International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 7, July 2014

An Efficient Fully Parallel Skeletonization Algorithm for Device Independent BMP Images

Sandeep Singh Sangwan[1], Er. Sukhvinder Kaur[2], Dr. Dinesh Arora[3]

***Abstract*— A number of image processing and pattern recognition application demand that are raw digitized binary pattern array be normalized, so that the constituent components of that array are of uniform thickness. The skeletonization process reduces such components to a thickness of one pixel or sometimes to a few pixels. Thin-line representations of elongated patterns would be more amiable to extraction of critical features such as end-points, junction-points, and connection among the components. 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.**

***Index Terms*—Digital Image, parallel skeletonization algorithm, connectivity, spurious branches, character recognition.**

I. INTRODUCTION

Skeletonization is a morphological operation that is used to remove selected foreground pixels from binary images [1,2]. It can be used for several applications, but 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 being used to denote the locus of centers of maximal blocks In recent years, it appears that 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 [3]. This concept is shown in Fig. 1.

The skeletons of one object may belong to several other different elongated objects as shown in Fig. 2.

***First Author name*:** *Sandeep Singh Sangwan, Student, Swami Devi Dyal Institute of Engineering and Technology, Panchkula, Haryana, India, MobileNo:+919417565720.*

***Second Author name*:** *Er. Sukhvinder Kaur,Asst. Prof., Swami Devi Dyal Institute of Engineering and Technology, Panchkula, Haryana, India, Mobile No:+919872074956.*

***Third Author name*:** *Dr. Dinesh Arora, Associate Prof., Gurukul Vidyapeeth, Banur, Punjab, India, Phone/Mobile No. +919814060260.*

Like other morphological operators, the behavior of the skeletonization operation is determined by a structuring element. The binary structuring elements used for skeletonization are of the extended type described under the hit-and-miss transform (*i.e.* they can contain both ones and zeros). The skeletonization operation is related to the hit-and-miss transform and can be expressed quite simply in terms of it. The skeletonization of an image *I* by a structuring element *J* is:

$$\text{thin}(I, J) = I - \text{hit-and-miss}(I, J) \quad (1)$$

Fig. 1 A set of objects with skeletons superimposed.

In everyday terms, the skeletonization operation is calculated by translating the origin of the structuring element to each possible pixel position in the image, and at each such position comparing it with the underlying image pixels. If the foreground and background pixels in the structuring element exactly match foreground and background pixels in the image, then the image pixel underneath the origin of the structuring element is set to background (zero). Otherwise it is left unchanged. Note that the structuring element must always have a one or a blank at its origin if it is to have any e

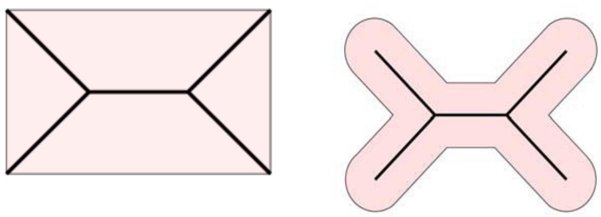


Fig. 2 The different objects with the same skeletons.

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 Fig. 3.

At each iteration, the image is first skeletonized by the left hand structuring element, and then by the right hand one, and then with the remaining six 90° rotations of the two elements. The process is repeated in cyclic fashion until none

ISSN: 2278 – 909X

of the skeleton produces any further change. As usual, the origin of the structuring element is at the center.

0	0	0
	1	
1	1	1

	0	0
1	1	0
	1	

Fig. 3 Example structuring elements for skeletonization.

Fig. 4 shows the result of this skeletonization operation on a simple binary image.

