ISSN:0975-9646

Harish Kumar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (5) , 2011, 2375-2379

A Comparative Study of Iterative Thinning Algorithms for BMP Images

Harish Kumar and Paramjeet Kaur

Deptt. Of CSA, Chaudhary Devi Lal University, Sirsa, Haryana, India[1],

Abstract **- Thinning is a very important preprocessing step for the analysis and recognition of different types of images. Thinning is the process of minimizing the width of a line, in an image, from many pixels wide to just one pixel (Lam *et al.*, 1992) [3]. Thus Correct and Reliable thinning of character patterns are essential to a variety of applications in the field of document analysis and recognition systems like pattern recognition; finger print recognition etc. Iterative parallel thinning algorithms which generate one-pixel-wide skeletons generally have difficulty in preserving the connectivity of an image. In present paper, four iterative parallel thinning algorithms namely the Fast Parallel Thinning Algorithm (FPTA), Guo & Hall's parallel thinning Algorithm (GHPTA), Robust Parallel Thinning Algorithm for binary images (RPTA), and Preprocessing Thinning Algorithms for Handwritten Character Recognition (PPTA) have been implemented in C Language and evaluated. A comparison among these was made on the basis of following factors: quality of skeleton, convergence to unit width and data reduction rate. This study may help to compare or to select the best algorithm. The comparative results are included to support our findings.**

Keywords: **Iterative Parallel Thinning, GHPTA, PPTA, RPTA, FPTA, Thinning, Pattern.**

I. Introduction

The present paper is organized in various sections. In the current Section-I, we will give the brief introduction about Thinning and usefulness of thinning in various applications. In Section II we will discuss in brief about Iterative parallel Thinning algorithms and the algorithms under study. The next section-III covers Comparative Analysis of results. We will refer the selected algorithms as FPTA, GHPTA, RPTA and PPTA in short in the entire paper. Section IV concludes the paper. In the next section References are included.

Thinning is the process of reducing an object in a digital image to the minimum size necessary for machine recognition of that object. After thinning, analysis on the reduced size image can be performed. Thinning is essentially a "pre-processing" step used in many image analysis techniques. The thinning process reduces the width of pattern to just a single pixel. Thinning when applied to a binary image, produces another binary image as output [1], [2].

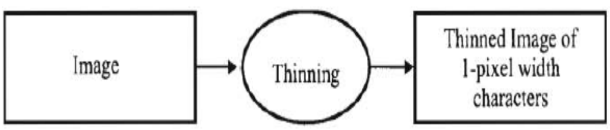

Figure 1: Thinning as a Process [2]

Thinning is usually performed on monochrome, or two-color, images.

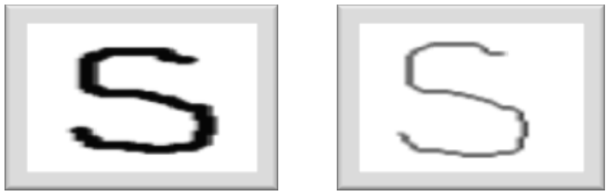

Figure 2: Thinning of Handwritten character 'S'

Thinning has been used in a wide variety of applications including: pattern recognition, medical imaging analysis, bubble-chamber image analysis (a device for viewing microscopic particles), text and handwriting recognition and analysis. A good thinning algorithm is one that posses the following set of desirable features: maintaining connectivity of resulting skeletons; producing skeletons of unit width; insensitive to noise; and time-efficient.

In real world there is a need for thinning of images due to following reasons:

a) Thinning reduces the amount of data required to be processed.
b) Thinning reduces the time required to process the pattern.
c) Shape analysis can be more easily made on the thin line like patterns

In image processing and pattern recognition, a binary digitized pattern can be represented by a matrix, where each element is either 1 (dark point) or 0 white point) and these points are called pixels. It is here assumed that the patterns consist of these elements which have value 1. 'Thinning' is a process that deletes the dark points and transforms the pattern into 'thin' line drawing known as a skeleton. [2]

All thinning algorithms can be classified as one of two broad categories:

1. Iterative thinning algorithms or
2. Non-iterative thinning algorithms.

In general, iterative thinning algorithms perform pixel-by-pixel operations until a suitable skeleton is obtained. Iterative algorithms may be further classified as: 1) Sequential Thinning Algorithms 2) Parallel Thinning Algorithms as shown in Figure 3.

