Robust and Efficient Fully Parallel 2D Thinning Algorithm

Ananda Kumar Kinjarapu, Jagadeeswara Rao Enni, and Kamakshi Prasad Valurouthu
Department of Computer Science Engineering, JNTUH College of Engineering
Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh-500 085, India

ABSTRACT

Thinning is a well-known iterative layer-by-layer reduction operator to obtain skeletons of the binary objects. These skeletons are used as shape descriptors in many image processing, image analysis and pattern recognition applications. Thus obtaining topology preserved, centrally aligned and connected single pixel thin skeleton, without spurs and excessive erosions, and noise tolerant at pre-processing stage is essential to the success of later processing stages. The end-point based algorithms preserve original shape but results in extra spurs due to the presence of unwanted endpoints. The isthmus-based algorithms produce less spurs but causes excessive erosions. Hence in this paper, we proposed an efficient fully parallel thinning algorithm for 2D binary images. we proposed a general methodology for removing noise prior to thinning. A comparison with recent algorithms by the proposed method showed the better skeleton quality, efficiency and robustness.

Keywords:

Fully parallel, 2D thinning, Noise tolerant, Topology preservation, Geometry preservation, Weight values

1. INTRODUCTION

Skeleton which is also called as *medial axis* as first defined by Blum, is extensively used for *shape representation and analysis* in many pattern recognition applications such as printed and handwritten character recognition, handwriting recognition, signature verification and fraud detection, finger print classification, circuit board inspection, visual analysis of industrial parts, biomedical image analysis and content-based image retrieval. In such applications the object represented as a 2D binary image is a simple mapping from integer coordinates in 2D digital space $\mathbb{Z}^2$ to $\{0$ (white), 1 (black)$\}$. The black point clouds form the objects and the white point clouds form the background and cavities [1, 2, 3, 4]. One cannot obtain true skeletons in such a 2D digital space due to limitations of discretization. Hence, only approximation to the true skeleton can be obtained. Thus the methodology for extracting approximated skeleton should *preserve topology* (i.e. doesn't cause disconnections, doesn't remove entire object, doesn't create new cavities, doesn't merge a white component with a black or another white component), *preserve geometry* (i.e. aligned to medial axis and retain original shape), *robust* (i.e. tolerant to small perturbations near boundaries) and produce *unique* (i.e. no two different objects result in same skeleton), *symmetrical* (i.e. invariant under translation, rotation and scaling) *single pixel wide* (thin) lines without *unwanted spurs* and *excessive erosions*.

The various skeleton approximation methods proposed during the past four decades are broadly classified into two types: *non-iterative* and *iterative*. The non-iterative or distance-based algorithms calculate distance between every non-border pixel to the nearest border pixel, and flag the local extreme points as skeleton points. In iterative algorithms, skeletons are obtained by the application of thinning reduction operator iteratively layer-by-layer on the object. The non-iterative algorithms execute in linear order, preserve geometric properties but fail in topology preservation and vulnerable to spurious skeleton branches. The methods to prune these unwanted branches are very time consuming [5]. On the other hand, iterative thinning algorithms are simple, easy to implement and efficient in terms of execution time.

The reduction operator used in iterative thinning algorithms flip only those object points whose removal doesn't change the topology. The reduction operator referred to as parallel if multiple simple points flipped simultaneously and sequential if flipping of only one simple point at a time. Thus thinning algorithms referred to as *parallel* or *sequential* depending on the type of reduction operator used for topology preservation. Topology preservation is easier to guarantee in sequential algorithms but the parallel algorithms outperform sequential algorithms in the quality of the skeleton produced. Parallel algorithms produce mostly one pixel thin medial axis skeletons and represent shape sufficient for object recognition and classification by preserving endpoints. Thus they are referred to as *endpoint-based* algorithms. These algorithms have a severe drawback that they produce a spur corresponding to each unwanted endpoint. To reduce spurs in obtained skeletons, Bertrand and Couprie proposed *isthmus-based* thinning (curve-thinning) by accumulating non-simple points called as isthmuses (i.e. curve interior points) to preserve the geometrical information [1].