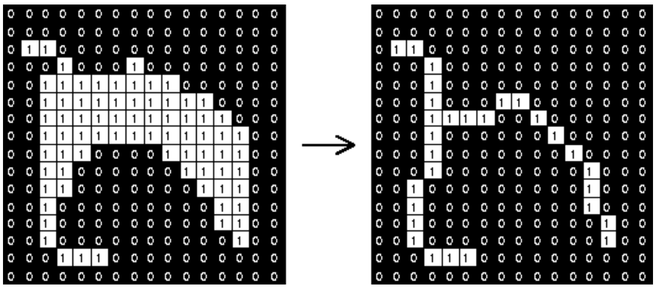


Fig. 4 Skeletonization of a simple binary shape, using the above elements

We have described the effects of a single pass of a skeletonization operation over the image. In fact, the operator is normally applied repeatedly until it causes no further changes to the image (i.e. *until* convergence). Alternatively, in some applications, e.g. pruning, the operations may only be applied for a limited number of iterations.

A. Requirements

The skeletonization process has following requirements:

(a) Geometrical: The skeleton must be in the middle of the original object and must be invariant to translation, rotation, and scale change.

(b) Topological: The skeleton must retain the topology of the original object.

B. Purpose of Skeletonization

The purpose of skeletonization is to reduce the amount of information in image pattern to the minimum needed for recognition. Thinned image helps the extraction of important features such as end points, junction points, and connections from image patterns. In real world there is a need for skeletonization of images due to following reasons:

(a) To reduce the amount of data required to be processed.

(b) To reduce the time required to be processed.

(c) Extraction of critical features such as end-points, junction-points, and connection among the components.

(d) The vectorization algorithms often used in pattern recognition tasks also require one-pixel-wide lines as input.

(e) Shape analysis can be more easily made on line like patterns.

C. Requirements

Skeletonization has numerous applications in image analysis and computer vision. For several of these applications significant amount of information is lost during the process of binarization. Applying skeletonization directly to gray scale images is motivated by the desire of directly processing images with gray levels distributed over a range of intensity values. This will avoid shape distortions that may irremediably affect the presence of features in the binary image generated even if an optimal thresholding algorithm is used to produce the binary image. Skeletonization is a very important technique extensively used in the areas of pattern recognition visual inspection, character recognition, fingerprint recognition etc. The main applications of skeletonization are:

(a) Handwritten and printed characters recognition

(b) Fingerprint patterns recognition

(c) Chromosomes & biological cell structures

(d) Circuit diagrams

(e) Engineering drawings

(f) Signature verification

(g) Raster-to-vector conversion

D. Some Preliminary Concepts

Two pixels p1 and p2 with a common value are said to be 8-connected (4-connected) if a sequence pixels a_0 (=p_1), a_1 , ……… 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.

X_4	X_3	X_2
X_5	P	X_9
X_6	X_7	X_8

Fig. 5 Neighbor pixels of N(p)

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.

II. SKELETONIZATION ALGORITHMS

Skeletonization is defined as a procedure to transform a digital pattern, say, a connected component, to a connected skeleton of unit width. Skeletonization is normally only applied to binary images, and produces another binary image as output. 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 This can be accomplished by either raster scan(s) or by contour following. The contour following algorithms can visit every border pixel of a simply connected object, and of a multiply connected picture, if all the borders of the picture and the holes are followed.

In parallel skeletonization algorithms, the pixels are examined for deletion based on the result of only previous 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. For this reason, these algorithms are suitable for implementation on parallel processors where the pixels satisfying a set of conditions can be removed simultaneously.

These algorithms visit all the pixels in the bitmap to identify the dark points. The dark points are then classified into edge points and non-edge points. Only the edge points need to be considered. Tests are conducted on each point's 8 neighbors to determine whether they are break points, end points or non-safe points. The non-safe points are then removed from the pattern at the end of the pass. The break and end-points are collectively known as safe points and should not be removed.

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 small set of neighboring points as shown in Fig. 6 below.

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. 6 Designation of 9 pixels in Neighbor pixels in 3 × 3 windows

In Parallel picture processing the new value given to a point at nth iteration depends on its own value as well as those of its 8 neighbors at the (n-1) th iteration, so that all picture points can be processed simultaneously. It is assumed that a 3 × 3 window is used, and that each element is connected with its 8-neighbouring elements [3]. This algorithm 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) $2 \leq B(P_1) \leq 6$ (2)
(b) $A(P_1) = 1$ (3)
(c) $P_2 \times P_4 \times P_6 = 0$ (4)
(d) $P_4 \times P_6 \times P_8 = 0$ (5)

Where $A(P_1)$ is the number of 01 patterns in the ordered of $P_2, P_3, P_4 \ldots P_8, P_9$ that are eight neighbors of P_1 (Fig. 6), and $B(P_1)$ is the non-zero neighbors of P_1, i.e.

$$B(P_1) = P_2 + P_3 + \ldots\ldots\ldots + P_9$$
(6)

If any condition is not satisfied then P_1 is not deleted from the picture.

