International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

A Survey of Thinning Techniques on Two Dimensional Binary Images

Moumita Sarkar[1], Santanu Chatterjee[2]

[1]PG Student, Dept. of Information Technology, MaulanaAbulKalam Azad University of Technology, India

[2]Assistant Professor, Dept. of Computer Sc. And Technology, MaulanaAbulKalam Azad University of Technology

Abstract: ***An image may be considered to contain sub-images sometimes referred to as regions-of-interest, ROIs, or simply regions. This concept reflects the fact that images frequently contain collections of objects. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. For performing image processing operations, thinning is much more essential and highly adaptive technique. Skeleton is the end product of the thinning technique. Skeleton is important shape descriptor in object representation and recognition and able to captures essential topology and shape information of the object in a simple form. So, thinning is extremely useful in solving various problems such as recognition, matching and retrieval in different types of image analysis. The aim of this paper is to survey the various thinning algorithms with their procedures, performance and limitations.***

Keywords: Binary image, Thinning, Skeleton, Connectivity, End point, Medial axis transformation.

1. Introduction

Thinning is the process of transformation of digital image into a simplified but topological equivalent image. Thinning is shown inFig 1.

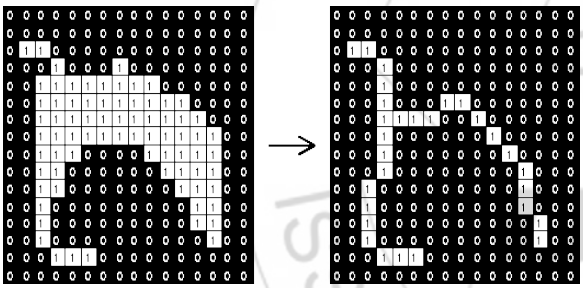

Figure 1: Image thinning

Classification of thinning algorithm

The classification of image thinning algorithms is shown in Fig 2.

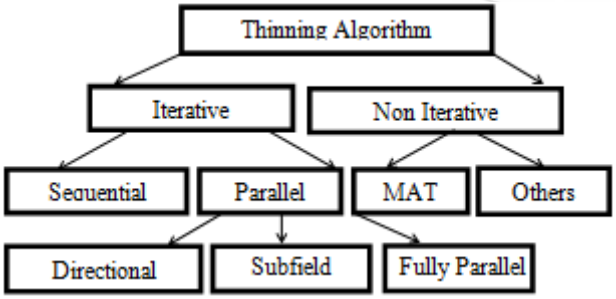

Figure 2: Classification of image thinning algorithms

1) **Iterative:** pixels are deleted iteratively with the same conditions in every iteration until suitable skeleton is found.
   a) **Sequential:** pixels are examined for deletion in a fixed sequence in every iteration, and the deletion of a point in the n^{th} iteration depends on all the operations that have been performed so far, i.e., on the result of the $(n-1)^{th}$ iteration as well as on the pixels already processed in the n^{th} iteration.
   b) **Parallel:** deletion of pixels in the nth iteration would depend only on the result that remains after $(n-1)^{th}$ iteration.
      - **Directional:** Each iteration is divided into sub iterations based on direction or combination of directions.
      - **Subfield:** This approach breaks down the picture into subfields based on some predefined criteria and pixels of same fields are removed in parallel.
      - **Fully parallel:** In this approach the same thinning operator (deletion criteria) is used in each iteration.
2) **Non Iterative:** Non iterative thinning algorithms do not scan individual pixels one by one. Instead, they produce a median line or some centerline of the pattern and then take a decision whether to delete that particular boundary pixel or not. Skeleton is obtained in step by step computation.
   - **Medial Axis Transformation (MAT):** The medial axis (MA) of a figure can be defined as the locus of centers of all the maximal disks, inside the figure but contained in no other disk.
   - **Others:** Shock graph and Reeb graph.

Need of thinning

- Reduces the amount of data to be processed; as a result time required for processing is reduced.
- Topology is preserved.
- One pixel wide skeletons produced are very useful for the purpose of pattern recognition.
- Shape analysis is made easy.

Requirements of thinning algorithm

The following algorithms are generally common for all thinning algorithms.

- **Connectivity preservation**: the output skeleton must have same connectivity as the original object.

- **No excessive Erosion**: should maintain the approximate end line location.
- **Consistency in topology**: the topology must remain consistent.
- **Centeredness**: the skeleton must be centered within the object boundary.
- **Medial line approximation**: The thinning results should approximate the medial lines.
- **Thinness**: the output skeleton should be as thin as possible
- **Boundary noise immunity**: Extraneous spurs caused by thinning should be minimized.

This survey is organized as follows: section 2 describes different thinning algorithms and section 3 contains a general conclusion.

2. Different approaches to Thinning

2.1 Iterative Thinning Algorithms

In section 2.1 iterative thinning algorithms are discussed where same deletion criteria are applied iteratively until the appropriate skeleton is produced.

2.1.1ZS Thinning Algorithm: Zhang and Suen in 1984 proposed very popular and well-proved thinning algorithm [1]. Henceforth, this algorithm will be referred as ZS algorithm. It is an iterative parallel thinning algorithm which works on 3×3 mask, considering P_1 and its 8 neighbor. P_1 is called the pivot point or the point of interest. Fig 3 shows the position of pivot pixel and its 8 neighbor in in the mask.

P_9	P_2	P_3
P_8	P_i	P_4
P_7	P_6	P_5

Figure 3: Pivot pixel and its 8 neighbor

This algorithm works on two sub-iterations.

Method

1st sub iteration: The criteria for deletion are
1) $2 \leq N(P_i) \leq 6$
2) $S(P_i) = 1$
3) $P_2 * P_4 * P_6 = 0$
4) $P_4 * P_6 * P_8 = 0$

2nd sub iteration: Conditions 3 and 4 in the first step are replaced with the following conditions.
1) $P_2 * P_4 * P_8 = 0$
2) $P_2 * P_6 * P_8 = 0$

$N(P_i)$ is the number of 01 pattern in the ordered set from P_2 to P_9 to P_2.
$S(P_i)$ is the number of non-zero neighbor of P_1.
Here, * denotes multiplication.

Performance
It is very simple and efficient algorithm but not able to produce proper thinned output for the 2x2 square and two pixel width slant lines (in both left to right and right to left direction). Fig 4 and Fig 5 shows the thinned output of 2x2 square and two pixel width slant line respectively.

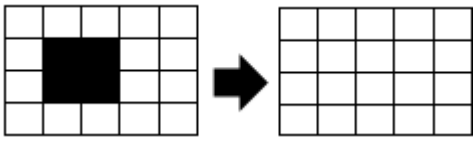

Figure 4: ZS thinning result on 2x2 square

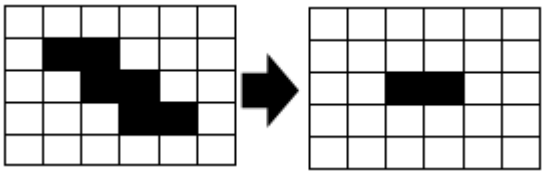

Figure 5: ZS thinning result on 2pixel width slant line

2.1.2LW Thinning Algorithm: Lu and Wang in 1985 proposed an algorithm [2] which is the single point modification of ZS algorithm. Henceforth, this algorithm will be referred as LW algorithm.This algorithm replaced the Condition 1 ($2 \leq N(P_i) \leq 6$) in ZS algorithm with the condition ($3 \leq N(P_i) \leq 6$). Other conditions remain same. The neighbors, $N(P_i)$ and $S(P_i)$ are as defined in ZS algorithm.

Performance
It keeps the two pixel width slant line intact, shown in Fig 6, which is erased by the ZS algorithm shown in Fig 5. Performance on other images is same as ZS algorithm.

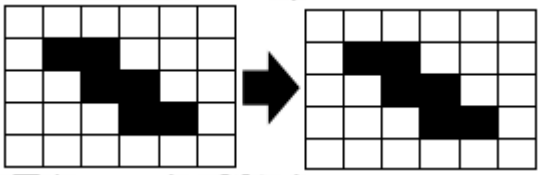

Figure 6: LW thinning result on 2pixel width slant line

2.1.3HSCP Thinning Algorithm: Holt, Stewart, Clint, and Perrott introduced a new thinning method [3] in 1987. Henceforth, this algorithm will be referred as HSCP algorithm. This algorithm reduced the sub-iterations from two to one compared to the ZS and LW algorithms. It is a fully parallel thinning algorithm, it first marks the pixels to be deleted and then unmark the pixels marked as deletable if certain conditions are satisfied.

3. Method

The criteria used to mark the pixel as deletable are:
1) $2 \leq N(P_i) \leq 6$
2) $S(P_i) = 1$

The neighbors, $N(P_i)$ and $S(P_i)$ are as defined in ZS algorithm.

The pixel that is marked as deletable should be retained if one of the following criteria is satisfied.
1) (P_4 is deletable) AND ($P_2 = 1$) AND ($P_6 = 1$)
2) (P_6 is deletable) AND ($P_4 = 1$) AND ($P_8 = 1$)
3) (P_4 is deletable) AND (P_5 is deletable) AND (P_6 is deletable)

Here, AND denotes logical AND.

Fig 7 shows the extended mask to check the pixel retain condition.

P_9	P_2	P_3	P_{10}
P_8	P_i	P_4	P_{11}
P_7	P_6	P_5	P_{12}
P_{16}	P_{15}	P_{14}	P_{13}

Figure 7: Extended 4x4 maskto elaborate criteria 1, 2 and 3 of the pixel retain condition

Performance: This algorithm produces skeletons almost same as that of ZS or LW algorithm but does not affect connectivity. The main drawback of the algorithm is that 45° or 135° inclined diagonal lines are removed and not able to produce one pixel thin horizontal and vertical lines. Fig 8 shows the thinning results of ZS and HSCP algorithms on a same sample image.