Ahmed and Ward [6] proposed a novel rule-based rotation invariant fully parallel thinning algorithm for characters. The authors proposed 20 rules to delete the points which are not part of two pixel wide either horizontal or vertical line i.e. which fit any of the rules [0, w, 1, 0], [0, 1, w, 0], $[0,\ w,\ 1,\ 0]^T$ or $[0,\ 1,\ w,\ 0]^T$. To preserve topology, authors retained points which are at the extreme of diagonal lines and different sets of points related to two pixel wide horizontal and vertical lines using 4x4 templates.

The skeletons extracted are one pixel boundary noise tolerant but diagonal lines are not thinned to single pixel.

Rockett [7] proposed an improvement by further thinning the provisional skeletons obtained using Ahmed and Ward [6]. The author constructed an adjacency matrix corresponding to the undirected graph of the local pixel connectivity over its 8-neighbors. The adjacency matrix elegantly traversed to identify the pixels to be removed and deleted immediately in single pass to obtain one pixel thin skeletons.

Han et al. [8] proposed a fully parallel thinning algorithm using only 3x3 masks grouped based on the count of black pixels (BP value). It extracts consistent skeletons in the presence of either one or two pixel wide boundary noise. The algorithm fails in obtaining single pixel wide skeleton. Actually, this algorithm examines 5x5 environment by using only 3x3 mask based computations.

Yet another fully parallel thinning algorithm proposed by G.S.Ng et al. [9]. This algorithm uses flag map to mark the pixels to be deleted and remove only those marked pixels which doesn't cause disconnections. It is noise sensitive, produces non-symmetric skeletons and not completely thinned to one pixel wide.

Couprie [10] stated that five out of fifteen parallel thinning algorithms evaluated are not topology preserved. Thus thinning algorithms preserving topology are referred as topological thinning algorithms. Initially Kong [11] proposed sufficient conditions for 2D and 3D thinning which are modified to combine them with parallel thinning approaches by Nemath, Kardosh and Palagyi [2].

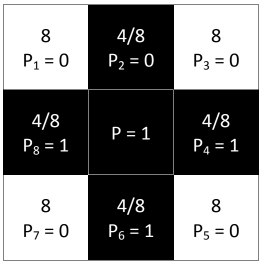
Nemath and Palagyi [3] proposed topology preserving algorithms based on sufficient conditions for topology and shape preservation. These algorithms produce spurs but are having less impact on later processing steps of the application. The skeletons produced are sensitive to noise near the boundaries.

As against to endpoint thinning algorithms, Nemath and Palagyi [1] proposed six parallel isthmus-based thinning algorithms. The fully parallel isthmus based thinning algorithm and the two 4-sub-iteration algorithms produce reasonable skeletons. The FP-Isthmus algorithm leads to excessive erosions and skeletons produced are different for multiples of right angles (asymmetric). This algorithm also uses a set of retain templates to preserve connectivity. The extracted skeletons are sensitive to boundary noise but produces less spurious branches.

Almost all fully parallel algorithms and parallel algorithms with sub-iterations failed in satisfying all the characteristics of skeletons despite of variety of techniques engineered to obtain the closure approximations to the true skeletons. These include: using variety of reduction operator ranging from 3x3 to 5x5 [1, 3, 4, 6, 8], efficient computations for 5x5 neighbor in terms of 3x3 neighbors [8], restore or retain templates to overcome the problem of discontinuities [1, 6], flag map techniques [9], an extra pass to thin parts of skeletons with two pixel wide [7], and noise elimination techniques [6, 8]. It is evident that no fully parallel algorithm exists which produces qualitative skeletons using only 3X3 mask based reduction operator without considering the techniques like flag map or retain templates, to preserve continuity.

In this paper we proposed a fully parallel thinning algorithm which is efficient and robust to noise perturbations. We proposed a method to remove noise as a pre-processing step. The obtained skeletons are based on the sufficient conditions for topology preserving parallel reductions [1, 2].