In this paper, we will study selected iterative parallel thinning algorithms. [4], [5], [6], [7].

Non-iterative thinning methods use methods other than sequential pixel scan of the image.

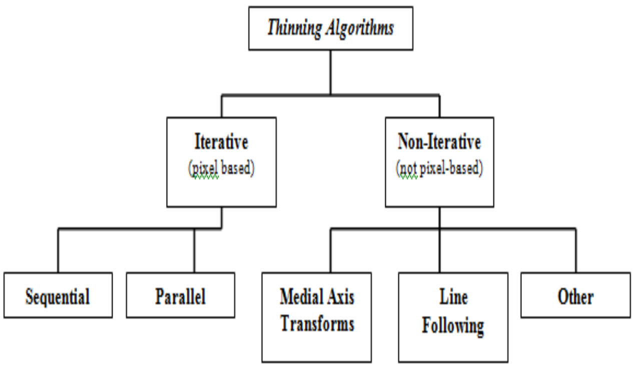

Figure 3: Classification of Thinning Algorithms [8]

II. ITERATIVE PARALLEL THINNING ALGORITHMS

In Iterative parallel thinning algorithms, the decision for individual pixel deletion is based on the results of the previous iteration. In parallel thinning, the value of a pixel at the n^{th} iteration depends on the values of the pixel and its neighbors at the $(n-1)^{th}$ iteration. Thus all the pixels of the pattern can be processed simultaneously. Like sequential algorithms, parallel thinning usually considers a 3*3 neighborhood around the current pixel. A set of rules for deletion of extra pixels are applied based on pixels in the neighborhood. In the next subsections we will discuss the iterative thinning algorithms selected for this study [4], [5], [6], [7].

A. Fast Parallel Algorithm for Thinning Digital Patterns

A Fast Parallel Thinning Algorithm is proposed by T. Y. Zhang and C. Y. Suen in 1984 [4]. It consists of two subiterations: one aimed at deleting the south-east boundary points and the north-west corner points while the other one is aimed at deleting the north-west boundary points and the south-east corner points as shown in Table I [4]. End points and pixel connectivity are preserved. Each pattern is thinned down to a "skeleton" of unitary thickness.

TABLE I
3X3 TEMPLATE FOR PIXEL P1 [4]

p_9 (i-1,j-1)	p_2 (i-1,j)	p_3 (i-1,j+1)
p_8 (i,j-1)	p_1 (i, j)	p_4 (i,j+1)
p_7 (i+1,j-1)	p_6 (i+1,j)	p_5 (i+1,j+1)

Algorithm: The method for extracting the skeleton of a picture consists of removing all the contour points of the picture except those points that belong to the skeleton. In order to preserve the connectivity of skeleton, each iteration is divided in to two sub-iterations.

In the first sub iteration, the contour point p_1 is deleted from the digital pattern. If it satisfies following conditions: [4]

(a) $2 \leq B(p_1) \leq 6$
(b) $A(p_1) = 1$
(c) $p_2 \times p_4 \times p_6 = 0$
(d) $p_4 \times p_6 \times p_8 = 0$

In the second sub iteration, only condition (c) and (d) are changed and rest two conditions remain the same as follows: [4]

(c') p2 x p4 x p8 = 0
(d') p2 x p6 x p8 = 0

where $A(p_1)$ is the number of 01 patterns in the ordered of p2,p3,p4,…p8,p9 that are the eight neighbors of p1 (see Table I), and B(p1) is the non-zero neighbors of p1, that is:

$$B(p1) = p2+p3+\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots+p9.$$

If any condition is not satisfied then p1 is not deleted from the picture. By condition (c) and (d) of the first sub iteration it will be shown that the first sub iteration removes only the south-east boundary points and the north-west corner points which do not belong to an ideal skeleton.