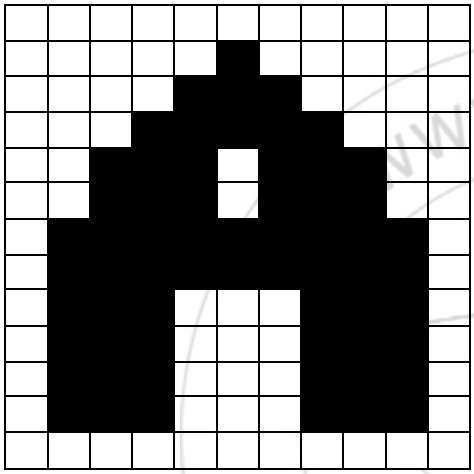

Figure 8 (a): Input image

Fig 8(a) shows the input binary image whereas Fig 8(b) and Fig 8(c) shows the thinned output of Fig 8(a) using ZS algorithm and HSCP algorithm respectively. In Fig 8(b) and 8(c) red cells denotes the pixels deleted in the first iteration and the blue cells indicates pixels deleted in the second iteration.

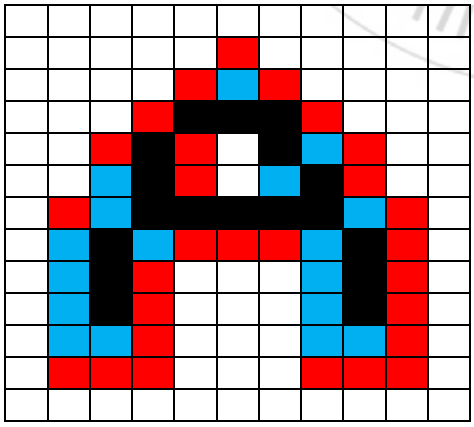

Figure 8(b): Thinned image using ZS algorithm

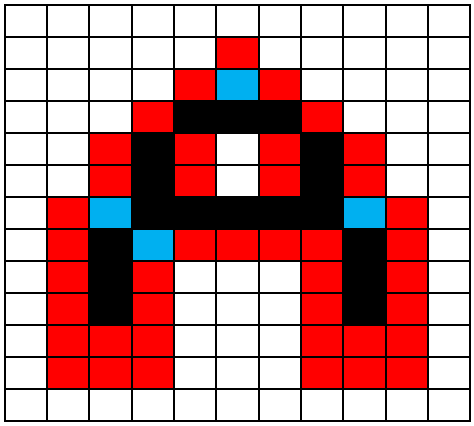

Figure 8(c): Thinned image using HSCP algorithm

2.1.4ZW Thinning Algorithm: Y.Y. Zhang and P.S.P. Whang proposed a single pass thinning algorithm [4] in 1988. Henceforth, this algorithm will be referred as ZW algorithm. Fig 9 shows the neighbors positions used in 4x4 mask.

P_{10}	P_{11}	P_{12}	P_{13}
P_9	P_2	P_3	P_{14}
P_8	P_i	P_4	P_{15}
P_7	P_6	P_5	P_{16}

Figure 9: Neighbors positions in 4x4 mask used in ZW algorithm

Method

The criteria for deletion are:
1) $P_i = 1$
2) $2 \leq N(P_i) \leq 6$
3) $S(P_i) = 1$
4) $P_2 * P_4 * P_8 = 0$ or $P_{11} = 1$
5) $P_2 * P_4 * P_6 = 0$ or $P_{15} = 1$

Performance: Remove completely the patterns which are removed by the ZS algorithm. Moreover two pixel width horizontal and vertical lines are also erased.

2.1.5 GH Thinning Algorithm: Zichan Guo and Richard W.Hall in 1989 proposed a two sub-iteration parallel thinning algorithm [5]. Henceforth, this algorithm will be referred as GH algorithm. In this algorithm, the authors used the same neighbor structure shown in ZS algorithm(Fig 3). Here, a new parameter NP is introduced. NP = min (NP_1, NP_2). Where NP_1 and NP_2 are defined as follows:
$NP_1 = (P_9 \mid P_2) + (P_3 \mid P_4) + (P_5 \mid P_6) + (P_7 \mid P_8)$
$NP_2 = (P_2 \mid P_3) + (P_4 \mid P_5) + (P_6 \mid P_7) + (P_8 \mid P_9)$

Another parameter C is defined as:
$C = (\sim P_2 * (P_3 \mid P_4)) + (\sim P_4 * (P_5 \mid P_6)) + (\sim P_6 * (P_7 \mid P_8)) + (\sim P_8 * (P_2 \mid P_9))$
Here, "~" denotes bitwise NOT and "|" denotes bitwise OR and "*" denotes multiplication.

Method
This algorithm deletes pixels if the following conditions are satisfied:
1. C = 1
2. NP = 2 or NP = 3

3. A) (P_2 | P_3 | ~P_5) * P_4 = 0 (for odd numbered iteration)
B) (P_6 | P_7 | ~P_9) * P_8 = 0 (for even numbered iteration)

Performance: Condition 1 preserves local connectivity. If NP = 1, the candidate pixel is an endpoint but the candidate pixel with NP = 2 may be an endpoint or a redundant point. Endpoint is points who have at most one nonzero neighbor and redundant points are those points whose deletion does not affect the connectivity or topological structure. Condition 2 illustrates this fact. Condition 3A) deletes the north-east pixels whereas 3B) removes south-west pixels. By deleting redundant pixels, this algorithm produces thinnerskeletons compared to ZS and LW algorithms. This algorithm is able to thin the two pixel width slant lines (both left-to-right and right-to-left) into one pixel width slant line, which was a major drawback of the previous algorithms. It produces medial axis points for horizontal, vertical and diagonal lines. It also produces the results in less number of iterations but does not preserve the branch points.

2.1.6 JC Thinning Algorithm: Ben. K. Jang and Roland. T. Chin in 1992 proposed a single pass parallel thinning algorithm [6].Henceforth, this algorithm will be referred as JC algorithm.

Method
This algorithm deletes the pivot pixel if it matches with one of the 20 deletion templates and retained if it matches with one of the retained templates again. Fig 10 shows the deletion templates.

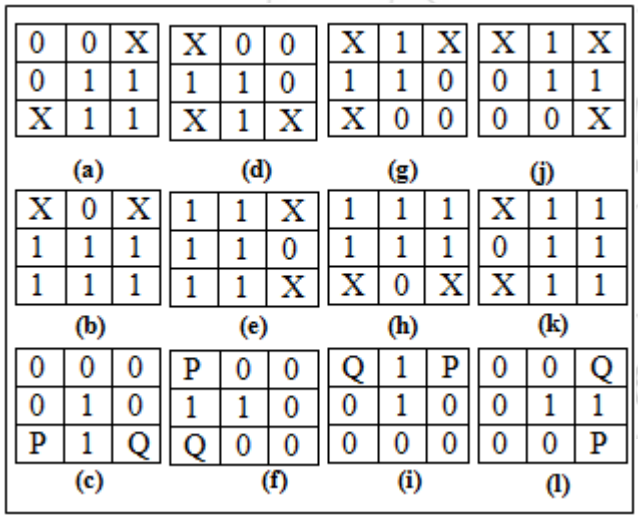

Figure 10 (i): 3x3 deletion mask used in JC algorithm

In Fig 10(i) and (ii), the variable P and Q are constraints by the logical operation (P OR Q) = 1.

Here, "X" denotes don't care condition.

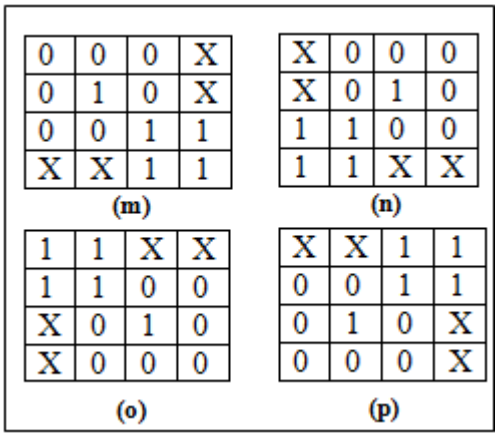

Figure 10(ii): 4x4 deletion mask used in JC algorithm

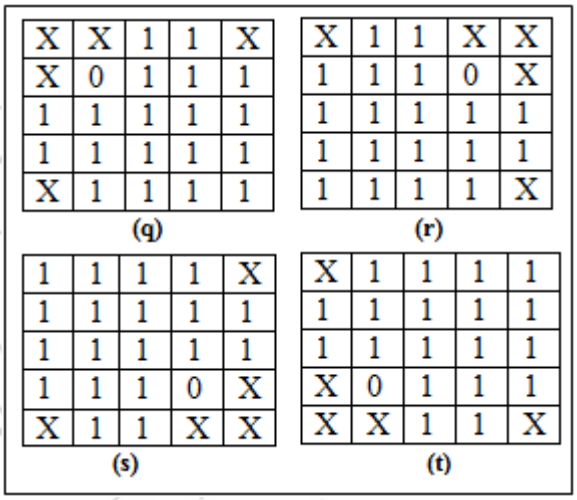

Figure 10 (iii): 5x5 deletion mask used in JC algorithm

Fig 11 shows 10 retain masks used in the JC algorithm. The first column shows the 3x4 masks, the second column shows the 4x3 masks and the third column shows the 4x4 masks.

Performance: This algorithm produce connected skeleton for the connected components. Two ones are connected if it is 8-connected and two zeros are connected if it is 4-connected. It is able to produce unit width skeleton.