The rest of the paper is organized as follows. In section 2, the basic concepts and notations of 2D digital topology are reviewed. In section 3, we described the proposed fully parallel thinning algorithm and in section 4, the results obtained by proposed algorithm and comparision with both endpoint-based and isthmus-based algorithms are discussed. Finally, we present conclusions in section 5.



8 $P_1 = 0$	4/8 $P_2 = 0$	8 $P_3 = 0$
4/8 $P_8 = 1$	P = 1	4/8 $P_4 = 1$
8 $P_7 = 0$	4/8 $P_6 = 1$	8 $P_5 = 0$

Fig. 1. 3x3 environment in a 2D digital space (P_1, P_2, P_3, P_4, P_5, P_6, P_7 and P_8 used to indicate possible neighbors of a reference black point P. The cells labeled 4 are 4-adjacent and labeled with 8 are 8-adjacent to reference black point P. Gray cells indicate object points and white cells indicate background points.)

2. BASIC DEFINITIONS AND NOTATIONS

A 2D binary digital picture I = ($\mathbb{Z}^2$, 8, 4, B) is a quadruple [1, 2, 3, 4, 12]. Each of the element P$\in\mathbb{Z}^2$ is called as a picture point or pixel, B$\in\mathbb{Z}^2$ is the set of black (1) points and $\mathbb{Z}^2\setminus$B is the result of excluding the black points from the whole image referred to as white (0) points. The 8 and 4 refer to 8-adjacency of black points and 4-adjacency of white points respectively.

In digital space, *4-adjacent* points to a reference point are the four points exist above, below, to the left and to the right of the reference point. The four diagonal points along with the 4-adjacent points are *8-adjacent* to the reference point (see Fig. 1). The 4-adjacent points of the reference point P is called as *4-neighborhood*, N_4(P) = $\{P_2, P_4, P_6, P_8\}$. The 8-adjacent points of the reference point P is called as *8-neighborhood*, N_8(P) = $\{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$. Two points are said to be *4-connected* or *8-connected* if they are 4-adjacent or 8-adjacent respectively to each other. A *black component* is a maximal 8-connected set of points in B. Similarly a *white component* is a maximal 4-connected set of points in $\mathbb{Z}^2\setminus$B. A white component surrounded by a black component is referred to as a *cavity*.

The *environment* of a black point P is the set of values of points in 8-neighborhood of P. The *environment weight value of P*, ENV(P) is the decimal equivalent of the binary number of the point values (0/1) combined together in the sequence P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 (see Fig.1). The environment weight value of the central pixel P defined for Fig. 1 as ENV(P) = $(10101010)_2$ = 170.

In 2D digital space, there are three spatial domain operators defined in the 8-neighborhood of the central black pixel: *BP*, *AP*, and *CP*. The BP refers to the number of black pixels i.e. number of 1's in the environment of reference point. The AP refers to the number of 4-connected black components i.e. the number of either one-to-zero or zero-to-one bit transitions. The CP refers to the 8-connected black components count. The BP, AP and CP values corresponding to the environment in Fig.1 are 4, 4 and 1 respectively.

A black point P is a *border* point if it is 4-adjacent to at least one white point, otherwise it is an *interior* point. A *simple* point is a border point that contains exactly one black component in its 8-neighborhood (CP = 1). An *isthmus* point is a non-simple border point. A simple point is called as *endpoint* if its 8-neighbor contains either only one black point or exactly two 4-adjacent black pixels i.e. BP = 1 or (BP = 2 and AP = 1) [1, 3]. Processing of border points belonging to a single layer are referred to as *sub-iteration* or *sub-cycle*. Scanning of all border points layer-by-layer till no more points can be deleted is referred to as a *pass* or *cycle*. A pass may have one or more sub-iterations. Similarly an algorithm may be composed of one or more independent passes. An algorithm with only one pass and without any sub-iterations is termed as *fully parallel thinning algorithm* [1, 8].

