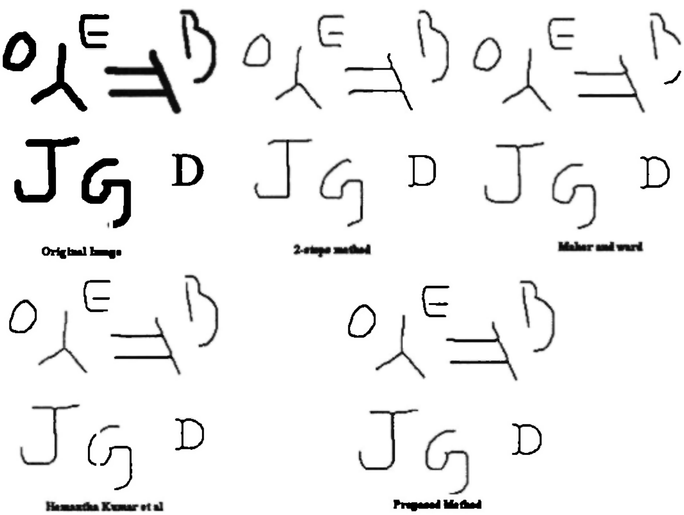

Fig. 3: The thinned results obtained after thinning

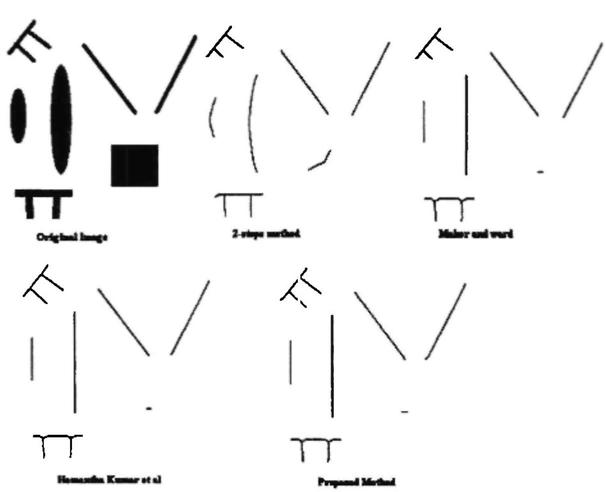

Fig. 4: The thinned results obtained after thinning

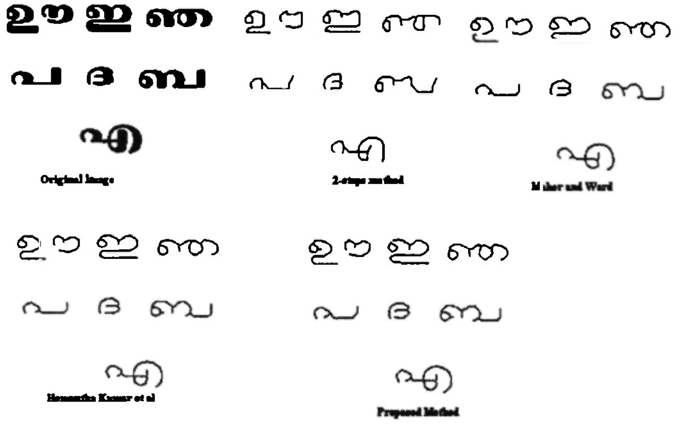

Fig. 5: The thinned results obtained after thinning

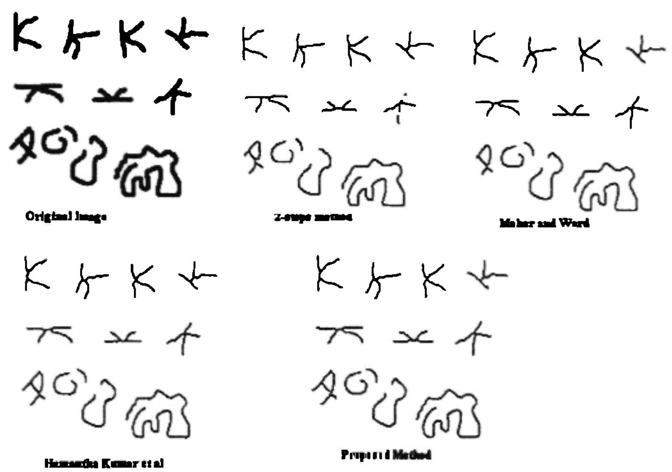

Fig. 6: The thinned results obtained after thinning

From Table 5, 6, 7, and 8, it is noticed that the proposed system deletes fewer pixels at each iteration compared to methods of Gonzales and Woods (2002), Ahmed and Ward (2002), and Hemantha et al (2004). From this, we can say that the proposed system preserves topology of the symbols and letters, irrespective of the scripts in which they are written. One of the advantages of comparing the proposed method with the 2 step method of Gonzales and Woods

TABLE 1

Results of time and computations obtained from figure 3

Method	Time	Computations
Gonzales and Woods, 2002	1.14	28610
Maher and Ward, 2002	5.98	1430515
Hemantha Kumar et al, 2004	4.21	975004
Proposed Method	1.39	49601

TABLE 2

Results of time and computations obtained from figure 4

Method	Time	Computations
Gonzales and Woods, 2002	1.09	5920
Maher and Ward, 2002	4.52	502016
Hemantha Kumar et al, 2004	3.11	191407
Proposed Method	1.20	8101

TABLE 3

Results of time and computations obtained from figure 5

Method	Time	Computations
Gonzales and Woods, 2002	1.09	8519
Maher and Ward, 2002	4.77	425990
Hemantha Kumar et al, 2004	3.311	299978
Proposed Method	1.29	14990

TABLE 4