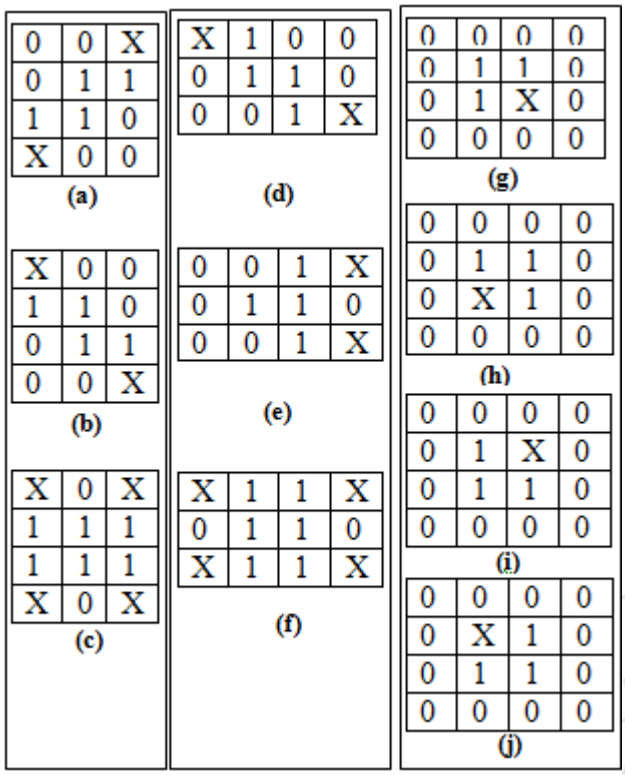

Figure 11: Retain mask used in JC algorithm

2.1.7 NJQ Thinning Algorithm: G.S. Ng, R. W. Zhou and C. Quek in 1994 proposed a single pass sequential algorithm [7]. Henceforth, this algorithm will be referred as NJQ algorithm.Here a filemap of same size as the size of the input image is used simultaneously to decide whether a particular pixel can be deleted or not. Smoothing templates are also used to smooth the final skeleton. Fig 12 shows the 3x3 mask to show the neighbors of pivot pixel and Fig 13 shows the 3x3 flag map to mark the deletable pixel. Initially all the values in the 3x3 flag map are set to 1. It is changed to 0 as soon as it is marked as deletable.

P_1	P_2	P_3
P_8	P_i	P_4
P_7	P_6	P_5

Figure 12: 3x3 mask with the 8 neighbors used in NJQ algorithm

Q_1	Q_2	Q_3
Q_8	Q_i	Q_4
Q_7	Q_6	Q_5

Figure 13: 3x3 flag map used in NJQ algorithm

Here, the authors have used three functions.
Previous neighbor function (PN)-
$PN(P_i) = \sum j = 1 \text{ to } 8 (P_j)$

This function is used to check whether the pivot pixel is a boundary point or not. If PN $(P_i) = 8$, then obviously it is not a boundary pixel. Current neighbor function (CN)-
$CN(P_i) = \sum j = 1 \text{ to } 8 (P_j \text{AND } Q_j)$

If one pixel in the neighborhood is flagged, that is marked as deletable, and then it can be exist no longer.

Transaction function (Trans)-
$Trans(P_i) = \sum j = 1 \text{ to } 8 \text{ Count } (P_j)$
Count $(P_j) = 1$, if $(P_j * Q_j) = 0$ AND
$(P_{j+1} * Q_{j+1}) = 1$
$= 0$, otherwise.

This function is used to measure the connectivity within the immediate neighborhood of the pixel. Like, if Trans $(P_i) = 1$ and CN $(P_i) > 1$, then the pixel can be deleted safely.

Here, "*" denotes multiplication and, AND denotes logical AND.

Method
Step 1: for each pixel in the image
Step 1.1: count PN(P), CN(P) and Trans(P).
Step 1.2: if P is a black boundary pixel that satisfies C_1 and (C_2 or C_3) then flag it.
C_1: $2 \le CN(P) \le 5$
C_2: Trans(P) = 1
C_3: P and its 8 neighbors match any of the smoothing templates shown in Fig 14.
Step 2: delete the flagged pixel.
Step 3: repeat Step 1 and Step 2 until no pixel can be deleted.

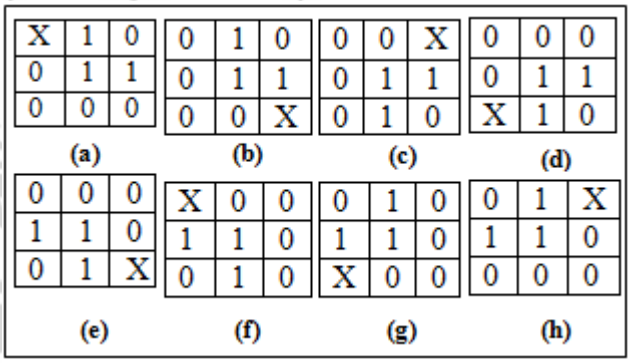

Figure 14: Smoothing template used in NJQ algorithm

In Fig 14, "X" denotes don't care condition.

Performance: This algorithm has high immunity to the boundary noise, able to maintain the thickness and connectivity.

2.1.8 Hilditch Thinning Algorithm:Hilditch's algorithm[8] consists of performing multiple passes on the pattern. It is a parallel-sequential algorithm. It is parallel because at one pass all pixels are checked at the same time and decisions are made whether to remove each of the checked pixels. It is sequential because this step just mentioned is repeated several times (until no more changes are done).

Method
At each pass the pixel will be deleted if it satisfies the following conditions:
1. $2 \le N(P_i) \le 6$
2. $S(P_i) = 1$
3. $P_2 * P_4 * P_8 = 0$ or $S(P_2) \ne 1$
4. $P_2 * P_4 * P_6 = 0$ or $S(p_4) \ne 1$

The neighbors, N (P_i) and S (P_i) are as defined in ZS algorithm.

Performance: This algorithm can produce connected one pixel thin images but two pixel wide slant lines and 2x2 squares will be completely erased by it.

2.1.9 PSH Thinning Algorithm:Se Hyun Park, Sang Kyoon Kim and Hang Joon Kim proposed a fully parallel thinning algorithm [9], in 1996, using weighted template.Henceforth, this algorithm will be referred as PSH algorithmThe 8 neighbors in the 3x3 mask is shown in Fig 15.This algorithm proposes a constant weight value at each location of the 3x3 mask as shown in Fig 16 and a sample mask is shown in Fig 17 to show the calculation of the connectivity value.

P_1	P_2	P_3
P_4	P_i	P_5
P_6	P_7	P_8

Figure 15: position of 8 neighbors in 3x3 mask in PSH algorithm.

256	257	263
259	P_i	287
271	319	383

Figure 16: 3x3 weighted templates used to calculate connectivity value

0	1	1
0	P_i	0
0	1	0

Figure 17: Sample 3x3 mask used to calculate connectivity value

In Fig 16, the connectivity value is (257 + 263 + 319) = 839.
A point is said to be connect point if any of the following criteria is satisfied:

1. $(P_1 \mid P_2 \mid P_3)$ AND $(P_6 \mid P_7 \mid P_8)$ AND $(\sim(P_4 \mid P_5)) = 1$
2. $(P_1 \mid P_4 \mid P_6)$ AND $(P_3 \mid P_5 \mid P_8)$ AND $(\sim(P_2 \mid P_7)) = 1$
3. $(P_2 * P_5 * P_6) \parallel (P_1 * P_5 * P_7) \parallel (P_3 * P_4 * P_7) \parallel (P_2 * P_4 * P_8) = 1$

Endpoint is a point who have at most one non-zero neighbor. An essential point is defined as one which includes a connect point and an end point.

Method

1. Calculate the connectivity value ofeach foreground pixel.
2. If the pixel is essential, preserve it, otherwise delete it.
3. Perform step 1 and 2 recursively until no changes takes place.

Performance: The algorithm is insensitive to noise of one pixel or two pixels at . It uses a combination of weighted template and pixel deletion rules as against to the Elimination for increase the execution speed of the algorithm.

2.1.10 HLR Thinning Algorithm:In 1997, another efficient fully parallel thinning algorithm was proposed by Another efficient fully parallel thinning algorithm was proposed by N H Han, C W La and P K Rhee [10]. Henceforth, this algorithm will be referred as HLR algorithm. It is based on blackpixel count in the 8-neighborhood of the 3x3 mask. The weight values of the 8 neighbors are shown in Fig 18. Pixels can be deleted if it satisfies the elimination rule shown in Fig 19, where there are seven groups of elimination conditions according to their weighted values. Groups are named as elimination condition group (ECG).

1	2	4
128	P_i	8
64	32	16

Figure 18: Weight value of 8 neighbors used in HLR algorithm

Method

1) Calculate the weighted value for each pixel.
2) For each boundary pixel, check the elimination conditions. If any condition is matched, delete that pixel.
3) Repeat step 1 and 2 until there is no pixel to satisfy the elimination condition.

Performance: The elimination condition is purely specified in terms of elimination templates (ECG 4) or purely specified as an elimination rule (ECG 1) or it is expressed as a combination of both elimination templates and elimination rules (ECG value other than 1 and 4).The obtained skeletons are shape preserved and insensitive to one or two pixel wide boundary noise. The algorithm does not compare all the possible templates as against to most of the other algorithms do. Instead it checks only the templates corresponding to the weight value of the candidate pixel. Thus, it is an efficient and faster approach compared to existing algorithms.

Elimination group	*Corresponding weight value*	*Elimination rule*
ECG1		Black neighbor is equal to or greater than three.
ECG2	3, 6, 10, 12, 24, 40, 48, 96, 129, 130, 160, 192	At least one black neighbor must be equal to or greater than three
ECG3	7, 14, 28, 44, 56, 104, 112, 131, 176, 193, 224	At least one black neighbor must be equal to or greater than seven
ECG4	15, 30, 60, 108, 120, 135, 177, 195, 225, 240	
ECG5	31, 62, 124, 144, 199, 227, 241, 248	At least one black neighbor must be eight
ECG6	63, 126, 159, 207, 231, 243, 249, 252	
ECG7	127, 223, 247, 253	At least two black neighbors must be eight

Figure 19: Elimination rule in HLR algorithm

2.1.11 KWK Thinning Algorithm:Kwon, Woong and Kang in 2001 developed a thinning algorithm [11].Henceforth, this algorithm will be referred as KWK algorithm. It is a two pass algorithm where the skeleton obtained by applying the criteria of pass-1 is further thinned against the pass-2 criteria to obtain the final skeleton. The pass-1 is in turn a two sub iteration process.