B. Guo and Hall's Parallel Thinning Algorithm

Zichang Guo and Richard W. Hall proposed two thinning algorithms in 1989 [5]. Only the first one, "*Parallel thinning with two sub-iteration algorithms*", will be discussed here. Using the labels as shown in Table II, Guo and Hall's algorithm is described below: [5]

TABLE II
3X3 TEMPLATE USED FOR PIXEL REMOVAL [5]

p8	p1	p2
p7	P	p3
p6	p5	p4

Algorithm*:* Let C (P) be the number of distinct 8-connected components of 1's in Ps 8-neighborhood.

N(P) =Min(N1(P), N2(P))
N 1(P) = (p1vp2)+(p3vp4)+(p5vp6)+(p7vp8)
N2 (P) = (p2vp3)+(p4vp5)+(p6vp7)+(p8vp1)

An edge point will be deleted if it satisfies:
a) ***C(P)=l;***
b) 2<=N(P)<=3;
c) Apply one of the following:
*1) (P2vP3vP5)*P4=0 in odd iterations; or*
*2) (P6vP7vP8)*P8=0 in even iterations*

where "v" expresses the logic "OR" operation. C(P)=1 means P is 8-simple.

In other words, there is only one group of 8-connected 1's around P. Under this condition, deletion of P will not break the connectivity of the elements in the 3*3 window under processing. Condition (a) guarantees P is not a break point. The GH algorithm is better in detecting the end points than the ZS [4] algorithm. The use of N(P) allows one to identify the end points whether or not they have one or two 1's 8-neighbors

C. *A Robust Parallel Thinning Algorithm For Binary Images*

This thinning algorithm was proposed by A. Datta and S. K. Parui in 1994 [6]. The thinning algorithm proposed here removes the outer layer of pixels in multiple passes. In each pass it removes a subset of the boundary pixels in parallel and the output is passed to the subsequent pass.

Algorithm: Based on the theorems and definitions mentioned, the thinning algorithm can be described as follows:

***Step* 1**. Remove all pixels in the input pattern in the following four sub-steps (passes):

(a) Remove all pixels that match template (a) and are removable; [6]
(b) Remove all pixels that match template (b) and are removable; [6]
(c) Remove all pixels that match template (c) and are removable; [6]
(d) Remove all pixels that match template (d) and are removable. [6]

***Step* 2.** Go on repeating Step 1 until no further changes occur in the image pattern, that is, no further pixels are removed. Note that the templates (a)-(d) ensure that only boundary pixels are removed and hence no hole is formed during any pass of the iterations, while the other criteria of a skeleton are met by the following proposition.

Proposition: After a finite number of iterations the algorithm converges and produces a non-empty skeleton which is 1 pixel thick and preserves the connectivity of the original object pattern.

D. *A Preprocessing Algorithm For Hand-Written Character Recognition*

This algorithm is proposed by W.H Abdulla, A.O.M. Saleh and A.H. Morad in 1988 [7] for extracting a simplified skeletal version of hand-written characters. This procedure is required to reduce the hand-written character into the unitary skeleton form each element in the picture. Each element is assigned the value ‘1’ if it is covered by part of the character, and the value ‘0’ otherwise.

Algorithm

This procedure involves two iterations as follows:

Iteration 1:

The skeleton is scanned horizontally by the 3 x 4 pixels window as shown in Table III. Any two points which are horizontally adjacent to each other and horizontally isolated from other points, are detected. With p_1 and p_4 representing these two points, apply the following test whether one of them is redundant.

TABLE III

SKELETON 3X4 FOR ITERATION 1 [7]

p9	p2	p3	p10
p8	p1	p4	p11
p7	p6	p5	p12

P_1 is deleted if one of the following conditions is true:
1. SP_1 and p_6=1:
2. SP_2 and p_2=1:
3. [(P_2 and P_3) or (P_3 and P_2 and P_9)] and [(P_5 and P_6) or (P_5 and P_6 and P_7)]

Where SP1=P_3 or P_2 or P_9. SP_2= P_6 or P_5 or P_7.and’ and ‘or’ are logical ‘AND’ and logical ‘OR’ respectively.

If p1 is not redundant then p4 must be deleted if the following condition is not true:
(P_3 and P_{10}) or (P_5 and P_{12})

Iteration 2:

In the second sub iteration the thin is scanned vertically by the 4*3 pixel window as shown in Table IV.

