www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume - 3 Issue -9 September, 2014 Page No. 8102-8108

Revised Reliable Algorithm with Results & Algorithm for Thinning Numeral Pattern

Er. Anup, Er. Sahil Verma, Dr. Kamal Sharma

M-Tech Student
Asst. Prof. in C.S.E. Deptt.
EMGOI , Badhauli.
sahilkv4010@yahoo.co.in
Prof. & Director of E.C.E. Deptt.
EMGOI , Badhauli.

Abstract

thinning algorithms have played an important role in the preprocessing phase of OCR systems.. Many algorithms for vectorization by thinning have been devised and applied to a great variety of pictures and drawings for data compression, pattern recognition and raster-to-vector conversion. The vectorization algorithms often used in pattern recognition tasks also require one-pixel-wide lines as input. But parallel thinning algorithms which generate one-pixel-wide thinnings can have difficulty in preserving the connectivity of an image or generate spurious branches. A few most common thinning algorithms have been implemented and evaluated on the basis of performance parameters.

Introduction to thinning

It is a morphological operation that is used to remove selected foreground pixels from binary images and is particularly useful for thinning. Thinning is normally only applied to binary images, and produces another binary image as output. The term '**thinning**' has been used in general to denote a representation of a pattern by a collection of thin arcs and curves. Other nomenclatures have been used in different context.

For example the term '**medial axis**' is used to denote the locus of centers of maximal blocks Some authors also refer to a '**thinned image**' as a line drawing

representation of pattern. In recent years, it appears that **thinning** and **thinning** have become synonyms in the literature, and the term '**thinning**' is used to refer to the result, regardless the shape of the original pattern or the method employed. Thus, "**THINNING"** is defined as process of reducing the width of pattern to just a single pixel. This concept is shown in figure 1.

Like other morphological operators, the behavior of the thinning operation is determined by a **structuring element**. The choice of structuring element determines under what situations a foreground pixel will be set to background, and hence it determines the application for the thinning operation. For example, consider the structuring elements as shown in figure 2.

Fig. 1. set of objects with thinnings superimposed.

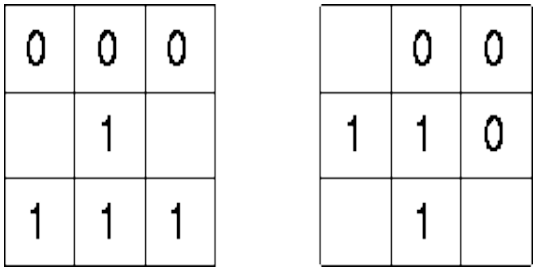

Fig. 2. Structuring element for thinning.

Figure 3 shows the result of this thinning operation on a simple binary image.

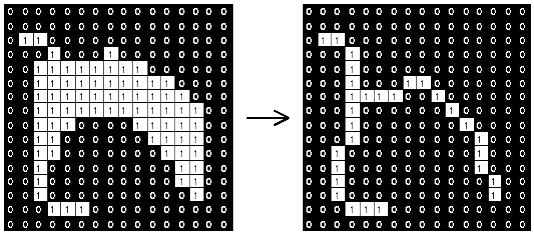

Fig. 3. Thinning of a simple binary shape, using the above structuring elements. Note that the resulting thinning is connected.

1.1 Why thinning

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

1. To reduce the amount of data required to be processed.
2. To reduce the time required to be processed.
3. Extraction of critical features such as end-points, junction-points, and connection among the components .
4. The vectorization algorithms often used in pattern recognition tasks also require one-pixel-wide lines as input.
5. Shape analysis can be more easily made on line like patterns.

1.2 Applications

- Handwritten and printed characters
- Fingerprint patterns
- Chromosomes & biological cell structures
- Circuit diagrams
- Engineering drawings.

2. Some preliminary definitions

Input image is represented by black pixels and white pixels. Black pixels and white pixels are denoted as 1's and 0's, respectively.

Definition 1: A pixel p has 4- neighbours denoted as X_3, X_5, X_7 and X_9 as shown in fig 4.

Definition 2: In addition to four neighbours described in definition 1, a pixel p has four diagonal neighbours denoted as X_2, X_4, X_6 and X_8 as shown in fig 4. These are collectively known as 8- neighbours of a pixel p.