3. PROPOSED FULLY PARALLEL THINNING ALGORITHM

In 2D digital space, object contours are described in terms of line segments. Thus the thinning reduction operator removes contour points iteratively layer-by-layer until the object is represented by one pixel thin line segments. The computation of directional change at any point requires its successor and predecessor points i.e. 1X3 neighborhood on the contour to be known. To ascertain the resultant pattern is of one pixel thin, 1X3 neighborhood examination of the contour is insufficient. For horizontal lines, the 1X3 neighborhoods above and below needs to be examined. Similarly, for vertical lines 1X3 neighborhoods to the left and right are required for ascertaining one pixel thin. This concludes that 3X3 neighborhood examination is sufficient for thinning. But for diagonal lines, one can easily notice that 5X5 neighborhood examination is required [6]. Computations for 5X5 neighborhood can be efficiently handled by 3X3 neighborhood computations [8]. In the following subsections, we derived conditions for horizontal, vertical and diagonal line segments along with 2X2 patterns to one pixel thin using efficient 3X3 neighborhood computations of 5X5 neighborhood.

Any thinning algorithm that preserves the endpoints and connectivity is considered as both topology and shape preserved. To preserve the connectivity, one can safely remove the candidate pixels whose AP = 1. The values of BP are ranging from 0 to 8. A point whose BP = 0 is an *isolated* point and cannot be deleted. If BP = 1 or BP = 2 with AP = 1 are endpoints and hence can't be deleted. The endpoints with BP = 1, AP = 2 and CP = 1 can be safely removed as they are redundant points. The interior points whose BP = 8 are skipped as the proposed method is based on a parallel reduction operator that erodes only the border points iteratively layer-by-layer. Out of eight templates correspond to BP = 7, four patterns are not border points and hence they can't be processed. The remaining four patterns can be deleted without causing discontinuity. But deletion of these patterns cause the skeletons deviated from the medial axis and hence those patterns with BP = 7 can not be deleted. Thus the proposed *pixel deletion criterions* are:

—BP $\geq$ 3 and BP $\leq$ 6 and AP = 1

—BP = 2, AP = 2 and CP = 1, which is equivalent to ENV(P) $\in$ $\{10, 40, 160, 130\}$

The above criterion do not preserve the topology in fully parallel algorithms as the two pixel wide horizontal and vertical lines along with 2x2 image patterns are completely deleted. Similarly two pixel wide diagonal lines are completely retained and are not thinned to one pixel. In the following subsections, we propose the additional conditions to overcome these problems.

3.1 Thinning 2X2 patterns to one pixel

A 2x2 pattern is one where all the pixels are both horizontally and vertically two pixel wide. In this pattern, each pixel has BP = 3 and all its neighbors also have BP = 3. Thus the maximum BP value of its neighbors = 3. If we add at least one pixel at any of the 12 possible adjacent cells of the 2x2 pattern, either the BP of the candidate pixel becomes 4 or one of its neighbor's BP becomes 4. If we remove any pixel from this pattern, then for all the three pixels BP = 2. It is evident from the above discussion that the 2x2 pattern can be uniquely identified with the condition: *candidate pixel as well as at least one neighbor pixel BP = 3*. All the four pixels in the 2x2 pattern satisfy the above criteria and hence we should retain any one among the four pixels to overcome the problem of complete removal. Thus the proposed condition to *retain the top left* pixel whose ENV(P) = 56 is:

X	X	X	X
X	1	1	Y
0	~~1~~	P = 1	0
X	1	1	Z
X	X	X	X

Fig. 2. Possible configuration of two pixels part of a two pixel wide vertical line (X, Y and Z may be either a black or white. Y and Z influence the BP value of the candidate pixel P)