TABLE IV

SKELETON 4X3 FOR ITERATION 2 [7]

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5
p_{12}	p_{11}	p_{10}

Any two points which are vertically adjacent to each other and vertically isolated from other points are detected.

With p1 and p6 representing these points, apply the following tests to locate the redundant point.

P1 is deleted if one of the following conditions is true:
1. SP11 and p4=1:
2. SP22 and p8=1:
3. [(P8 and P7) or (P7 and P8 and P9)] and [(P4 and P5) or (P5 and P4 and P3)]
where SP11 =P9 or P8 or P7, SP22= P3 or P4 or P5, ‘and’ and ‘or’ are logical ‘AND’ and logical ‘OR’ respectively.

If p1 is not redundant then p6 must be deleted if the following condition is not true: (P7 and P12) or (P5 and P10).

III. Comparative Analysis Of Thinning Results

Due to the increased number of thinning algorithms, the choice of algorithm for an application has become very difficult. For this reason, it is proposed to evaluate the performance of four parallel thinning algorithms [4], [5], [6], [7] and to examine the effects. The algorithms are chosen for their significance and representation of different modes of operation in parallel thinning.

The performance of these iterative parallel thinning algorithms can be compared and evaluated on the basis of following parameters:

a) Quality of Skeleton/Thinned pattern [10]
b) Convergence of the thinned image to unit width[11]
c) Connectivity of Pattern.[11]
d) Data Reduction Rate(DRR)[10]

The selected algorithms are implemented in C Language and results are compared to evaluate the best thinning algorithm. The experiments are performed on 20 different 64 x 99 sized bmp images containing different types of alphanumeric characters. The different parallel thinning algorithms give different results in terms of maintaining the connectivity and convergence to one pixel width. The following subsections shows the results based on the above said factors.

A. Quality of Skeleton/Thinned Pattern

The quality of character pattern after applying the above said thinning algorithms [4], [5], [6], [7] has been compared using images of typical alphanumeric test patterns. The result of Implementation of algorithms and their evaluation shows that the robust parallel thinning algorithm (RPTA) provides us the better quality thinned pattern.

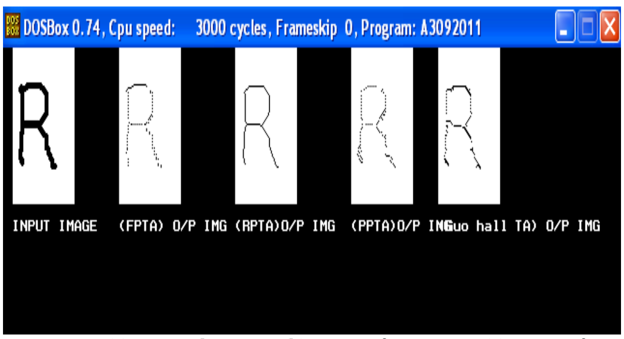

Figure 4: (a) Original Pattern, (b) Pattern from FPTA (c) Pattern from RPTA, (d) Pattern from PPTA, (e)Pattern from GHPTA

As shown in Figure 4(a) and 5(a) represents the Original binary image pattern. After the application of three above said algorithms, we get the skeleton shown in Figure 4(b) & 5(b), 4(c) & 5(c), 4(d) & 5(d), 4(e) & 5(e) respectively. The experiment shows that pattern obtained from RPTA provides the best thinned patterns in terms of quality. This experiment is applied on 20 bmp images which contain different alphanumeric characters.

B. Convergence to Unit width

A good thinning algorithm is expected to produce skeletons of unit width. This will lead to better feature extraction results because it is a crucial step in most character recognition algorithms. The existence of extra pixels in the resulting skeleton may lead to not extracting some of the features. Therefore, unit-width property is very desirable to be considered as one of the evaluation criteria. [11]