Method

Pass1

First sub iteration:
The criterions of deletion of pivot pixel are:

1. $2 \leq N(P_i) \leq 6$
2. $S(P_i) = 1$
3. $P_2 * P_4 * P_6 = 0$
4. $P_4 * P_6 * P_8 = 0$

Second sub iteration:

The criteria used in for deletion are:
1. $3 \leq N(P_i) \leq 6$
2. $S(P_i) = 1$
3. $P_2 * P_4 * P_8 = 0$
4. $P_2 * P_6 * P_8 = 0$

The neighbors, $N(P_i)$ and $S(P_i)$ are as defined in ZS algorithm, "*" denotes multiplication.

Pass2

The pass-2 is used to thin the 2-pixel wide skeletons further to 1-pixel wide and the redundant pixel is deleted if any one of the following criteria is matched:
1. $(P_1 = 1)$ AND $(P_8 = 1)$ AND $(P_6 = 1)$ AND $(P_3 = 0)$
2. $(P_3 = 1)$ AND $(P_4 = 1)$ AND $(P_6 = 1)$ AND $(P_1 = 0)$
3. $(P_5 = 1)$ AND $(P_6 = 1)$ AND $(P_8 = 1)$ AND $(P_3 = 0)$
4. $(P_4 = 1)$ AND $(P_6 = 1)$ AND $(P_7 = 1)$ AND $(P_1 = 0)$

Here, AND denotes logical AND.

Performance: This algorithm is able to maintain the connectivity and can produce single pixel thinned image with no excessive erosion no noise sensitivity and no diagonal line problem.

2.1.12 AW Thinning Algorithm: Ahmed and Ward, in 2002 proposed a rule based algorithm [12] which aims to produce rotation invariant skeletons. Henceforth, this algorithm will be referred as AW algorithm. A pixel is deleted if any one of the proposed 20 rules is matched. The 3x3 templates for deletion are shown in Fig 15.

In the fig15 the middle pixel is the pivot pixel or the pixel of interest. "X" denotes Don't Care. The value of those positions can be 0 or 1. Every pixel of the image will be checked with the 3x3 mask. If the pixel orientation of the mask matches with any of the shown 20 cases, then the pixel in the pivot position will be deleted i.e, will be changed to 0.

Performance: The main drawback of this algorithm is that it is constrained only for thinning digital characters. Besides this restriction, it produces skeletons of two-pixel width for some of the character patterns and it is sensitive to noise. But it can maintain the connectivity and there is no diagonal line problem.

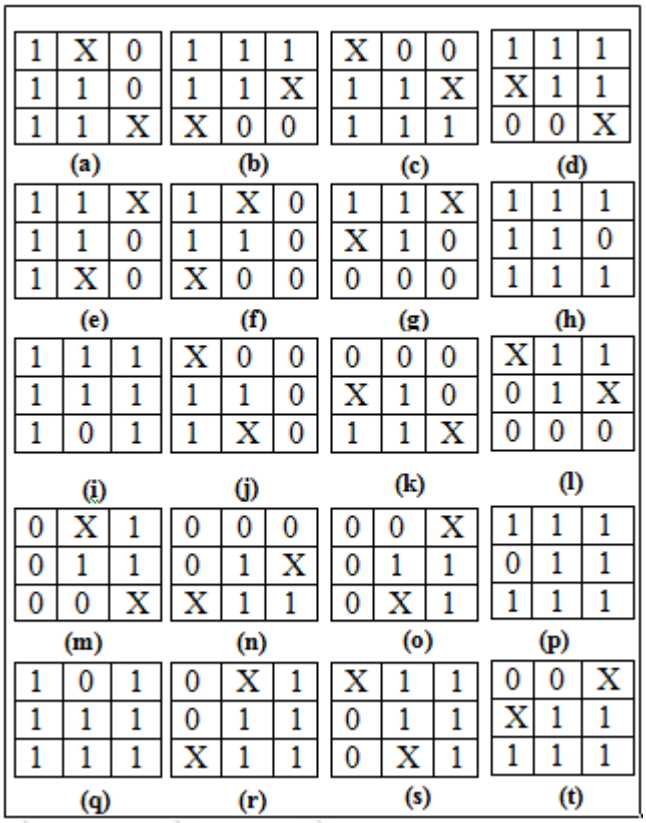

Figure 20: 20 rules for deletion in AW algorithm

2.1.13 HWL Thinning Algorithm: Lei Huang, Genxun Wan, Changping Liu proposed a parallel thinning algorithm [13] in 2003. It is again a rule based algorithm.

The elimination rules are shown in Fig 21, where group is done as per number of non-zero neighbors. In Fig 21, $N(P_i)$ is the number of non-zero neighbor of the pivot pixel P_i. The blank cells denote 0 values. When $N(P_i)$ is 0, 1 or 8, the pivot pixel will not be deleted.

Fig 22 shows the preserved templates for this algorithm.

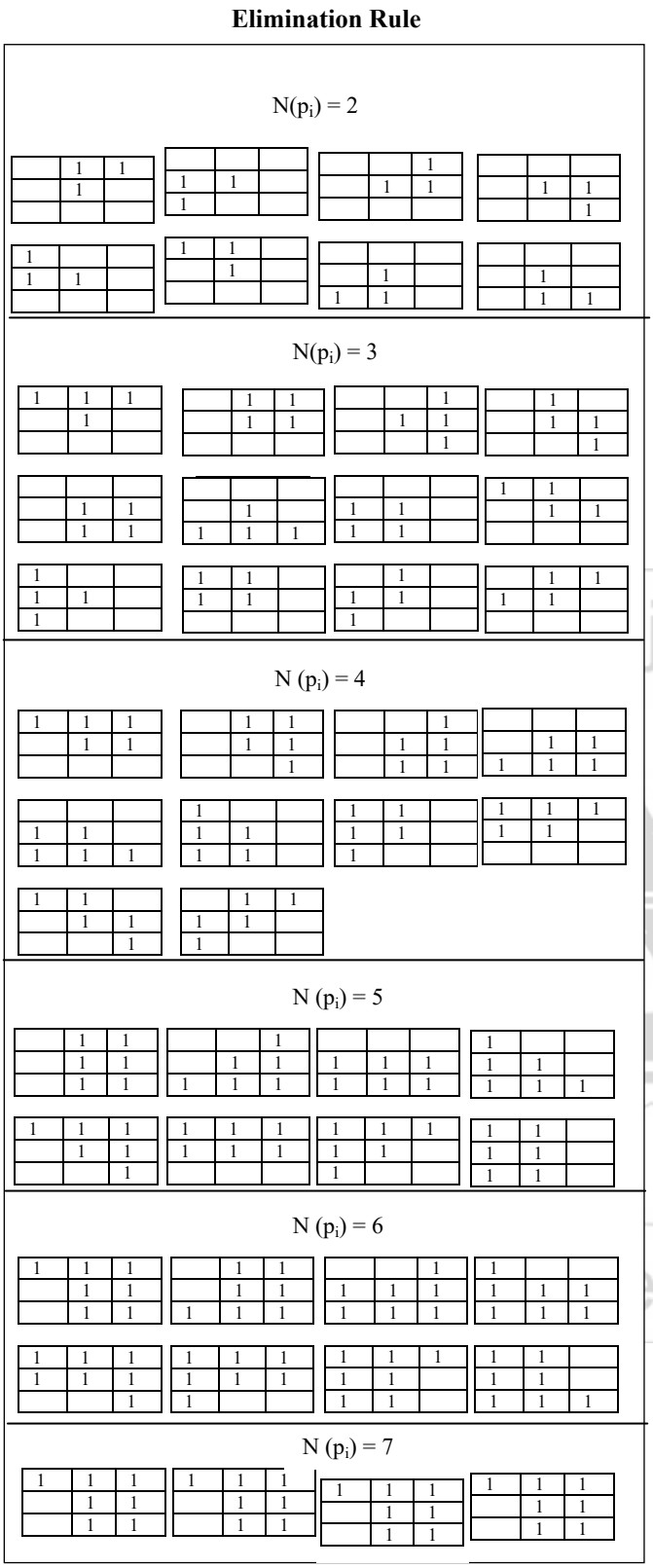

Figure 21: Deletion rules in HWL algorithm

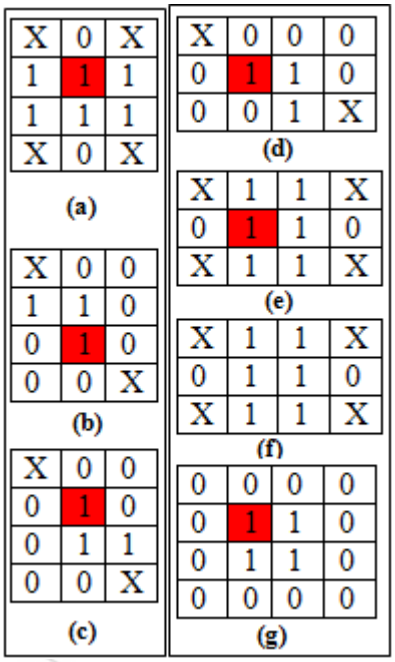

Figure 22: Preserved templates in HWL algorithm

In Fig 22, the red cells indicate the pivot pixels.

Method

1. Count the number of non-zero neighbors of the pivot pixel.
2. Check with the deletion rules. If the pixel is not deletable, go to step 4.
3. If the object is not two pixel width delete it, otherwise check with the preserved templates. If match found, keep that pixel else, delete it.
4. Repeat step 1 to 3 until no pixel can be eliminated.

Performance: the skeletons produced by this algorithm are very robust to contour noise and preserve the connectivity. There is no loss of information. This algorithm does not work well for two pixel width patterns.

2.1.14 Rockett Thinning Algorithm:Peter Rockett proposed a rotation invariant thinning algorithm [14] in 2005, which is an improvement to Ahmed and Ward algorithm (2.1.12).

Method

In the first pass, the skeleton obtained by using the AW algorithm which may contain parts of 2-pixel width lines. This algorithm constructs an undirected graphs including the candidate pixel in the 8-neighborhood. This undirected graph can be represented as an adjacency matrix. Consider the pixel configurations of Fig 23,their corresponding undirected graphs and adjacency matrices including the candidate pixels shown in Fig 24 and Fig 25 respectively.