In the second sub iteration, only condition (c) and (d) are changed as follows

(c) $P_2 \times P_4 \times P_8 = 0$ (7)
(d) $P_2 \times P_6 \times P_8 = 0$ (8)

and the rest remain the same.

By condition (c) and (d) of the first sub iteration it will be shown that first sub iteration removes only the south-east boundary 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-points of skeleton line. The iterations continue until no more points can be removed.

B. Parallel Skeletonization Algorithm 2

This algorithm is required to reduce the hand-written character in to the unitary thin form, each element in the picture [4]. Each element is assigned the value'1' if it is covered by part of the character, and the value '0' otherwise. Depending whether a point is thin or not depend on its 8-neighbors. Thus, a window of 3 × 3 pixels used. Firstly, the thin of the character is extracted. The output thin is not unitary as it involves distortion at some points. This distortion will cause difficulties in the detection of edge and tree points, to overcome this drawback a procedure is developed which is capable of producing noise free unitary thin. This procedure involves two iterations as follows:

The algorithm is divided into two sub iterations:

The skeleton is scanned horizontally by the 3 × 4 pixels window shown in Fig. 7 below.

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. 7 A 3 × 4 window

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:

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

(a) SP_1 and P_6 =1:
(b) SP_2 and P_2 =1:
(c) [(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 SP_1=P_3 or P_2 or P_9.
SP_2= P_6 or P_5 or P_7. and ( ) 'and ','or' are

ISSN: 2278 – 909X

complement, logical 'AND' and logical 'OR' respectively.

If P_1 is not redundant then P_4 must be deleted if the following condition is not true:

(P_3 and P_{10}) or (P_5 and P_{12})

In this iteration the thin is scanned vertically by the 4 × 3 pixel window shown in Fig. 8 below.

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. 8 A 4 × 3 pixels window

Any two points which are vertically adjacent to each other and vertically isolated from other points are detected. With P_1 and P6 representing these points, apply the following tests to locate the redundant point.

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

(a) SP_{11} and P_4 =1:

(b) SP_{22} and P_8=1:

(c) [(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 P_6 must be deleted if the following condition is not true:

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

III. AN ALTERNATE PARALLEL ALGORITHM

Describing the shape of objects is often necessary in image processing. While edges are used to represent the boundary of an object, the general shape of the object is represented by "stick figures". The former technique for shape representation of objects is boundary- based. and the latter one is region- based. Both the techniques have attracted researchers in the field of shape analysis and recognition of digital images. Here we propose a shape – based technique for the description of a digital binary object- that is, the process of obtaining the "stick- figures", the so-called skeletonization process from a binary object.

X_4	X_3	X_2
X_5	P	X_1
X_6	X_7	X_8

Fig. 9 8-Adjacency set of a pixel

The algorithm belongs to class of multi-pass iterative boundary removal skeletonization algorithms [9,10]. 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 Fig. 9. 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:

H_1: At least one neighbor of p must be unmarked.

H_2: X_h (p) =1 at the beginning of the iteration.

H_3: If X_3 is marked, setting $X_3 = 0$ does not change X_h (p).

H_4: If X_5 is marked, setting $X_5 = 0$ does not change X_h (p).

Condition H_1 was designed to prevent excessive erosion of small "circular" subsets, H_2 to maintain connectivity, and H_4 to preserve two- pixel wide lines.

IV. RESULTS AND DISCUSSION

The skeletonization algorithm reduces the hand-written regional language numeral into the unitary thin form. When we apply the parallel skeletonization algorithms on Hindi and regional language numerals we visualize the results to observe the convergence to unit width, connectivity and spurious branches. A skeletonization algorithm is not suitable for any application unless it maintains the connectivity of pixels in the thinned pattern. Therefore we discussed an alternative parallel skeletonization algorithm. When we apply the same database of regional language numerals we observe the difference in the results. The results are shown in Fig. 12.