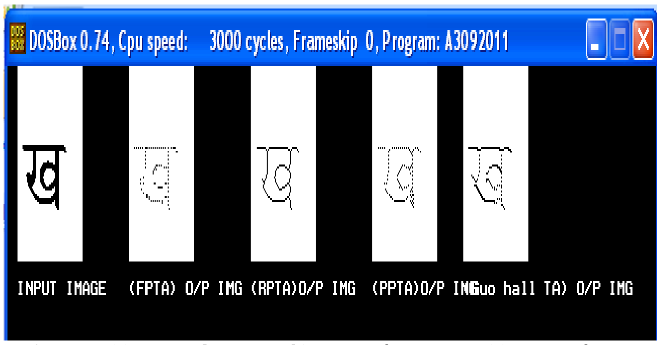

Figure 5: (a) Original Pattern, (b) Pattern from FPTA (c) Pattern from RPTA, (d) Pattern from PPTA, (e)Pattern from GHPTA

The skeletons obtained from thinning process shows that RPTA converges the pattern to unit width whereas the other three algorithms GHPTA, FPTA and PPTA, as shown in Fig. 4 and Fig. 5, do not provide the skeletons of unit width.

C. Connectivity in Pattern

By comparing the original images with their skeletons obtained by four algorithms [4], [5], [6], [7], we can see that the connectivity (i.e. represents the topology features of the images [11]) is kept unchanged after thinning.

From the thinning results of Image 'R', image 'KH' in Hindi Lang., Image 'Q' as shown in Figure 4(b,c,d,e), 5(b,c,d,e), 6(b,c,d,e) respectively, it is clear that the RPTA algorithm generate better results in the terms of connected patterns. It generates both the qualitative as well as complete thinned pattern. The FPTA results in disconnected pattern but better than PPTA and GHPTA. For the PPTA algorithm, it is desired to acquire images in good condition. Additionally it does not result in unit width pattern. GHPTA and PPTA show the noisy pattern as shown in Figure 6. GHPTA can not properly thin the diagonal lines aligned at 45 and 135 degree angles. Therefore GHPTA results in connected but noisy patterns, still better than FPTA.

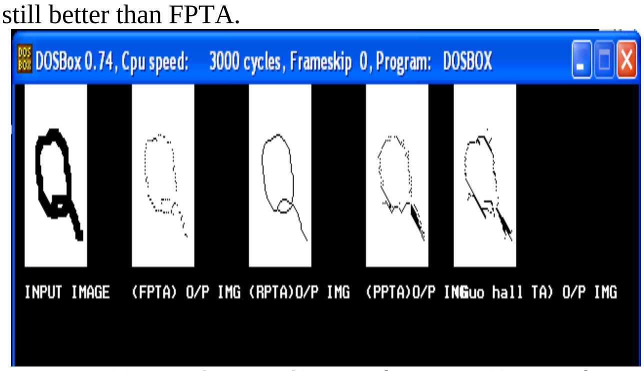

Figure 6: (a) Original Pattern, (b) Pattern from FPTA (c) Pattern from RPTA, (d) Pattern from PPTA, (e) Pattern from GHPTA

D. Data Reduction Rate

The algorithm which produces perfect skeleton will certainly guarantee the highest data reduction value. It reveals information about how good is the algorithm in data reduction when comparing the skeleton S and the original pattern P. Formally: this can be measured as follows [10]:

$$M_{dr} = |S|/|P|$$

It should be noted that this measure is to be considered if the resulting skeleton is acceptable. This means the resulting skeleton must preserve connectivity and reveal the shape of the original pattern.

Following Table V and Figure 7 shows the Data Reduction rate results for four thinning algorithms. [4]- [7]

TABLE V
DATA REDUCTION RATE

BMP Images	RPTA	FPTA	PPTA	GHPTA
1.bmp	0.21	0.14	0.14	0.21
4.bmp	0.21	0.17	0.14	0.26
5.bmp	0.15	0.09	0.17	0.21
6.bmp	0.15	0.07	0.16	0.17
7.bmp	0.33	0.20	0.20	0.33
8.bmp	0.28	0.16	0.19	0.27
9.bmp	0.11	0.08	0.20	0.16
10.bmp	0.12	0.07	0.08	0.18

That is, only that pattern is beneficial for us which is acceptable by the system in terms of shape of thinned image, connectivity etc., rather than that thinned pattern which is not acceptable but results in reduced data rate.