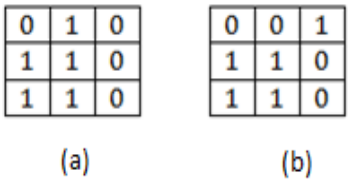

Figure 23: Two sample pixel configuration

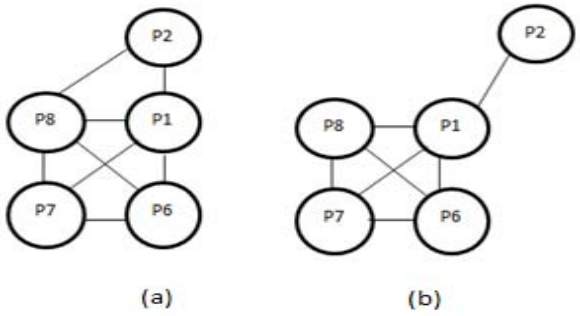

Figure 24: Graph of Fig 23

In Fig 24, the undirected graph shows 8-connectivity of all the foreground pixels in the 3x3 masks.

Sink Vertex \ Source Vertex	P2	P6	P7	P8
P2	0	0	0	1
P6	0	0	1	1
P7	0	1	0	1
P8	1	1	1	0

(a)

Sink Vertex \ Source Vertex	P2	P6	P7	P8
P2	0	0	0	0
P6	0	0	1	1
P7	0	1	0	1
P8	0	1	1	0

(b)

Figure 25: Adjacency matrices corresponding to Fig 24

From Fig 24(a) P_1 can be removed safely whereas from Fig 24(b) the removal of P_1 causes disconnections. By observing these two illustrations we can differentiate between these cases by checking whether every row of the adjacency matrix contains at least one nonzero entry or not. In Fig 25(b), there is one row with all zeros. If every row of the adjacency matrix contains at least one nonzero entry then the candidate pixel can be deleted safely. Otherwise the corresponding candidate pixel deletion causes discontinuity.

Performance: This algorithm can produce one pixel wide thinned image with no diagonal line problem as in ZS or LW algorithm. But this algorithm is sensitive to noise.

2.1.15XXCT Thinning Algorithm:Fei Xie, Guili Xu, Yuehua Cheng, Yupeng Tian proposed a Thinning algorithm [15] in 2009. Henceforth, this algorithm will be referred as XXCT algorithm. It is mainly for human body recognition. The pivot pixel and its 8 neighbors are shown in Fig 26.

P_4	P_3	P_2
P_5	P_0	P_1
P_6	P_7	P_8

Figure 26: Pivot pixel and its 8 neighbor position according to XXCT algorithm

Method

1. If the number of non-zero neighbor is 7, the point should be preserved. If it is less than 7, go to step 2.
2. Calculate the parameter NC.
   $NC = \sum_{K \varepsilon S} \{(\sim P_K) - ((\sim P_K)^* (\sim P_{K+1})^* (\sim P_{K+2}))\}$
   Here, S= {1, 3, 5, 7}; $\sim P_K$= 1-P_K and $P_9 = P_1$
3. If NC ≠ 1, keep the pixel.
4. Repeat step 1 to step 3 until no pixel can be eliminated.
   Here, "~" denotes NOT, and "*" denotes multiplication.

This algorithm can be explained using the following examples using Fig 27 which shows the sample 3x3 mask and Fig 28 which shows the values of each position in the mask shown in Fig 27 after applying XXCT algorithm.

1	1	1
0	1	0
1	1	1

Figure 27: Sample values for 8 neighbors in XXCT algorithm

The number of non-zero neighbor is 5, so we can go to the next step.
For K = 1,
$\{(\sim P_K) - ((\sim P_K)^* (\sim P_{K+1}) * (\sim P_{K+2}))\}$ is 1
For K = 3,
$\{(\sim P_K) - ((\sim P_K)^* (\sim P_{K+1}) * (\sim P_{K+2}))\}$ is 0
For K = 5,
$\{(\sim P_K) - ((\sim P_K)^* (\sim P_{K+1}) * (\sim P_{K+2}))\}$ is 1
For K= 7,
$\{(\sim P_K) - ((\sim P_K)^* (\sim P_{K+1}) * (\sim P_{K+2}))\}$ is 0

So, NC = 2.
The pivot pixel will not be deleted.

If it is deleted the structure would have been disconnected.

1	1	1
0	0	0
1	1	1

Figure 28: Values after applying XXCT algorithm on Fig 27

Performance: This algorithm is able to produce one pixel width skeletons fast.

2.1.16 PP Thinning Algorithm:Padole and Pokle introduce two simple thinning algorithms [16].
The first algorithm is based on Iterative border removal using only two operations: edge detection and subtraction. Henceforth, this algorithm will be referred as PP1 algorithm. Edges are detected by canny edge detection.

Canny edge detection algorithm has five phases.

- *Smoothing:* Elimination of noise. For smoothing Gaussian filter has used.
- *Finding Gradient:* Edges are marked where the gradient has larger magnitude. Smoothened image is then filtered (may be with a Sobel kernel) in both horizontal and vertical direction to get first derivative in horizontal direction (G_x) and vertical direction (G_y).

Edge Gradient (G) = sqrt (G^2_x+G^2_y)
Angle (θ) = $\tan^{-1}$ (Gy/Gx)

Gradient direction is always perpendicular to edges. It is rounded to one of four angles representing vertical, horizontal and two diagonal directions.

- *Non-maximum suppression:* After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge. For this, at every pixel, pixel is checked if it is a local maximum in its neighborhood in the direction of gradient.

- *Hysteresis thresholding to measure edge strength:* Two threshold values, upper and lower are selected.Pixels, whose values are above the upper threshold (strong pixels), are classified as edges. Fig 29 shows the thresholds.

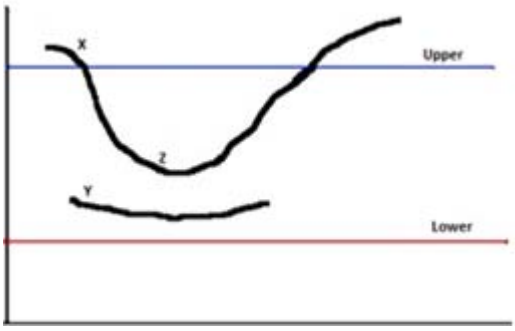

Figure 29: Threshold to determine strong and weak edge

- Edge linking: Finally the algorithm performs edge linking by incorporating the weak pixel that is 8-connected to the strong pixel. Those who lie between these two thresholds are classified edges or non-edges based on their connectivity. If they are connected to strong edge pixels, they are considered to be part of edges. Otherwise, they are also discarded.

The edge X in Fig 29 is the Upper, so considered as Strong Edge. Although edge Z is below Upper, it is connected to edge X, so that also considered as valid edge and we get that full curve. But edge Y, although it is above Lower and is in same region as that of edge Z, it is not connected to any strong edge, so it is discarded. It is very important to select upper and lower accordingly to get the correct result.

Method of PP1 algorithm

1) Detect the edge using canny's edge detection.
2) Remove the edge from the image.
3) Continue step 1 and 2 until a properly thinned image is formed.

The authors of PP algorithm also designed another iterative algorithm using successive erosion.Henceforth, this algorithm will be referred as PP2 algorithm.

Method of PP2 algorithm

1) Binarize the input image.
2) Build a black background around the image.
3) Check each pixel taking as pivot. If it satisfies the three criteria keep the pixel, otherwise erode it.
   - C1: check all the three consecutive pixels around the pivot pixel if any two of them are foreground pixel.
   - C2: check the connectivity by computing the number of non-zero neighbors.
   - C3: check if the pixels at the position P_2, P_4, P_6 and P_8 are well connected to the pivot pixel.
4) Repeat step 3 for each foreground pixel.

2.1.17 JK Thinning Algorithm: A. Jagna and V. Kamakshiprasad, proposed a parallel thinning algorithm for binary images [17] in 2010. Henceforth, this algorithm will be referred as JK algorithm. This thinning algorithm consists of several iterations, where many non-zero pixels will be deleted in each iteration and when no more pixels will be available to be deleted, the iteration will stop.

Method

1st sub iteration: the pixel Pi will be deleted if it satisfies the following conditions
1) $2 \leq N(P_i) \leq 6$
2) $S(P_i) = 1$
3) $P_2 * P_4 * P_6 = 0$
4) $P_4 * P_6 * P_8 = 0$

2nd sub iteration: the pixel Pi will be deleted if it satisfies the following conditions
1) $3 \leq N(P_i) \leq 6$
2) $S(P_i) = 1$
3) $P_2 * P_4 * P_6 = 0$
4) $P_4 * P_6 * P_8 = 0$

The neighbors, N (Pi) and S (Pi) are as defined in ZS algorithm and "*" denotes multiplication.

Performance: This algorithm is end point preservative and produces one pixel width thinned image.

2.1.18 JS Thinning Algorithm: Jun-Sik Kwon introduced a parallel thinning algorithm [17] to obtain unit width skeleton, basically for fingerprint recognition, considering the inner pixel in 2013. Henceforth, this algorithm will be referred as JS algorithm This algorithm iterate through three sub iterations and produce one pixel wide skeleton which is very close to medial axis.

Method

1st sub iteration: Takes off only the outer most boundary pixels using the inner points. If the object is rotated, the skeleton caused by thinning is also rotated.
1) Counter = 0
2) If the value of N (P_i) is between 1 and 7, and the value of S (P_i) is 1, then increment the counter by 1.
3) Probe into inner point.
4) If Counter ≠ 0, go to step 1, otherwise go to 2nd sub iteration.

2nd sub iteration: find out the two pixel wide skeleton.
1) If the value of N (P_i) is between 2 and 7, and the value of S (P_i) is 1, then increment the counter by 1.
2) Apply exception masks and removal decision masks.
3) If Counter ≠ 0, make Counter = 0 and go to step 1, otherwise, go to 3rd sub iteration.