—ENV(P) = 56 and max(neighbors BP) = 3

3.2 Thinning two pixel wide horizontal or vertical lines to one pixel

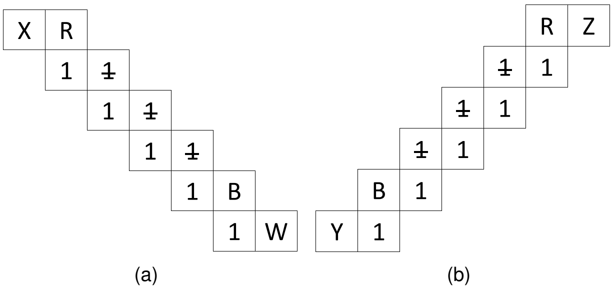
For vertical lines, pixels to the left are deleted and for horizontal lines, pixels to the bottom are deleted. All the corner pixels are deleted as they match with the deletion criteria and their deletion shorten the line but still the shape and topology are preserved. Thus two pixel wide vertical lines approach is equally applicable for horizontal lines as well. If two pixels are part of a vertical line of exactly two pixel wide, the pixel to the left of the left pixel and to the right of the right pixel should be white. Besides this, pixels above and below to these black pixels should be black as they represent either corner pixels or another set of pixels belonging to same vertical line. Thus a possible configuration of two pixels (the pixels labeled with strike through one and labeled with P = 1) part of a two pixel wide vertical line is shown in Fig. 2. The X, Y and Z may be either black or white where Y and Z reflect the BP value of the candidate pixel P and thus the environment weight value. Among the two pixels, one should retain one pixel (P) and delete the other (one strike through). It is easily observed that every neighbor of P contains minimum one zero and thus the maximum BP value is not more than 7. Hence P can be safely removed if it has at least one neighbor with BP = 8 and should be retained otherwise. By adopting similar procedure, for the horizontal two pixel wide lines, we can derive deletion criterion for upper pixel. Thus the possible BP and the corresponding environment weight values for *elimination of complete removal of two pixel wide horizontal or vertical lines* excluding BP = 7 are as shown in Table 1.

3.3 Thinning two pixel wide diagonal lines to one pixel

The criterion derived so far *retains* the two pixel wide diagonal lines as it is as shown in Fig. 3. To thin these diagonal lines the 1's with strike through (~~1~~) and the pixels labeled R and B as well as the pixels labelled W and Y if exist are to be deleted.

Table 1. Deletion criterion for two pixel wide vertical or horizontal lines

Y	Z	BP	ENV(P)		Condition
			V	H	
0	0	5	143	227	max(neighbor BPs) = 8
0	1	6	159	243	
1	0		207	231	



(a)

(b)

Fig. 3. Two pixel wide diagonal lines (a) Prime diagonal (b) Opposite diagonal (ones that are strike through, blue and red colored, and the pixels labeled W and Y if exists are to be deleted.

The 1's with strike through (~~1~~) are with BP = 4 and ENV(P) = 177 and 108. One can easily notice that the maximum neighbor BP of either the pixel labeled W or Y is equal to exactly 4. if W = 0 then ENV(B) = 161 and if W = 1 then ENV(B) = 177. Similarly, if Y = 0 then ENV(B) = 44 and if Y = 1 then ENV(B) = 108. the ENV(R) = 176 or 104 corresponding to X = 1 or Z = 1 respectively. if either Z = 0 or X = 0, the pixel labeled R becomes an endpoint and can be deleted only when its maximum neighbor BP = 4. Thus the *deletion criterions for thinning two pixel wide diagonal lines* are:

—ENV(P) $\in \{12, 48, 96, 193\}$ and max(neighbors BP) = 4

—ENV(P) $\in \{44, 104, 161, 176\}$ (BP = 3)

—ENV(P) $\in \{108, 177\}$ (BP = 4)

3.4 Criterion for elimination of noise of one or two pixels

Generally image noise can be classified as boundary noise, scratch noise and dithering noise. Boundary noise affects the smoothness of the boundary by ON/OFF the pixels on the boundary. Scratch noise resulted during the process of scanning the documents which are written using sketch pens or markers on white paper or board. These are the unwanted noise referred to as holes in black regions which affect the object topology. Dithering noise comes from binarizing the scanned images.