Results of time and computations obtained from figure 6

Method	Time	Computations
Gonzales and Woods, 2002	1.11	8979
Maher and Ward, 2002	4.69	448986
Hemantha Kumar et al, 2004	3.29	301356
Proposed Method	1.29	14156

TABLE 5

Results of number of pixels deleted obtained from figure 3

No of Iteration	Gonzales and Woods, 2002	Maher and Ward, 2002	Hemantha Kumar et al, 2004	Proposed Method
1	1126	2250	2256	721
2	1121	2186	2191	720
3	1001	1450	1447	722
4	1020	1176	1173	701
5	891	971	972	681
6	791	901	908	666
7	611	840	845	651
8	588	771	763	601
9	458	600	614	410
10	450	529	517	411
11	355	418	409	399
12	359	255	264	381
13	299	213	213	225
14	289	198	197	221
15	256	180	181	210
16	225	161	165	200
17	201	115	149	199
18	220	110	133	190
19	125	102	117	168
20	126	89	101	178
21	111	65	85	120
22	101	61	69	122
23	100	58	53	126
24	99	49	37	129
25	84	32	0	101
26	83	22		102
27	78	11		99
28	76	0		97
29	77			66
30	69			61
31	39			59
32	38			60
33	28			32
34	26			31
35	14			24
36	15			22
37	11			20
38	12			19
39	09			12
40	07			14
41	04			11
42	03			10
43	00			09
44	00			05
45				00
Total number of pixels deleted	10823	10869	10819	10816

TABLE 6

Results of number of pixels deleted obtained from figure 4

No of Iteration	Gonzales and Woods, 2002	Maher and Ward, 2002	Hemantha Kumar et al, 2004	Proposed Method
1	1689	3270	3352	1589
2	1276	2840	2833	1476
3	1559	2244	2266	1259
4	1569	604	629	1169
5	1101	29	28	990
6	800	0	0	700
7	596			596
8	400			398
9	312			311
10	0			301
11				211
12				124
13				14
14				0
Total number of pixels deleted	9001	8987	9108	9011