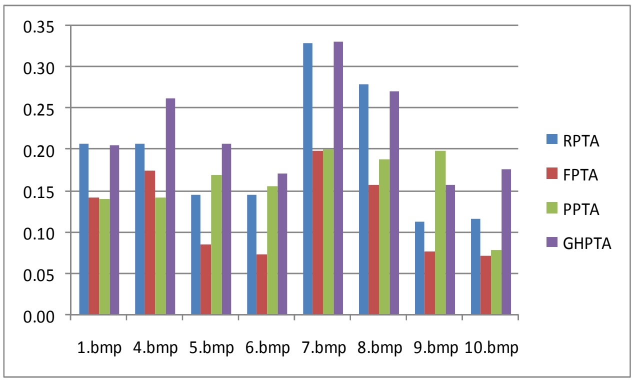

Figure 7: Data Reduction Rate

Figure 7 (above) shows that FPTA results in best data reduction rate as compared to other three. PPTA is on second rank and RPTA on the third and GHPTA on the last rank, but as per the requirement of recognition system only that algorithm results in best thinning which produces the acceptable pattern. Thus RPTA is selected as the best algorithm which thins the pattern betterly and qualitatively.

IV. CONCLUSION

In this research paper, four iterative parallel thinning algorithms [4], [5], [6], [7] are studied and implemented using C language and compared.

The RPTA algorithm is better than the FPTA, GHPTA and PPTA algorithms. But the skeletons of this RPTA algorithm are not as good as that of other three algorithms in terms of Data reduction rate. The RPTA algorithm obtains better skeletons than that of the other algorithms. Its noise resistant ability is higher and it results in completely connected patterns, but the DRR is found to be lowest, at least in this study.

For future work we plan to study the performance of more recent thinning algorithms. Using the lessons learned from these studies, we will try to design new thinning approach that is more accurate and efficient than the existing methods.

REFERENCES

[1]. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1986

[2]. Lawrence O'Gorman and Rangachar Kasturi, *Document Image Analysis*, IEEE Computer Society Executive Briefings, 1997

[3]. L. Lam, S.W. Lee and C.Y. Suen, "*Thinning Methodologies -A Comprehensive Survey*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, Sept. 1992, pp. 869-885

[4]. T. Y. Zhang and C. Y. Suen, "*A fast parallel algorithm for thinning digital patterns*", *Comm. of the ACM* 27, 1984, pp. 236-239

[5]. Zichang Guo and Richard W. Hall, "*Parallel Thinning with Two-Subiteration Algorithms*," *Communications of ACM, vol. 32(3)*, March 1989, pp. 359-373.

[6]. A Datta and S. K. Parui, *"A Robust Parallel Thinning Algorithm for Binary Images."* Pattern Recognition, vol. 27, no. 9, March 1994, pp.1181-1192.

[7]. W.H.Abdulla,A.O.M. Saleh and A.H. Morad, "*A preprocessing algorithm for hand-written character recognition*", Pattern Recognition Letters 7, Jan 1988, pp. 13-18

[8]. E. Hastings, "*A Survey of Thinning Methodologies,*" Pattern analysis and Machine Intelligence, IEEE Transactions, vol. 4, Issue 9, 1992, pp. 869-885.

[9]. C. M. Holt, A. Stewart, M. Clint and R. H. Perrott, "*An improved parallel thinning algorithm*", Comm. of the ACM 30, 1987, pp. 156-160.

[10]. N.H.Han, C.W.La, and P.K.Rhee, "*An Efficient Fully Parallel Thinning Algorithm*," in Proc. IEEE Int. Conf. Document Analysis and Recognition, Vol. 1, 1997, pp.137-141.

[11]. Saad Harous and Ashraf Elnagar "*Handwritten Character-Based Parallel Thinning Algorithms: A Comparative Study*", University of Sharjah Journal of Pure & Applied Sciences, Volume 6, No. 1, February 2009, pp. 81-101

[12]. Gulshan Goyal, Maitreyee Dutta, Akshay Girdhar, "*A Parallel Thinning Algorithm for Numeral Pattern Images in BMP Format*", International Journal of Advanced Engineering & Application, Jan. 2010, pp. 197-202