Skeletonization Algorithm for Numeral Patterns

Gupta Rakesh and Kaur Rajpreet

Department. of CSE, SDDIET Barwala, Panchkula, India
and Department of CSE / IT, BBSBEC, Fatehgarh Sahib, Punjab, India
{Gupta Rakesh and Kaur Rajpreet}

Abstract

Skeletonization has been a part of morphological image processing for a wide variety of applications. Skeletonization algorithms have played an important role in the preprocessing phase of OCR systems. Many algorithms for vectorization by skeletonization 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 skeletonization algorithms which generate one-pixel-wide skeletons can have difficulty in preserving the connectivity of an image or generate spurious branches. In this paper an alternative parallel skeletonization algorithm has been developed and implemented. This algorithm is better than already existing algorithms in terms of connectivity and spurious branches. A few most common skeletonization algorithms have been implemented and evaluated on the basis of performance parameters and compared with newly developed algorithm.

Keywords: Digital image, parallel skeletonization algorithm, structuring element elimination rule, connectivity, execution time, spurious branches, character recognition.

1 Introduction

Skeletonization [1, 2] is a morphological operation that is used to remove selected foreground pixels from binary images. Skeletonization is normally only applied to binary images, and produces another binary image as output. The term ‘skeleton’ 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 ‘skeletonized image’ as a line drawing representation of pattern. In recent years, it appears that skeletonization have become synonyms in the literature, and the term ‘skeleton’ is used to refer to the result, regardless the shape of the original pattern or the method employed. Thus, SKELETONIZATION 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 skeletonization 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 skeletonization operation.

For example, consider the structuring elements as shown in figure 2. Input image is represented by black pixels and white pixels. Black pixels and white pixels are denoted as 1's and 0's, respectively. A pixel p has 4- neighbors denoted as X3, X_5, X_7 and X_9 as shown in fig 4.

In addition to four neighbors described in definition 1, a pixel p has four diagonal neighbors denoted as X2, X_4, $X_{6\ and}$ X8 as shown in fig 4. These are collectively known as 8- neighbors of a pixel p.

Fig. 1. A set of objects with skeletons superimposed.

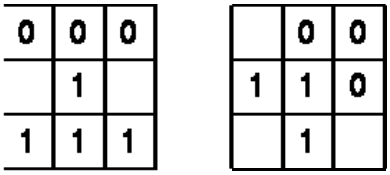


Fig. 2. Structuring element for skeletonization.

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

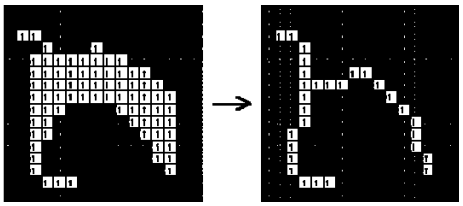


Fig. 3. Skeletonization of a simple binary shape, using the above structuring elements. Note that the resulting skeleton is connected.

A 1.1 Why skeletonization

In real world there is a need for skeletonization 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.

A 1.2 Applications

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

B. Some preliminary concepts

Two pixels p_1 and p_2 with a common value are said 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-neighbor (4-neighbor) of a_{i-1} ($1 \leq i \leq n$) and all a_i have the same values as p_1 and p_2.

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.

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

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

A. Parallel skeletonization algorithm 1

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 as shown in figure 5.

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)

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

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 3*3 windiow is used, and that each element is connected with its 8-neighbouring elements. This algorithm[4] requires only simple calculations.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 into 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 neighbours of p1 (fig. Above), and B(p1) is the non-zero neighbourss of p1, that is

$$B(p_1) = p_2+p_3+\ldots+p_9.$$

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) $p_2 \times p_4 \times p_8 = 0$
(d) $p_2 \times p_6 \times p_8 = 0$
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 skeleton.

By condition (a), the end-points of a skeleton line are preserved. Also, condition (b), prevents the deletion of those points that lie between the end-point of skeleton line. The iterations continue until no more points can be removed.

B. Parallel skeletonization algorithm 2

Each element is assigned the value'1' if it is covered by part of the character, and the value '0' otherwise. Whether a point is thin or not depend on its 8- neighbours. The algorithm [5] involves two iterations as follows:

Iteration 1:

The skeleton is scanned horizontally by the 3*4 pixels window shown in fig 6 below.

Any two points which are horizontally adjacent to each other and horizontally isolated from other points, are detected. With p1 and p4 representing these two points, apply the following test whether one of them is redundant:

P1 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 complement, 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 this iteration the thin is scanned vertically by the 4*3 pixel window shown in figure7 below.