TABLE 7

Results of number of pixels deleted obtained fram figure 5

No of Iteration	Gonzales and Woods, 2002	Maher and Ward, 2002	Hemantha Kumar et al, 2004	Proposed Method
1	1501	2998	3143	1623
2	1496	1928	1987	1347
3	1001	1483	1469	998
4	927	899	898	847
5	789	329	322	757
6	699	159	6	669
7	597	8	0	589
8	450	0		359
9	200			225
1	169			101
11	0			99
12				54
13				0
Total number of pixels deleted	7847	7804	7825	7821

TABLE 8

Results of number of pixels deleted obtained from figure 6

No of Iteration	Gonzales and Woods, 2002	Maher and Ward, 2002	Hemantha Kumar et al, 2004	Proposed Method
1	1455	3210	3280	989
2	1389	2191	2101	845
3	1112	111	175	823
4	999	11	22	745
5	501	9	0	658
6	4	0		526
7	0			502
8				315
9				102
10				25
11				0
Total number pixels deleted	5573	5532	5578	5574

(2002) is that it emphasizes the proposed method's special property of preserving the topology of the symbols even after they become rotated. Compared with the method of Ahmed and Ward (2002) or Hemantha et al (2004), the proposed method consumes less time and requires fewer computations. The experimental results show that the proposed thinning algorithm is better than either Gonzales and Woods, Ahmed and Ward, or Hemantha et al, in terms of time, computations, and preserving topology.

We also extended our proposed thinning algorithms by adding noise to the boundaries. The proposed system works well for the noise images. Figure 7 shows the results of thinning an image with noise.

4. CONCLUSIONS

Good thinning algorithms should preserve the topology (shape) of symbols. Preserving the shape is accomplished by representing symbols by their central lines. The proposed system thins the symbols to their central

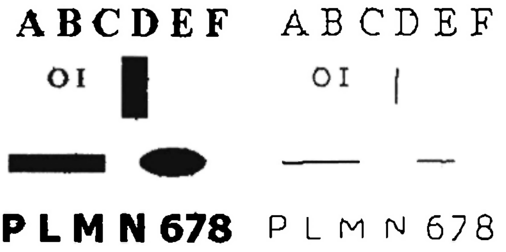

Fig. 7: Results of thinning an image with noise

lines. Our method has the advantage of being rotationally invariant: if the original image is rotated, the resultant thinned image is also rotated by the same angle. The proposed thinning algorithm has greater advantage of being compared with the 2-steps method (Gonzales & Woods, 2002). Compared with all three methods (Gonzales & Woods; Ahmed & Ward, 2002; Hemantha et al, 2004), the proposed method is better with respect to time, computations, and preserving the topology. The experimental results are presented as symbols, characters, and letters written in different languages, and on rotated and noisy symbols. The results show that the developed method is accurate, effective, fast, and can thin any symbol in any language, irrespective of the direction of rotation or flipping. However, the presence of heavy noise remains the subject of further research.

REFERENCES

Abuhaiba, I., Holt, M., Datta, S. 1996. Processing of binary images of handwritten text documents, *Pattern Recognition*, 29, 1161-77.

Ahmed, M and Ward, R. 1999. A novel intelligent system for defining similar symbols *IEEE Pacific Rim Conference on Communications. Computers and Signal Processing (PACRIM)*, 197-200.

Ahmed, M and Ward, R. 2002. A rotation invariant rule-based thinning algorithm for character recognition, *IEEE. Trans. Pattern Analysis and Machine Intelligence*, **24**, 1672-78.

Ahmed, M. and Ward, R. 1998. A fast one pass knowledge-based system for thinning *Electronic Imaging*, 7, 111-16.

Ahmed, M., and Ward, R. 2000. An expert system for general symbol recognition, *Pattern Recognition*, **33**, 1975-88.

Altuwaijri., M and Bayoumi, M. 1995. A new thinning algorithm for arabic characters, using a self-organizing neural network, *Proceedings of the IEEE Int'l Symp. Circuits and System*, **3**, 1824-27.

Amin, A. 2000. Recognition of hand-printed latin characters based on generalized hough transform and decision tree learning techniques, *Pattern Recognition and Artificial Intelligence*, **14**, 369-87.