Skeleton extraction algorithm is said to be stable, if its skeleton does not deviate from center line even in the presence of noise. It is quite difficult to remove noise completely using the template based thinning techniques. The difficulty arises due to the inability in differentiating boundary noise to be deleted with endpoints to be retained.

The scratch noise can be removed using dilation and erosion morphological operators applied with a kernel size bigger than the scratch to be filled. The algorithm *fill_holes*() is used to remove scratch noise which is derived from K-fill filter [13] with K value equals to 3. The algorithm flips the white background pixels to black pixels until no more pixels to flip if the following criterions matches:

—CP = 1 and BP $\geq$ 6

—CP = 1 and BP = 5 and the number of corner pixels in the 8-neighbor = 2

Table 2. Noise elimination conditions proposed by Han et al. [8] and proposed modifications

BP	Han et al. [8] condition	Proposed modification
1	max(neighbor BPs) $\geq$ 3	max(neighbor BPs) $\geq$ 4
2	max(neighbor BPs) $\geq$ 3	max(neighbor BPs) $\geq$ 7

The *remove_noise*() algorithm is based on the assumption that the input image is of two or more pixels wide. It is used to remove boundary and dithering noise flip the black pixels to background that satisfy the following criterions is proposed until no more pixels to flip:

—BP = 1 and max(neighbors BP) $\geq$ 2

—BP = 2, AP = 1 and max(neighbors BP) $\geq$ 5

—ENV(P) $\in \{7, 28, 112, 193\}$

One pixel noise at the corner of the corner pixel can be removed using the first condition above. The noise pixel at the possible 4-neighbors of corner pixel can be removed using second condition above. The noise pixel at all other places can be removed using the above third condition. Any pixel successor of the first noise pixel can be deleted using the first condition.

Noise may be introduced during the process of thinning and should be removed [8]. The conditions proposed by Han et al. [8] to remove noise along with the proposed modifications are as shown in Table 2

The above modifications are proposed as the removal of the points excluded from deletable points of Han et al. [8] cause excessive erosions. This happens due to the removal of endpoints and complete removal of diagonal lines besides disconnections at junction points.

3.5 Reduction of redundant points

By applying recursively the criterion described so far left some redundant points with environment weight values 42, 138, 162 and 168 whose removal does not cause any disconnections. Simultaneous deletion of these redundant points with the pixel deletion criterion causes disconnections. This happens because both the point with environment weight value 42 and its right neighbor with environment weight value 193 are deletable points. Similarly for points with environment weight values 138, 162 and 168 whose respective top, left and bottom neighbors with environment weight values 112, 28 and 7 are also deletable points. Hence the following criterions proposed to remove these redundant points.

—ENV(P) = 42 and ENV(right neighbor) $\neq$ 193

—ENV(P) = 162 and ENV(left neighbor) $\neq$ 28

—ENV(P) = 138 and ENV(top neighbor) $\neq$ 112

—ENV(P) = 168 and ENV(bottom neighbor) $\neq$ 7

The criterions for thinning are summarized and the proposed algorithm named as AJK is described in the following subsection.

3.6 Proposed AJK Algorithm

The steps in fully parallel topological thinning **AJK** algorithm is described in Algorithm 1:

The FP-deletable points are summarized in Table 3.

```
Algorithm 1: AJK Algorithm
Input: image($\mathbb{Z}^2$, 8, 4, X)
Output: image($\mathbb{Z}^2$, 8, 4, Y)
X = fill_holes(X)
X = remove_noise(X)
Y = X
repeat
    B = {P / P is a border point in Y}
    D = {P / P $\in$ B and P is "FP-deletable" in Y}
    Y = Y\D
until $D = \phi$
```