A. Performance Evaluation Parameter

Due to the proliferation of skeletonization 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 connectivity.

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:

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

Each algorithm is used to thin the hand written regional language numeral patterns.

Visual observations are recorded for skeletonization of these large sets of data.

B. Comparison of Skeletonization Algorithms Parameters

The performance of these algorithms is evaluated on the basis of following parameters [1,3,7,9]:

(a) Convergences of the thinned image to a unit width skeleton.
(b) Connectivity of pixels in the skeletonized image.
(c) Spurious branches that may be produced.

The main features of parallel algorithms are as described below:

B.1 Measures of convergence to unit width

A skeletonized 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 the pattern Q_K as shown in Fig. 10 then S_M is one pixel wide. To measure the width of the resultant Skeleton, m_t is defined as:

$$m_t = 1 - \frac{\text{Area}\left[\bigcup_{1\le k\le 4} S_M Q^k\right]}{\text{Area}\left[S_M\right]} \quad (5.1)$$

where Area [$^\bullet$] is the operation that counts the number of one-pixel that have the values true of '1'. This measure has a non-negative value less than or equal to 1, with $m_t = 1$, if S_M is a perfect unit width skeleton.

B.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 skeleton.

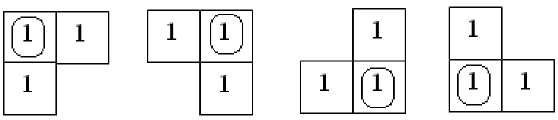


Fig. 10 Template Q_k ($1\le k \le 4$) are used to examine the width of convergence.

B.3 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.

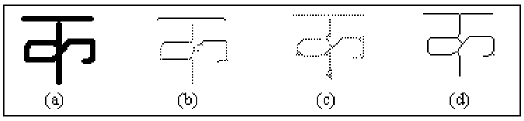
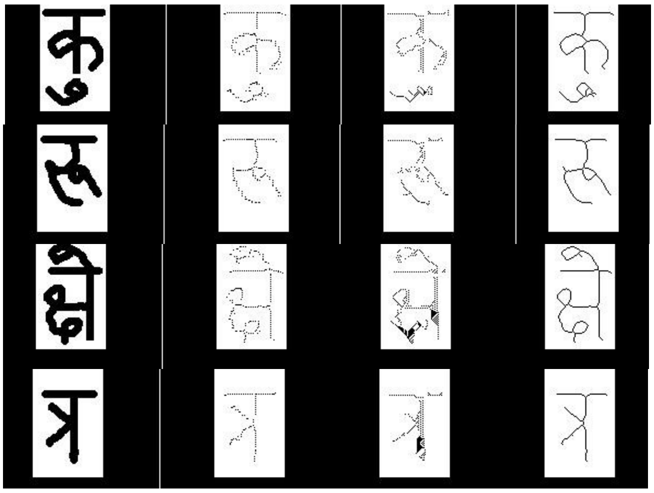


Fig. 11 (a) Input Image (b) Algorithm 1 (c) Algorithm 2 (d) Alternative Algorithm

Fig. 11 shows how connectivity is preserved using parallel algorithms. The proposed alternative parallel skeletonization algorithm is showing better results in terms of connectivity and has less generation of spurious branches.

The iterative phases are clearly aimed at deleting pixels in a specific neighborhood, therefore a general idea of this method is easily found. The universality of the algorithm allows wide and diversified applications.

C. Results of Alternative Skeletonization Algorithm



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

Fig. 12 Skeletonization of Hindi Characters 'Ku', 'Ru', 'She', 'Tra'
(a) Input Image (b) Algorithm 1 (c) Algorithm 2 (d) Alternative Algorithm

The quality of character pattern after applying the above said skeletonization algorithms has been compared using images of typical alphanumeric Hindi test patterns of "Kurushetra". The results of implementation of algorithms and their evaluation shows that the Alternative Parallel Skeletonization Algorithm provides us the better quality skeletonized pattern.