Definition 3 : Two pixels p_1 and p_2 with a common value aresaid to be 8- connected(4-connected) if a sequenceof pixels $a_0(=p_1)$, a_1,

X_4	X_3	X_2
X_5	p	X_9
X_6	X_7	X_8

Fig. 4. Neighbor pixels of N(p)

….,a_n $(=p_2)$ exists such that each a_i is 8-neighbour(4-neighbour) of a_{i-1} ($1 \leq i \leq n$) and all a_i have the same values as p_1 and p_2.

Definition 4 : Connectivity has been defined by the following : two 1's are connected if they are 8- connected, and two 0's are connected if they are 4
connected.

Definition 5 : A pixel p is deletable if its removal does not change 8-connectivity of p, otherwise the pixel is said to be undeletable

THINNING ALGORITHMS

According to the way we examine pixels, these algorithms can be classified as '**Sequential**' and '**Parallel**'. In sequential algorithm, the pixels are examined for deletion in a fixed sequence in each iteration, and the deletion of p in the nth iteration depends on all the operations performed so far, i.e. on the results of (n-1)th iteration; as well as on the pixels already processed in (n)th iteration. In a parallel algorithm, the deletion of pixels in the nth iteration depends only on the result of nth iteration; therefore, all the pixels can be examined independently in the parallel manner in each iteration.

Many thinning algorithms (or modifications of existing ones) have been proposed in recent years. Here we discuss two parallel algorithms that can be applied to numerals.

3.1 FAST PARALLEL ALGORITHM FOR THINNING DIGITAL PATTERNS

A binary digitized picture is defined by a matrix IT where each pixel IT(i,j) is either 1 or 0. the pattern consists of those pixels that have value 1. Each stroke in the pattern is more than one element thick. Iterative transformation are applied to matrix IT point by point according to values of a smallset of neighbouring points.

In parallel picture processing the new value given to a point at n^{th} iteration depends on its own value as well as those of its 8 neighbours at the $(n-1)^{th}$ iteration, so that all picture points can be processed simultaneously. It is assumed that a 3X3 windiow is used, and that each element is connected with its 8-neighbouring elements. This algorithm requires
only simple calculations.The method for extracting the skelton of a picture consists of removing all the contour points of the picture except those points that belong to the skelton. In order to preserve the connectivity of skelton,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:

(a) $2 \leq B(p_1) \leq 6$

(b) $A(p_1) = 1$

(c) $p_2 \times p_4 \times p_6 = 0$

(d) $p_4 \times p_6 \times p_8 = 0$

where $A(p_1)$ is the number of 01 patterns in the ordered of p2,p3,p4,…p8,p9 that are the eight neighbourss of p1 (fig. Below), and B(p1) is the non-zero neighbourss of p1, that is

B(p1) = p2+p3+…+p9.

P9 (i-1,j-1)	p2 (i-1,j)	p3 (i-1,j+1)
P8 (i,j-1)	p1 (i,j)	p4 (i,j+1)
P7 (i+1,j-1)	p6 (i+1,j)	p5 (i+1,j+1)

Fig.5 Designation of 9 pixels in 3X3 window:

If any condition is not satisfied then p1 is not deleted from the picture.In the seond subiteration, only condition (c) and (d) are changed as follows

(c') p2 x p4 x p4 = 0

(d') p2 x p6 x p8 = 0

and the rest remain the same.

By condition (c) and (d) of the first subiteration it will be shown that the first subiteration removes only the south- east boundry points and the north-west corner points which do not belong to an ideal skelton. By condition (a), the end-points of a skelton line are preserved. Also, condition (b), prevents the deletion of those points that lie between the end-point of skelton line.

The iterations continue until no more points can be removed.

3.2 PREPROCESSING THINNING ALGORITHM

This algorithm reduces the hand-written character into the unitary thin form. Each element is assigned the value '1' if it is covered by part of the character, and the value '0' otherwise. This algorithm involves two sub iterations [9]. In the first sub

iteration the thinning is scanned horizontally by the 3*4 pixels window. Any two points which are horizontally adjacent to each other and horizontally isolated from other points, are detected. With p1 and p4 representing terse two points, apply the following test whether one of them is redundant.

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}).

P9	P2	P3	P10
P8	P1	P4	P11
P7	P6	P5	P12

Fig.6 A 3×4 pixel window.