Arcelli, C., and Sanniti di Baja, G. 1993. Parallel pattern compression by octagonal propagation, *Pattern Recognition and Artificial Intelligence*, 7, 1077-102.

Arcelli, C., Kwok, P.C.K., and Sanniti di Baja, G. 1992. Medical lines by parallel wise erosion of successive contour pairs, *IEEE. Trans. TENCON: Technology Enabling Tomorrow*, **1**, 66-70.

Fitz, A., Green, R. 1995. Fingerprint pre-processing on a hexagonal grid, *Proceedings of Europe Convention on Security and Detection*, Brighton, UK, 257-60.

Gonzales, R.C., and Woods, R.E. 2002. *Digital image processing*, 2nd ed., Pearson Education Asia.

Hall, R.W. 1989. Fast parallel algorithms: parallel speed and connectivity preservation, *Comm. ACM*, **32**, 124-31.

Han, N.H., La, C.W., and Rhee, P.K. 1997. An efficient fully parallel thinning algorithm *IEEE. Trans. Document Analysis and Recognition*, **1**, 137-41.

He, R., and Yan, H. 2000. Stroke extraction as pre-processing step to improve thinning results of Chinese characters, *Pattern Recognition Letters*, **21**, 817-25.

Hemantha Kumar, G., Shivakumara, P., Manjunath Aradhya., V. N., Noushath, S. 2004. Efficient and rotation invariant rule- based thinning algorithm for character recognition, *Proceedings of 12th International Conference on Advanced Computing and Communication*, Ahmedabad, Gujarat, INDIA, 167-72.

Holt, C.M., Stewart, A, Clint, M., and Perrott, R.H. 1987. An improved parallel thinning algorithm, *Comm. ACM*, **30**, 156-60.

Kim, Y., Choi, W and Kim, S. 1992. High-speed thinning processor for character recognition system, *IEEE Trans. Consumer Electronics*, **38**, 762-66.

Lam, L. and Suen, C. 1995. An evaluation of parallel thinning algorithms for character recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **17**, 914-19.

Lam, L., Lee, S. and Suen, C. 1992. Thinning methodologies - a comprehensive survey, *IEEE Trans. Pattern Analysis and machine Intelligence*, **14**, 869-85.

Lin. Y and Chen. Z. 1995. A Chinese-character thinning algorithm based on global features and contour information, *Pattern Recognition*, **28**, 493-512.

Luk, A., Leung, S., Lee, C., Lau, W.1991. A two-level classifier for fingerprint recognition, *Proceedings of IEEE international Symposium on Circuits and System*, Singapore, 2626-28.

Peter I. Rockett. 2005. An improved rotation-invariant thinning algorithm, *Proceedings of IEEE Trans. Pattern Analysis and Machine Intelligence*, **27**, 1671-74.

Raju., G and Xu, Y. 1991. Study of parallel thinning algorithms, *Proceedings of the IEEE Int'l Conf. Systems, Man, and Cybernetics*, **3**, 661-66.

Suzuki, T., Mori, T., and Mori, S. 1994. Structural description of line images by the cross section sequence graphs, *Thinning Methodologies for Pattern Recognition*, World Scientific, Singapore.

Tellache, M., Sid-Ahmed, M., Anaza, B.1993. Thinning algorithms for Arabic OCR, *Commum. Comput. Signal Process, IEEE Pacific Rim*, 248-51.

Xiaodong, Gu., Daoheng, Yu., Liming Zhang. 2004. Image thinning using pulse coupled neural network, *Pattern Recognition Letters*, **25**, 1075-84.

Zhang, Y.T., and Suen, C. 1984. A fast parallel algorithm for thinning digital patterns, *Communicaitons ACM*, **27**, 236-39.

Zhang, Y.Y., and Wang, P.S.P. 1996. A parallel thinning algorithm with two-sub iteration that generates one-pixel wide skeletons, *Proceedings of the IEEE Int'l Conf. Pattern Recognition*, .**4**, 457-61.