3rd sub iteration: prunes the needless pixels and yields the thinnest possible skeleton.
1) Apply sub deletion condition mask sequentially to the object.
2) Increment Counter by 1.
3) If Counter ≠ 0, make Counter = 0 and go to step 1, otherwise get the final skeleton.

The neighbors, N (Pi) and S (Pi) are as defined in ZS algorithm.

In the 1st sub iteration if pivot pixel has more than one inner pixels and it is located at the boundary, then it is a pure boundary pixel and would be deleted. The author of JS algorithm proposed different exception masks deletion masks to take care of horizontal and vertical bars as well as two pixel width squares. Also, four sub deletion conditions are defined.

- $(P_2 * P_4) = 1$ AND $P_7 = 0$
- $(P_4 * P_6) = 1$ AND $P_1 = 0$
- $(P_6 * P_8) = 1$ AND $P_3 = 0$
- $(P_2 * P_8) = 1$ AND $P_5 = 0$

Here, "*" denotes multiplication.

Performance: It preserves the connectivity, produce unit width skeleton with no redundant pixel.

2.1.19 Jagna's Thinning Algorithm:A. Jagna in 2014 introduced a thinning algorithm [19]. This algorithm performs thinning in two pass. Henceforth, this algorithm will be referred as Jagna's algorithm. It is a two iteration algorithm. In the first iteration, the entire image is thinned into two pixel width and in the second iteration; it is further thinned into one pixel wide without any discontinuities in the final image. The 3x3 mask used in this algorithm is the same mask used in ZS algorithm.

Method

1st pass

1) Count the number of non-zero neighbors and the number of one-to-zero or zero-to-one transition in the ordered manner (P_2, P_3, P_4, P_5, P_6, P_7, P_8, and P_9).
2) If the number of non-zero neighbor are in the ranges from two to six, and also either number of the zero-to-one or one-to-zero transition is exactly one, then the candidate pixel is will be deleted.
3) Iterate step 1 and 2, until no more changes takes place in pass 1, otherwise, go to pass 2.

2nd pass

1) Count the number of non-zero neighbor of the candidate pixel pi. If it is in the range of three to five, then delete the pixel under consideration.
2) Repeat step 1, until no more changes takes place.

Fig 30 shows four sample cases where candidate pixel would not be considered in 1st iteration.

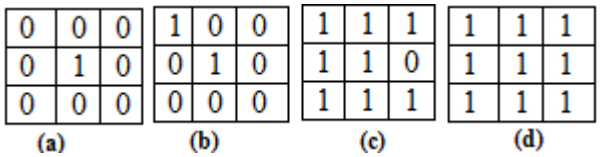

Figure 30: Sample cases where pivot pixels would not be considered in 1st iteration of Jagna's algorithm

In Fig 30(a), the number of non-zero neighbor is zero, so pivot pixel would not be considered for further processing.

In Fig 30(b), the number of zero-to-one or one-to-zero transition in the ordered manner is exactly one, but number of non-zero neighbor is one. So, the pivot pixel would not be considered for further processing.

In Fig 30(c), the number of zero-to-one or one-to-zero transition in the ordered manner is exactly one, but number of non-zero neighbor is seven. So, the pivot pixel would not be considered for further processing.

In Fig 30(d),the number of zero-to-one or one-to-zero transition in the ordered manner is zero. So, the pivot pixel would not be considered for further processing.

When any two to five consecutive neighbors of the pivot pixel would be 1, then it would be considered for 1st iteration.

Performance: This algorithm preserves the topology, connectivity and end points of the original image in the thinned image.

2.1.20 BST Thinning Algorithm:Lynda Ben Boudaoud, Abderrahmane Sider and Abdelkamel Tari proposed a Hybrid Thinning algorithm [20], a combination of directional and sub field approach. Henceforth, this algorithm will be referred as BST algorithm. This algorithm divides the image into two sub fields as per the parity of pixels. Fig 31 shows the coordinate position of the pixels.

i-1, j-1	i-1, j	i-1, j+1
i, j-1	i,j	i, j+1
i+1, j-1	i+1, j	i+1, j+1

Figure 31: pixel coordinate position in 3x3 mask

In this algorithm, the same 3x3 mask is used as in ZS algorithm.

1st sub field: The set of odd pixels, whose sum of the coordinate value (i,j) is odd.

2nd sub field: The set of all those pixels whose sum of the coordinate value (i,j) is even.

Method

1st sub iteration: the conditions for deletion are:

1. $(i + j)$ MOD 2 = 0
2. $\sim P_2 * (P_3 | P_4) + \sim P_4 * (P_5 | P_6) + \sim P_6 * (P_7 | P_8) + \sim P_8 * (P_9 | P_2) = 1$
3. Number of Non Zero neighbor in the ordered list is in between 2 to 7.
4. $P_2 * P_4 * P_6 = 0$
5. $P_4 * P_6 * P_8 = 0$

2nd sub iteration: the conditions for deletion are:

1. $(i + j)$ MOD $2 \neq 0$
2. $\sim P_2 * (P_3 | P_4) + \sim P_4 * (P_5 | P_6) + \sim P_6 * (P_7 | P_8) + \sim P_8 * (P_9 | P_2) = 1$
3. Number of Non Zero neighbor of P_1 in the ordered list is in between 2 to 7.
4. $P_2 * P_4 * P_8 = 0$
5. $P_2 * P_6 * P_8 = 0$

Here, "~" denotes NOT, "|" denotes bitwise OR and "*" denotes multiplication.

Performance: This algorithm preserves the topology, connectivity and end points of the original image in the thinned image and can produce one pixel thin images. But it

has staircase effects in the output image and produces some extra pixels also.

2.2 Non-iterative thinning algorithms

Non iterative thinning algorithms do not scan individual pixels one by one.

Medial Axis transformation: The MAT is defined as: given an object represented, say by a polygon Q, the medial axis is the set of medial points internal to Q such that there exists at least two points on the object's boundary that Are equidistant from the medial point and are closest to the medial point Associated with the medial axis Is a radius function, which defines for each point on the axis its distance to the boundary of the object.

The medial axis (MA) of a figure can be defined as the locus of centers of all the maximal disks, inside the figure but contained in no other disk. The medial axis transformation consists of the MA and the Radius function. The MA of a figure is also called the skeleton or the symmetric axis of the figure. With The MA and the radius function one can reconstruct the figure by taking the union of all circles centered on the points comprising the MA, each with a radius given by the radius function. Fig 32 demonstrates the Medial Axis. A medial point can be classified into one of three types (i.e. end point, normal point, and branch point) depending on the order of the point. End points are of order one, normal points of order two, and branch Points of order three or more, corresponding to the number of neighboring medial point(s).

Medial representations of shapes are useful due to their use of an object centered coordinate system that directly captures the shape such as thickness, bending, and elongation of an image.

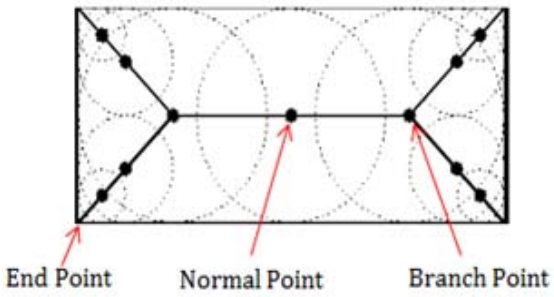

Figure 32: Medial axis line by joining the center of maximal disk

Medial axis has some interesting properties.

- For a given shape, there is a unique medial axis [21].
- Medial axis is topologically equivalent to its shape [22].
- Any shapes can be reconstructed from its medial axis [23].
- The dimension of medial axis is always lower than its shape [24].

2.2.1 Blum Medial axis: The medial axis is first introduced by Blum [24].Blum's medial axis can be defined in a few different but equivalent ways:

- *Grassfire Analogy*: Suppose fire started at the boundary that spreads with uniform speed and burns everything in its path. As the fire spreads into the middle, each part eventually meets other parts of the fire started at other parts of the boundary. When these two wave fronts meet, they quench each other because there is nothing left for either to burn. Since the fire spreads at uniform speed, these quench points must be equidistant from two different parts of the boundary. The locus of these quench points is the medial axis.
- *Maximal Discs:*(discussed inreference of Fig 32)
- *Ridges in a Distance Transform:*As we progress from away from a particular point on the boundary, wefind increasing values in the distance map until we reach a point as which we are now closer to some other part of theboundary than the place that we left. So, the distance increases as you progress inwards until it forms a ridge, a locus of points with values higher than those just off the ridge on either side. The locus of these ridges in thedistance transform is the medial axis.

2.2.2Medial axis from path based distance transformation: Gabriella Sanniti Di Baja and Edouard Thiel in 1996 proposed a skeletonization algorithm constructed on path-based distance function [25]. First, the distance map is computed. Every foreground pixel receives distance information from the foreground pixels closer to the boundary.

Second, the extraction of medial line has done as the pixel identified as the carter of maximal disk followed by a path linking procedure.

Third, the loops which do not surround the hole of foreground pixel are filled in to create the topologically correct skeleton. Next, the extracted medial line is reduced to unit width through several topology preserving removal operations.

Pruning is performed to remove the unnecessary branches.

Beautification is done to remove the zigzag effects where the actual skeleton pixel has to be removed by its neighbor.

The time of computation of this algorithm depends on the number of background as well as background pixels and also on the adopted distance function. Cost of computation is modest. Pruning and beautification allows molding the skeleton structure as per user's requirement.

2.2.3 Constructive medial axis generation: Chuhua Xian, Yusheng Li, Shuming Gao in 2007 proposed an method which constructively generate medialaxis with two segments and a Boolean operation [26].

Initially two segments have their corresponding medial axes. The Boolean operation (union/ subtraction) is performed which breaks the effected medial axis.

The vanishing boundary is identified which is to be disappeared at the effect of the Boolean operation. Next the regeneration region is to be identified i.e, the medial axis segments of the two segments that need to be combined to generate the correct medial axis of the whole model. New

medial axis is constructed with all medial axis segment based on every two elements on regeneration region.