In the second sub iteration the thin is scanned vertically by the 4*3 pixel window. 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

P_1 is deleted if one of the following conditions is true:

1. SP_{11} and p_4=1:

2. SP_{22} and p_8=1:

3. [(P_8 and P_7) or (P_7 and P_8 and P_9)] and [(P_4 and P_5) or (P_5 and P_4 and P_3)]

Where SP_{11} =P_9 or P_8 or P_7,

SP_{22}= P_3 or P_4 or P_5,

'and' and 'or' are logical 'AND' and logical 'OR' respectively.

If p_1 is not redundant then p6 must be deleted if the following condition is not true:

(P_7 and P_{12}) or (P_5 and P_{10})

P9	P2	P3
P8	P1	P4
P7	P6	P5

P12	P11	P10

Fig.7 A 4×3 pixel window.

4. Performance evaluation parameters and comparison of two thinning algorithms.

Due to the proliferation of these algorithms, the choice of algorithm for an application has become very difficult, and a researcher in this area is often faced with the question of *which algorithm to* ***use.*** For this reason, we propose to evaluate the performance of two thinning algorithms and to examine the effects based on real-life data. The algorithms are chosen for their significance and representation of different modes of operation in parallel thinning. The performance of these algorithms is evaluated on the basis of following parameters:

1. Convergences of the thinned image to a unit width thinning.
2. connectivity.
3. spurious branches
4. Processing time.

We compare the These algorithms are compared in terms of:

1. Convergences of the thinned image to a unit width thinning
2. Connectivity of pixels in the thinned image.
3. Spurious branches that may be produced.
4. The time taken for execution.

The algorithms are chosen for their significance and representation of different modes of operation in parallel thinning. In order to conduct an experiment of considerable scope, the following procedure has been adopted:

1) Each algorithm is implemented using a suitable programming language with a verification of the results.

2) Each algorithm is used to thin the pattems of hand written regional language.

3) Results and observations are recorded from thinning these large sets of data.

The main features of parallel algorithms are as described below:

4.1 Measures of convergence to unit width

A thinning algorithm is perfect if it can generates one-pixel-wide thinnings.It is obvios that if the converged thinning S_M does not contain any one of the pattern Q_K as shown in figure 8, then S_M is one pixel wide. To measure the width of the resultant skelton , m_t is defined as:

$$m_t = 1 - \frac{\text{Area}\left[\bigcup_{1 \le k \le 4} S_M Q^k\right]}{\text{Area}[S_M]}$$

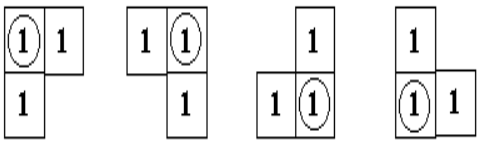

Fig. 8. Template Q_K ($1 \le k \le 4$) are used to examine the width of converge Where , Area [•] is the operation that counts the number of one-pixel that have the values true or '1'. This measure has a non- negative value less than or eqal to 1, with $m_t = 1$, if
S_M is aperfect unit width skelton.

4.2 Connectivity

Preserving connectivity of a connected component is essential for shape analysis.the topological features of an 8- connected pattern may change completely if it becomes disconnected. Therefore a connected component must have a corresponding connected thinning.

4.3 Spurious branches

The spurious branches refers to the extraneous branches that may be generated as an output of thinning process. A good thinning algorithm should be capable of avoiding generation of spurious branches.

4.4 Execution time

The computational cost or the time required by a thinning algorithm for skeltonization of the original pateern is another important factor. This time is actually the amount of CPU time utilized by the thinning algorithm. Number of example images are considered for the purpose of measuring the computational cost. The factor m_c i.e. the total CPU time used divided by the area of origanal image is considered as a facyor for comparison i.e.

$$m_c = \frac{\text{CPU time}}{\text{Area [•]}}$$

Where Area [•] is the number of dark point or object pixel in the original image.