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_9	P_2	P_3	P_{10}
P_8	P_1	P_4	P_{11}
P_7	P_6	P_5	P_{12}

Fig. 6. 3*4 pixel window

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5
P_{12}	P_{11}	P_{10}

Fig. 7. 4*3 pixel window

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 complement, 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})

3. An Alternative Parallel Skeletonization Algorithm

The algorithm belongs to class of multi-pass iterative boundary removal skeletonization algorithms [7, 9]. 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 4. To prevent sequentially eliminating an entire branch in one iteration, a sequential algorithm usually marks (or flags) all the pixels to be deleted and all the marked pixels area then removed at the end of iteration. This generally ensures that only one layer of pixels would be removed in each cycle.

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 into two sub-iterations.The pattern is scanned from 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, and H4 to preserve two- pixel wide lines.

4. Performance Evaluation Parameters and comparison of Skeletonization 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 skeletonization 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 skeletonization. The performance of these algorithms is evaluated on the basis of following parameters [2, 4, 7, 11]:

1. Convergences of the thinned image to a unit width skeleton.
2. Connectivity.
3. Spurious branches.

These algorithms are compared in terms of:
1. Convergences of the thinned image to a unit width skeleton
2. Connectivity of pixels in the thinned image.
3. Spurious branches that may be produced.

The algorithms are chosen for their significance and representation of different modes of operation in parallel skeletonization. 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 skeletonizing these large sets of data.

The main features of parallel algorithms are as described below:

A. Measures of convergence to unit width

A skeletonization algorithm is perfect if it can generates one-pixel-wide skeletons.It is obvious that if the converged skeleton 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 Skeleton, m_t is defined as:

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

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 a perfect unit width skeleton.

B. 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 skeleton.

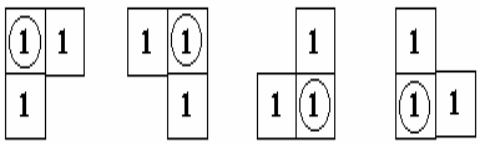
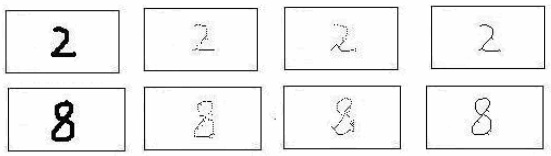


Fig. 8. Template Q_K ($1 \leq k \leq 4$) are used to examine the width of converge.

C. Spurious branches

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



(a) (b) (c) (d)

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

Figure 9 shows how connectivity is preserved using parallel algorithms.

An Alternative algorithm is showing better results in terms of connectivity and spurious branches.

5. Conclusion

The discussion of various aspects such as convergence to unit width, connectivity, spurious branches can be taken together to compare these algorithms for regional language numerals for the purpose of skeletonization. The results obtained by applying the above discussed parallel skeletonization algorithms are shown in figure 9. The differences in the outputs are very much clear from the visual inspection.Newly developed algorithm is better in terms of connectivity and has less generation of spurious branches.

In future the information loss can be studied and skeleton and contour can be incorporated.

6. References

1. Jain, A.K., : Fundamentals of Digital Image Processing, Prentice-Hall, 1986.
2. Gonzalez, R. and Woods, R.E., : Digital Image Processing , Prentice Hall, 2002.
3. Dutta, A. and Parui, S.K., : A robust parallel skeletonization algorithm for binary images, Pattern recognition vol. 27, no. 9, pp, 1181-1192, 1994
4. Zhang, T.J., and Suen, C.Y., : A fast parallel algorithm for skeletonization digital pattern, Comm. of ACM 27 ,pp 236-239.
5. Abdulla, W.H., Saleh, A.O.N. and Morad, A.H., : 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. Ahmed, M., Ward, R., : A Rotation Invariant Rule-Based Skeletonization Algorithm for Character Recognition, December 2002, vol. 24, no. 12, pp. 1672-1678
8. Fingerprint minutiae extraction based on principal curves, Pattern Recognition Letters vol. 28, issue 16, 1 December 2007, pp. 2184-2189.
9. Employing Generic Algorithms for precise Fingerprint Matching based on Line Extraction, GVIP journal, vol. 7, issue 1, April, 2007.
10. Sagar, V.K., Greening, C., Tan, W.Y. and Leung, C.S.A., : Hardward software Co-Design of a Fingerprint Recognition System, Department of Electronic Systems Engineering University of Essex, Colchester CO4 3SQ, UK.
11. Han, N.H., La, C. W., and Rhee, P.K., : An Efficient Fully Parallel Skeletonization Algorithm, in Proc. IEEE Int. Conf. Document Analysis and Recognition, vol. 1, pp. 137-141, 1997.