As shown in Fig. 12, the experiment shows that pattern obtained from Alternative Parallel Skeletonization Algorithm provides the best skeletonized pattern in terms of quality. This experiment is applied on 4 BMP images which contain different alphanumeric Hindi characters.

The aim of skeletonization is defined as the reduction of the number of pixels of an image or pattern that will preserve the structure of the image very well. As a result, after the transformation, lines that contribute to the letter are of a shape similar to the original and cross in a similar way.

As shown in Fig. 12, Alternative parallel skeletonization algorithm preserves the letter shapes. For numerous applications, it is especially important and essential to maintain the original connectivity of the script lines as well as the straight angles. Alternative parallel algorithm does an excellent job here. The algorithm preserves right angles at

ISSN: 2278 – 909X

the lines interconnections, which result in better correspondence between the original and the modified image.

A good skeletonization 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. The Alternative Parallel Skeletonization Algorithm produces a one-pixel-wide skeleton. The skeletons obtained from skeletonization process shows that Alternative Parallel Skeletonization Algorithm converges the pattern to unit width whereas the Parallel algorithm 1 and algorithm 2 do not provide the skeletons of unit width.

V. CONCLUSION

The discussion of various aspects such as convergence to unit width, connectivity and 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 parallel skeletonization algorithms are shown in Fig. 12. The differences in the outputs are very much clear from the visual inspection. We observe that the proposed alternative skeletonization algorithm provides the best connectivity for regional language numerals among the algorithms discussed previously. In future the information loss can be studied and skeleton and contour can be incorporated.

REFERENCES

[1] R. Gonzalez and R. E. Woods, "Digital Image Processing", Prentice Hall, 2002.

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

[3] T. Y., Zhang and C.Y. Suen et al., "A Fast Parallel Algorithm for Thinning Digital Patterns", Communications of the ACM, Vol. 27, No. 3, pp. 236-239, March 1984.

[4] W.H. Abdulla, A.O.M. Saleh and A.H. Morad et al., "A Preprocessing Algorithm for Hand-written Character Recognition", Pattern Recognition Letters, Vol. 7, No. 1, pp. 13-18, Jan 1988.

[5] A. Datta and S.K. Parui et al., "A Robust Parallel Thinning algorithm for Binary Images", Pattern Recognition, Vol.27, Issue 9, pp. 1181-1192, Sep. 1994.

[6] V.K. Sagar, C. Greening, W.Y. Tan and C.S.A. Leung et al., "Hardware/Software Co-Design of a Fingerprint Recognition System", IEEE Colloquium on Partitioning in Hardware-Software Codesigns Conference, Feb. 1995.

[7] N.H. Han, C. W. La and P. K. Rhee et al., "An Efficient Fully Parallel Thinning Algorithm", Proceedings of IEEE International Conference on Document Analysis and Recognition (ICDAR), pp. 137-141, Aug. 1997.

[8] M. Ahmad and R.K. Ward et al.," An Expert System for General Symbol Recognition," Pattern Recognition, Vol. 33, No. 12, pp. 1975-1988, Aug. 1999.

[9] M. Ahmed and R. Ward et al., "A Rotation Invariant Rule-Based Thinning Algorithm for Character Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 12, pp. 1672-1678, Dec. 2002.

[10] M.R. Girgis, A.A. Sewisy and R.F. Mansour et al., "Employing Generic Algorithms for Precise Fingerprint Matching based on Line Extraction", GVIP Journal, Vol. 7, Issue 1, pp. 51-59, April 2007.

[11] D. Miao, Q. Tang and W. Fu et al.,"Fingerprint Minutiae Extraction Based on Principal Curves", Pattern Recognition Letters, Vol. 28, Issue 16, pp. 2184-2189, July 2007.

[12] W. Abu-Ain, S.N.H.S. Abdullah, B. Bataineh, T. Abu-Ain and K. Omar et al., "Skeletonization Algorithm for Binary Images", 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013), Vol. 11, pp. 704-409, 2013.