Figure 9 shows how connectivity is preserved using parallel two algorithms

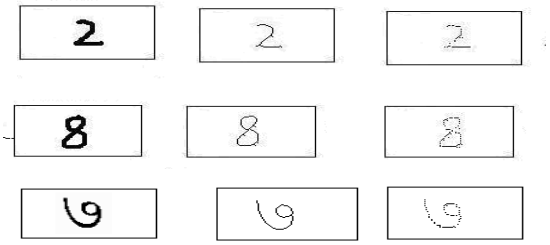

Fig.9 (a) orginal image(b) algorithm2 (c) algorithm1

The discussion of various aspects such as convergence to unit width, connectivity, spurious branches and execution time can be taken together to conclude which of these two algorithms cosidered is better for regional language numerals for the purpose of thinning. The results obtained by applying the above discussed parallel thinning algorithms are shown in figure 9. this suggests that algorithm 2 preserves connectivity better than algorithm1. In future the information loss can be studied and thinning and contour can be incorporated.

5 . IMPROVED PARALLEL THINNING ALGORITH FOR NUMERAL PATTERNS

X_4	X_3	X_2
X_5	p	X_1
X_6	X_7	X_8

Fig.10 8-Adjacency set of a pixel

The first algorithm belongs to class of multi-pass iterative boundary removal thinning algorithms. Iterative boundary removal algorithms delete pixels on the boundary of a pattern repeatedly until only unit pixel-width thinned image remains.When a contour pixel is examined, it is usually deleted or retained according to the configuration of N(p) shown in figure 10. To prevent sequentially eliminating an entire branch in one iteration, a sequentially algorithm usually marks(or flags) all the pixels to be deleted, and all the marked pixels area then removed at the end of an iteration. This generally ensures that only one layer of pixels would be removed in each cycle.

The method for extracting the skelton of a picture consists of removing all the contour points of the picture except those points that belong to the skelton. In order to preserve the connectivity of skelton, each iteration is divided in to two sub-iterations.The pattern is scanne left to right and from top to bottom , and pixels are marked for deletion under four additional conditions:

H1: At least one black neighbour of p must be unmarked.

H2: $X_h(p) = 1$ at the beginning of the iteration.

H3: If x_3 is marked, setting $x_3=0$ does not change $X_h(p)$.

H4: If x_5 is marked, setting $x_5=0$ does not change $X_h(p)$.

Condition H1 was designed to prevent excessive erosion of small "circular" subsets, H2 to maintain connectivity, andH4 to preserve two- pixel wide lines.

6. RESULT AND DISSCUSSION

A database consisting of hand-written roman, devnagri and gurumukhi numerals from different users is used to visualize the performance of parallel thinning algorithms. The two parallel thinning algorithms when applied on the numeral pattern database provide the results reflecting poor connectivity. The results do not converge to unit pixel wide thinnings. provide the results reflecting poor connectivity. The results do not converge to unit pixel wide thinnings.These results are shown in Fig. 9. But after applying the proposed alternative parallel thinning algorithm, the results are shown in Fig. 11. The differences in the outputs are very much clear from the visual inspection.

Fig.11 (a) Algo1 (b) Algo2 (c) Algo3

The results shown in Fig. 11 shows that the proposed alternative parallel thinning algorithm provides better pixel connectivity and convergence to unit pixel width. Presently, the algorithm has been implemented and tesed for BMP format of numeral patterns. In future, the algorithm can be generalized for other commonly available image formats also. Further studied can be carried out to eliminate the spurious branches and to improve the time complexity of the proposed algorithm.

7. REFRENCES

1. A. K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, 1986.
2. R. Gonzalez and R. E. Woods, Digital Image Processing, Prentice Hall, 2002.
3. A. Dutta and S. K. Parui, A robust parallel thinning algorithm for binary images, Pattern recognition,Vol, 27, No. 9, pp, 1181-1192,1994
4. T.J., Zhang and C.Y. Suen, A fast parallel algorithm for thinning digital pattern, Comm. Of ACM 27 , 236-239 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,pp.137-141(1997**).**

5. W. H. Abdulla, A.O.N. Saleh and A.H.Morad, A preprocessing algorithm for hand-written character recognition, pattern recognition letters 7,1998

6. M. Ahmad and R. Ward," An Expert system for General Symbol Recognition," Pattern Recognition, vol. 33, no. 12, pp. 1975-1988,2000.

7. M. Ahmed, R. Ward, A Rotation Invariant Rule-Based Thinning Algorithm for Character Recognition December 2002 (Vol. 24, No. 12) pp. 1672-1678

8. Fingerprint minutiae extraction based on principal curves, Pattern Recognition Letters Volume 28, Issue 16, 1 December 2007, Pages 2184-2189