2.2.4 Medial axis based thinning: Soumen Bag and Gaurav Harit in 2010 proposed a thinning algorithm based on Medial axis mainly for character images [27]. First the image is binarized using Otsu's algorithm and contour extraction is done.

For all pixelsC_i, in the ordered set of contour pixel $\{C_1 \ldots C_n\}$, on one of the contour lines, another pixel $C_j$is identified on the other contour line on the same stroke, for which the distance $|C_iC_j|$ is minimum.i is the ordered set of all odd numbers and j is the ordered set of all even numbers The minimum distance is verified using the five forward and five backward neighborhoods. Fig 33 shows the nearest opposite pixel pair.

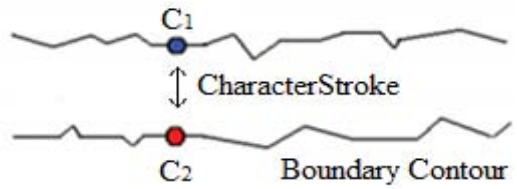

Figure 33: Closest point on opposite contour

If the pixel pair$<C_iC_j>$ satisfies all the three conditions (i, ii and iii) then mark the mid-point of the line joining C_i and C_j as a pixel on the medial axis.

1) All the pixels in-between the pixel pair $<C_iC_j>$ should be foreground pixels.
2) $|C_iC_j|$ must be $\leq\xi$(value is set to 0.45 times of the average height of the character image).
3) The median of the local orientations in the neighborhood of C_i and C_j should be similar.

Skeleton segments are now detected using by 8 connected component analysis. For each extreme point (start and end points of a skeleton segment) proximal extreme points belonging to other skeleton segments are found and the proximity set is built.

In a given proximity set, if any two skeleton segments have the orientation differ by ±10 degrees then the extreme point of either of these segments is selected and extend it to meet the extreme point of the other one in this set. If there is a skeleton segment whose orientation does not match with any other segment in the same proximity set, then its extreme point is extended till it meets the skeleton extrapolated from other segments.

2.2.5 Skeleton point detection by slope change: M. Narayana, Sandeep V.M and Subhash Kulkarni in 2011 introduced a simple, robust, fast and efficient content based image retrieval system [28] where they detect the skeleton points through slope change.

The boundary is detected using Chan-Vase level set model, and then the distance map level is set using this boundary information. Skeleton points are extracted using distance mapped function by scanning the distance map from row wise and column wise through backward and forward distance. A change in slope of the distance map denotes the skeleton point. With this property, skeleton points can be extracted easily by searching the slope deviation.

With backward distance, a point(x,y) can be a skeleton point if one of the following conditions are true

- D(x,y-1)-D(x,y) ≠D(x,y)-D(x,y+1)
- D(x-1,y)-D(x,y) ≠D(x,y)-D(x+1,y)

1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	1
1	2	3	3	3	3	3	2	1
1	2	3	4	4	4	3	2	1
1	2	3	4	5	4	3	2	1
1	2	3	4	4	4	3	2	1
1	2	3	3	3	3	3	2	1
1	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1

Figure 34: Detection of skeleton point through slope change

Highlighted points of Fig 34indicate the skeleton points.

2.2.6 Extended Grass Fire Transform: Medial axis sometimes contains branches that do not contain relevant features. To get rid of this problem in 2011 Lu Liu, Erin W. Chambers, David Letscher, and Tao Ju proposed extended distance function (EDF) and extended medial axis (EMA) [29]. It is based on Blum's grassfire methodology.

EDF measures the radius of the longest tube centered at a medial axis point that fits in the shape. EMA is the center where these longest tubes are confined at both the ends.

In the extended grassfire analogy, the fire is exploded at the ends of medial axis and propagates geodetically at the medial axis at uniform speed. The EDF at the medial axis point is the burning time of the extended grassfire where the quench site of the fire fronts is the EMA if the shape is connected. If the shape contains interior holes, then EMA is the remaining unburned portion of the medial axis.

Suppose, "M" is the weighted graph that captures a piece wise approximation of the medial axis. The distance of the shape boundary is given at each degree1 node of "M".

- First, the extended grassfire is computed.
- Then, the degree1 node i, with the smallest distance and smallest extended grassfireR_i is removed.
  - If the removal exposes new degree1 node j, the extended grassfire of the new node will be (R_i + weight of arc {i,j}).
- Stop the process when the graph is reduced either a single node or a set of cycle which is EMA of "M".

EMA preserves the homotopy of the medial axis unique for connected 2D shape. EDF offers stable global measures of the shape elongations and EMA is a stable choice of shape center for simple connected shapes.

2.2.7 Skeleton based on MAT and symmetry information: Yung-sheng chen and Ming-te chao in 2012 proposed a method for skeleton based on MAT and Symmetry information [30].
This algorithm works on four phases

- *Extraction of Medial Axis with Symmetry Information:* MAT points are found using Euclidian distance. To maintain the symmetry property each foreground point P on MAT is further inspected. P will satisfy the symmetry property if there exist two background point b_1 and b_2 satisfying condition I and II.

1) $|d(P,b_1)-d(P,b_2)|\leq 1$
2) Angle between b_1Pb_2 is equal to θ_1 and the Euclidian distance between P and set of the background points is greater than or equal to δ_1.

Or

Angle between b_1Pb_2 is greater than or equal to θ_2 and the Euclidian distance between P and set of the background points is greater than or equal to δ_2. θ_1, θ_2, δ_1 and δ_2 are set to some predefined value.

Fig 35 shows the initial image and Fig 36 shows image after this phase.

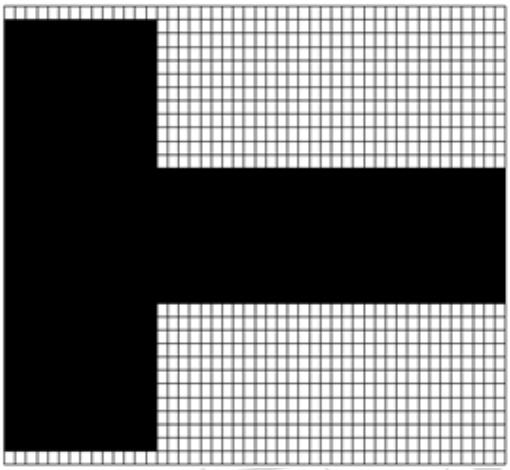

Figure 35: Initial image

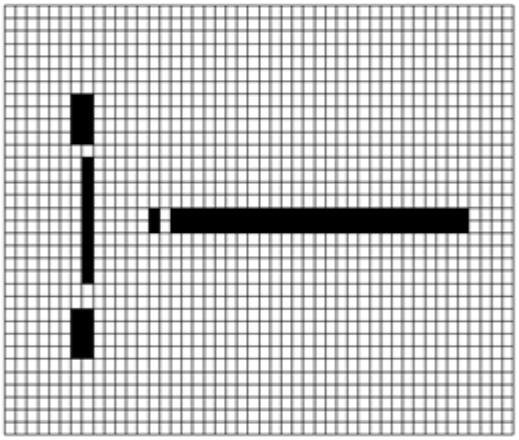

Figure 36: Image after phase 1

- *Clustering:* pixels yields by phase1 are distributed into different places, and they are disjointed. So, before linking, clustering is done. Fig 37 shows clusters.

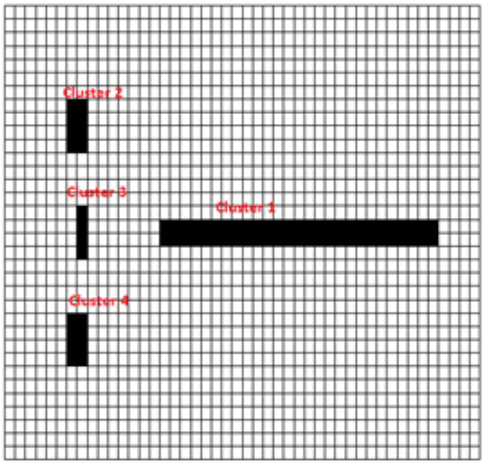

Figure 37: Clusters

- *Linking:* To link the clusters there should be a unique link between two clusters and for this multi path searching is done. The authors used the following rules for multipath searching.

Let ,
N^1: total number of link members.
L_m: link members (1… N^1).
Eds: Euclidean distance summation
$\sum m=1$ to n^1,ED(L_m, B), where ED is the Euclidian distance and B is the set of background points.
Slds: straight line distance summation between link members
$\sum m=1$ to n^1-1,ED(L_m, L_{m+1})
SC: Step count. = N^1.
Rule 1: The link lies on the pixels having local maximum of EDs.
Rule 2: No links exist between two pixels in the same cluster.
Rule 3: Close loop is not allowed.
Rule 4: The foreground boundary pixels cannot be a member of link.
Rule 5: If many links are found, the best link is selected based on the minima of defined, sc, Eds, and Slds.
Rule 6: The nearest neighbor is selected for next step if more than one neighbor has the same maximum ED.
Rule 7: For the sake of efficiency, a stop condition is set for multipath searching.

Fig 38 shows the connection result.

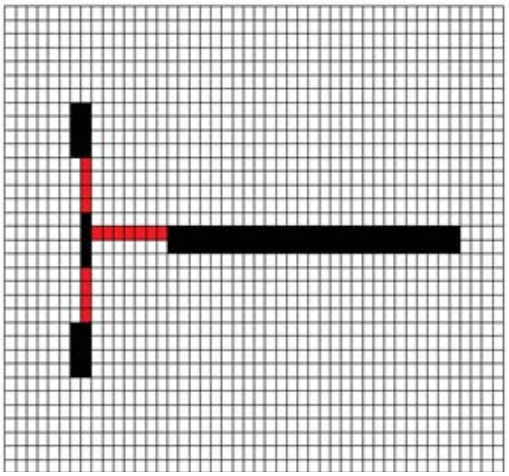

Figure 38: Connection of clusters

- *Post rule based Thinning:* To solve the two pixel width phenomenon keeping the connectivity, the author used a rule based thinning algorithm of [31] to get one pixel width thinned final skeleton.

Fig 39 shows the final skeleton.