Table 3. FP-Deletable Points

	Deletion Criteria	
BP	ENV(P)	Deletion Condition
1	-	max(neighbor BPs) $\geq$ 4
2	10,40,160,130	-
	12,48,96,129	max(neighbor BPs) = 4 or $\geq$ 7
	3,6,24,192	max(neighbor BPs) $\geq$ 7
3	7,14,28,112	-
	224,193,131	
	56	max(neighbor BPs) $\neq$ 3
	44,104,161,176	-
	42	ENV(right neighbor) $\neq$193
	162	ENV(left neighbor) $\neq$28
	138	ENV(top neighbor) $\neq$112
	168	ENV(bottom neighbor) $\neq$7
4	15,30,60,120	-
	240,225,195,135	
	108,177	
5	31,62,124	-
	248,241,199	
	143,227	max(neighbor BPs) = 8
6	63,126,252,249	-
	243,231,207,159	max(neighbor BPs) = 8

4. RESULTS AND COMPARISONS

In this section, the proposed algorithm was tested and its results are compared with [7] algorithm which is an improved version of [6] algorithm, the fully parallel FP-Isthmus based algorithm [1] and OIST algorithm [4]. All these algorithms tested on objects of different shapes but we present the results of only two images. The first test image is a 502X429 image of a duck with 71268 object points. The second test image of size 594X113 with 9183 object points and consists of words from Indic languages like Hindi, Telugu, Kannada, Tamil, Urdu, Oriya and English. The result of duck was shown as the skeleton superimposed on the original object. The pairs of numbers at each image in Figs. 4 and 5 are the count of skeleton points and the number shown in paranthesis is the number of parallel reduction operations required.

The proposed AJK algorithm produced more number of skeleton points compared to all other algorithms [1, 4, 7]. This is because of its uniqueness in preserving every important geometric feature of input object. The OIST algorithm produces symmetrical skeletons but not thinned to one pixel thin. we considered this as reference in comparing the results of the other algorithms. It is evident that the proposed algorithm compared to the other two algorithms extracts the skeletons which are closely matched with OIST but results produced are one pixel thin. The number of parallel reduction operators are same as OIST and is less than the other two algorithms. The AWPR algorithm suffers with excessive erosion and disconnections. The FP-Isthmus algorithm also suffers with excessive erosions and missing shape features.

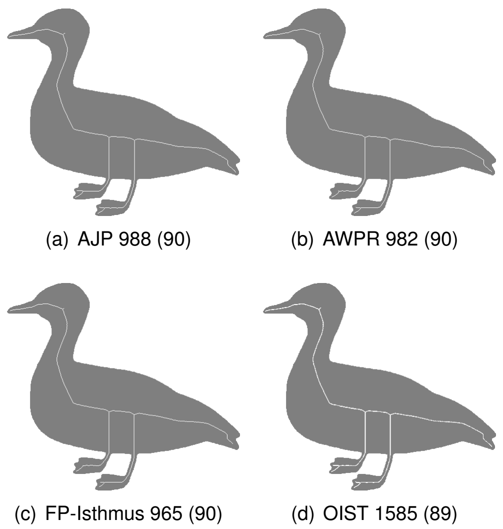

(a) AJP 988 (90)
(b) AWPR 982 (90)
(c) FP-Isthmus 965 (90)
(d) OIST 1585 (89)

Fig. 4. Result obtained for duck image of size 502x429 with 71268 object points

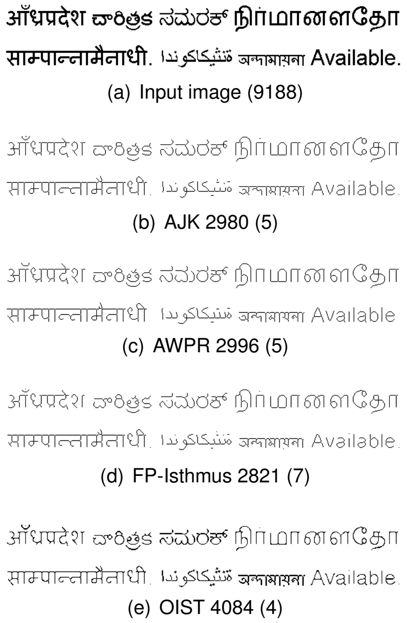

(a) Input image (9188)
(b) AJK 2980 (5)
(c) AWPR 2996 (5)
(d) FP-Isthmus 2821 (7)
(e) OIST 4084 (4)

Fig. 5. Result obtained for Indic script image of size 594x113 with 9183 object points

The ability to tackle noise is demonstrated in Fig. 6. The noise in the image is removed before applying the proposed thinning algorithm.