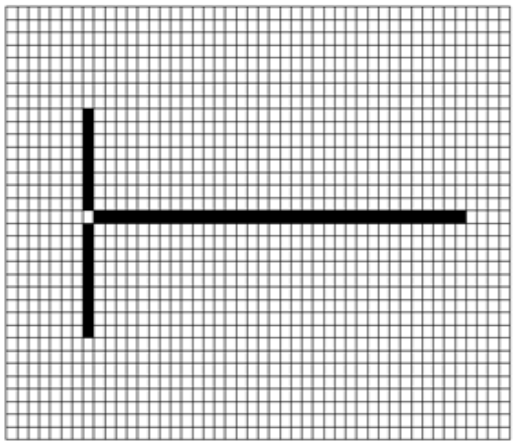

Figure 39: Final skeleton

2.2.8 MAT based skeleton

Sharma and Kaur in 2013[32] produce a medial axis transformation based skeleton. Here the medial axis is extracted based on Euclidian distance.

The steps used for the extraction of MAT.

- Take the input image.
- Compute the Euclidean Distance Transformation (EDT), which represents the Euclidean Distance as Gray Levels.
- Convert the EDT image into 2dimentional matrix extracting pixel coordinate of the maximum gray level intensity.

The row and column of the matrix depicts the MAT line of the input image.

2.2.9 Dynamic and competitive skeletonization: Pankaj M.Pandit, Dr. Sudhir G. Akojwar and Salim A. Chavan in 2014introduced a Skeletonization method for character recognition which is trained by neural network, and determines the representative points and connections making up the skeleton by combining the AVGSOM non-supervised learning[33].

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically 2D), discretized representation of the input space of the training samples, called a map. Self-organizing maps are different from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and in the sense that they use a neighborhood function to preserve the topological properties of the input space.

- The initial step of this algorithm is placing the artificial neurons in one of the most representative areas in the image.Each of these neurons is represented as the node of completely connected graph, where the weight of each arc represents the distance between the neurons it connects.
- Then search the minimum spanning tree.
- Iterate through arcs, to segment the arcs whose value is greater than a predefined value α, which specify the maximum distance between two neurons of the tree.
- Stop when all the arcs are equal or below α.

During iterations, the arc having greatest weight is segmented and a new AVGSOM network if formed which initially consists of three neurons: two, which are connected by arc and a new one whose initial weight is average weight of two partition.

Fig 40 (page 2, [33]) shows the construction of final skeleton from the neurons through different stages of competition and replacement.

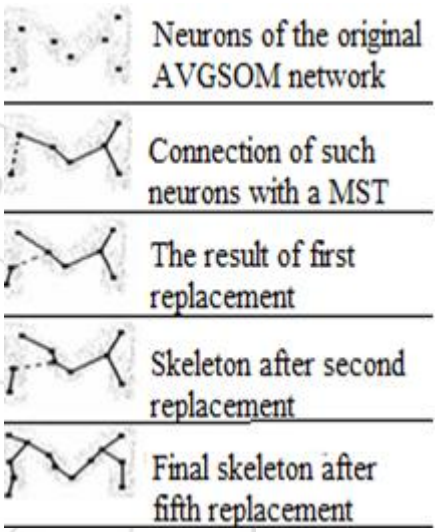

Figure 40: Initial neuron to final skeleton through several stages of replacement

4. Conclusion

The survey is based on the different methods of image thinning and skeletonization. General ideas of what a good skeleton should be. More studies are required to measure the application specific measurement criteria. This survey will help in the new step in digital image processing.

References

[1] T. Y. Zhang and C. Y. Suen, "*A Fast Parallel Algorithm for Thinning Digital Patterns*", ACM, VOL.27, NO. 3 March 1984.
[2] H.E. Lu and P.S.P. Wang, "*An improved fast parallel algorithm for thinning digital patterns*", IEEE Conf. on computer vision and pattern recognition .pp. 364-367, 1985.
[3] Christopher M. Holt, Alan Stewart, Maurice Clint, and Ronald H. Perrott, "*An Improved Parallel Thinning*", Communications of the ACM, Volume 30 Number 2, February 1987.
[4] Y.Y. Zhang and P.S.P. Whang, "*A Modified Parallel Thinning Algorithm*", CH2614-6/88/0000/1023, IEEE 1988.

[5] Z. Guo and R. Hall, "*Parallel thinning with two-sub iteration algorithms*", Communications of the ACM, vol. 32, pp. 359–373, Mar 1989.
[6] Ben. K. Jang and Roland. T. Chin, "*One-Pass Parallel Thinning: Analysis, Properties and Quantitative Evaluation*", IEEE transactions on pattern analysis and machine intelligence, vol. 14, no. 11, November 1992.
[7] G.S. Ng, R. W. Zhou and C. Quek, " *A Novel Single Pass Thinning Algorithm*", IEEE transaction on system man and cybernetics, 8 September 1994.
[8] http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton.html [Accessed: 10-October-2015]
[9] Se Hyun Park, Sang Kyoon Kim and Hang Joon Kim, "*A fully parallel thinning algorithm using a weighted template*", Digital Signal Processing Application, 0-7803-3679-8 IEEE, 1996.
[10] N H Han, C W La and P K Rhee,"An Efficient Fully Parallel Thinning Algorithm", 0-8186-7898-4/97, IEEE, 1997.
[11] J.S. Kwon, J.W. Gi, and E.K. Kang, "*An Enhanced Thinning Algorithm Using Parallel Processing*", Proc. of 11th IEEE International Conference on Image Processing, vol. III, Thessaloniki Greece, Oct. 2001.
[12] Maher Ahmed and Rabab Ward, "*A Rotation Invariant Rule-Based Thinning Algorithm for Character Recognition"*, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 24, No. 12, December 2002.
[13] Lei Huang, Genxun Wan, Changping Liu, "*An Improved Parallel Thinning Algorithm*", Proceedings of the Seventh International Conference on Document Analysis and Recognition, IEEE, 2003.
[14] Peter I. Rockett, "*An improved rotation-invariant thinning algorithm*", IEEE Trans Pattern Analysis And Machine Intelligence. Volume:27 , Issue: 10, Oct 27, 2005.
[15] Fei Xie, Guili Xu, Yuehua Cheng, Yupeng Tian, "*An Improved Thinning Algorithm for Human Body Recognition*", IEEE Conference Publishing, IST 2009 - International Workshop on Imaging, May 11-12, 2009.
[16] Miss G. V. Padole Dr. S. B. Pokle, *"New Iterative Algorithms For Thinning Binary Images"*, Third International Conference on Emerging Trends in Engineering and Technology,IEEE,2010.
[17] A. Jagna and V. Kamakshiprasad, *"New parallel binary image thinning algorithm"*, ARPN Journal of Engineering and Applied Sciences. VOL.5, NO.4, April 2010.
[18] Jun-Sik Kwon, "*Improved parallel thinning algorithm to obtain unit-width skeleton*", The International Journal of Multimedia & Its Applications (IJMA) Vol.5, No.2, April 2013.
[19] A. Jagna, "*An Efficient Image Independent Thinning Algorithm*", International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 10, October 2014.
[20] Lynda Ben Boudaoud, Abderrahmane Sider, Abdelkamel Tari, "*A New Thinning Algorithm for Binary Images*".
[21] Cecil G.A, "*Modeling requirements for finite element analysis*", Computer-Aided Design, 1994, 26, (7), pp. 573-578.
[22] Attali D, Boissonnat J.D, and Edelsbrunner H, "*Stability and computation of medial axes-a state-ofthe-art report*", Mathematical foundations of scientific visualization, computer graphics, and massive data exploration, 2009, pp. 109–125.
[23] Gelston, S.M, and Dutta, D, "*Boundary surface recovery from skeleton curves and surfaces*", Computer Aided Geometric Design, 1995, 12, (1), pp. 27–51.
[24] H. Blum, "*A transformation for extracting new descriptors of shape*". In W. Wathen-Dunn, editor, Models for the Perception of Speech and Visual Form, pages 362-380. MIT Press, 1967.
[25] Gabriella Sanniti Di Baja and Edouard Thiel, "Skeletonization *algorithm running on path-based distance maps*", Image and Vision Computing, ELSEVIER, March 1996.
[26] Chuhua Xian, Yusheng Li, Shuming Gao, "Constructive MA Generation for 2D Models", 978 -1-4244-1579-3/07, IEEE, 2007.
[27] Soumen Bag and Gaurav Harit, "*A Medial Axis based Thinning Strategy and structural feature extraction of character images*", Proceedings of 2010 IEEE 17th International Conference on Image Processing, September 26-29, 2010.
[28] M. Narayana Sandeep V.M Subhash Kulkarni, " *Skeleton based Signatures for Content based Image Retrieval*", International Journal of Computer Applications (0975 – 8887)Volume 23– No.7, June 2011.
[29] Lu Liu, Erin W. Chambers, David Letscher, Tao Ju, "*Extended Grassfire Transform on Medial Axes of 2D Shapes*" Preprint submitted to Elsevier, 20 June 2011.
[30] Yung-Sheng Chen, Ming-Te Chao, *"Skeletonization based on the medial-axis and symmetry Information"*, IEEE, Proceedings of the 2012 International Conference on Machine Learning and Cybernetics, Xian, 15-17 July, 2012.
[31] Y. S. Chen and W. H. Hsu, "*A systematic approach for designing 2-subcycle and pseudo 1-subcycle parallel thinning algorithms*", Pattern Recognition, Vol. 22, No. 3,267-282, 1989.
[32] Shivani Sharma and Maninder Kaur, "*Medial Axis Transformation based Skeletonzation of Image Patterns using Image Processing Techniques*", IOSR Journal of Computer Engineering, e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 4 ,Jul. - Aug. 2013.
[33] Pankaj M.Pandit, Dr. Sudhir G. Akojwar, Salim A. Chavan, "*Dynamic and Competitive Skeletonization for Recognition of Decorative Characters*" , International Conference on Electronic Systems, Signal Processing and Computing Technologies, 978-1-4799-2102-7/14 IEEE, 2014.