5. CONCLUSIONS

This paper presented a fully parallel topological thinning algorithm that is efficient and robust. The algorithm preserves topology

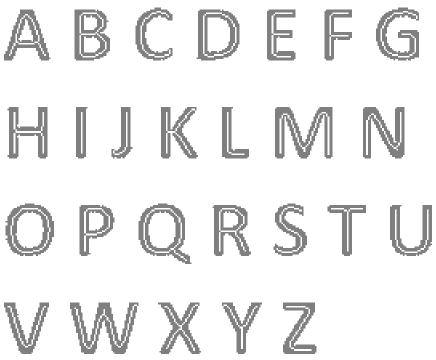


Fig. 6. Noise tolerance of the proposed algorithm illustrated using an image consists English alphabet with coarse boundary

without using techniques such as retain templates or flag maps. It also represented the geometric features sufficient for pattern recognition applications inspite of some redundant points at crossing points. This algorithm also produces skeletons without spurs for images with noises of two pixel wide.

6. REFERENCES

[1] Gabor Nemeth and Kalman Palagyi. 2d parallel thinning algorithms based on isthmus-preservation. In S. Loncaric, G. Ramponi, and D. Sersic, editors, *Proceedings of the International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 585–590, Dubrovnik, Croatia, September 2011. IEEE.

[2] Gabor Nemeth, Peter Kardos, and Kalman Palagyi. 2d parallel thinning and shrinking based on sufficient conditions for topology preservation. *Acta Cybernetica*, 20:125–144, 2011.

[3] Gabor Nemeth and Kalman Palagyi. Topology preserving parallel thinning algorithms. *International Journal of Imaging Systems and Technology*, 21:37–44, 2011.

[4] Peter Kardos and Kalman Palagyi. Isthmus-based order-independent sequential thinning. In M. Petrou, A.D. Sappa, and G.A. Triantafyllidis, editors, *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA 2012)*, pages 28–34, Crete, Greece, June 2012. IASTED.

[5] Wei Shen, Xiang Bai, Rong Hua, Hongyuan Wanga, and Longin Jan Latecki. Skeleton growing and pruning with bending potential ratio. *Pattern Recognition*, 44:196–209, 2011.

[6] Maher Ahmed and Rabab Kreidieh Ward. A rotation invariant rule-based thinning algorithm for character recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(12):1672–1678, 2002.

[7] Peter I Rockett. An improved rotation-invariant thinning algorithm. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 27(10):1671–1674, October 2005.

[8] N. H. Han, C. W. La, and P. K. Rhee. An efficient fully parallel thinning algorithm. In IEEE Computer Society, editor, *Proceedings of IEEE International Conference on Document Analysis and Recognition (ICDAR)*, pages 137–141, Ulm, Germany, August 1997. IEEE.

[9] G.S.Ng, R.W.Zhou, and C.Quek. A noval single pass thinning algorithm. *Pattern Recognition Letters*, pages 1267–1275, 1995.

[10] M Couprie. Note on fifteen 2d parallel thinning algorithms, 2009.

[11] T.Y Kong. On topology preservation in 2-d and 3-d thinning. *International Journal of Pattern Recognition and Artificial Intelligence*, 9:813–844, 1995.

[12] T.Y Kong and A. Rosenfeld. Digital topology: Introduction and survey. *IEEE Computer Vision, Graphics, and Image Processing*, 48:357–393, 1989.

[13] L. O'Gorman. Image and document processing techniques for the rightpages electronic library system. In *11th IAPR International Conference on Patern Recognition. Conference B: Pattern Recognition Methodology and Systems*, pages 260–263, The Hague, 1992.