

Experimental Approach for Performance Analysis of Thinning Algorithms for Offline Handwritten Devnagri Numerals

Gulshan Goyal^{1*} and Maitreyee Dutta²

¹I.K. Gujral Punjab Technical University (Punjab) & Faculty of Engineering, CCET, Chandigarh – 160019, India; gulshan_goyal1@rediffmail.com
²NITTTTR, Chandigarh-160019, India; d_maitreyee@yahoo.co.in

Abstract

Objectives: Performance and efficiency of thinning algorithms is essential in the field of image analysis and recognition. The present paper aims at experimental approach for performance analysis of different thinning algorithms for offline handwritten devnagri numeral script on multiparameter scale. **Methods/Statistical Analysis:** Algorithms based on datasets are reviewed and three algorithms based on their characteristics and strengths are implemented and their performance is evaluated based on pixel count in output image, compression ratio, pixel removal parameter, connectivity, triangle counts, unit pixel width, and information loss and topology preservation measure. **Findings:** Experimental findings indicate the strength and weakness of each thinning algorithm. **Application/Improvements:** The novelty of work is use of large parameter set for experimental performance evaluation. The findings and subsequent discussion aim at providing parametric strength of different thinning algorithms.

Keywords: Devnagri Numeral, Handwritten Character Recognition, Skeleton, Thinning, Topology, Triangle Count, Unit Pixel Width

1. Introduction

The concept of thinning was introduced by Blum in 1964. Thinning is an important preprocessing stage^{1,2} in HCR to obtain a skeleton of given pattern. Thinning makes feature extraction and recognition process easy and efficient by reducing a given pattern to unit pixel thickness. The thinning process transforms a digital pattern from one form to another with less thickness as described in Figure 1. The main advantages of thinning algorithms are:

- a) To represent a pattern by a collection of thin arcs and curves of unit pixel width.
- b) To reduce the amount of data and processing time required so that process of feature extraction is speeded up.
- c) To facilitate the extraction of critical features such as end-points, junction-points³.

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

e) Shape analysis and classification can be more easily done on line like patterns⁴.

A good thinning algorithm should satisfy geometrical and topological requirements⁵ i.e. the skeleton must be in the middle of the original object and must be invariant to translation, rotation, and scale changes. Further, the skeleton must retain the topology of the original object i.e. output pictures should be topologically equivalent to the input ones.

2. Thinning Procedure for Pixel Deletion

In a thinning algorithm, pixels are examined based on pixel deletion criteria specified in the algorithm and

*Author for correspondence

flagged for deletion depending on the whether the specified criteria is satisfied or not. Algorithmic procedure for pixel deletion to generate a skeleton is described in Figure 2.



Figure 1. Thinning Process for Transforming Digital Patterns.

The result of thinning transformation applied on a binary devnagri numeral image is illustrated in Figure 3.

2.1 Classification of Thinning Algorithms

Thinning algorithms can be classified into two categories i.e. iterative thinning algorithms and non iterative thinning algorithms^{2,6,7}. Non iterative thinning algorithms produces a median line or some centre line of the pattern and then take a decision whether to delete that particular boundary pixel or not. Iterative (pixel based) thinning algorithms examine the individual pixels in a binary image based on pixel deletion criteria, flags the pixels for deletion and deletes the boundary pixels until a unit width skeleton is generated^{2,6,7}.

Iterative thinning algorithms can be either sequential or parallel thinning algorithms. In Sequential thinning algorithms pixels selected for deletion are chosen in a predetermined order. In Parallel thinning algorithm pixels are selected for deletion purpose based on the result of previous iterations^{2,6,7,8}.



Figure 3. (a): Input Numeral Image (b): Thinned Numeral Image.

2.2 History Specifying Application of Thinning to Different Datasets

Since the introduction of thinning concept, a large of thinning algorithms are proposed by different researchers. Researchers have applied thinning algorithms on different scripts summarized in Table 1.

From the Table 1, it is observed that very few researchers have emphasized on devnagri numeral script. Devnagri script is used for writing many official languages in India. More than 300 million people use Devnagri script for documentation⁹. Therefore, processing of devnagri script in HCR system is an active area of research. In present study, experimental results of performance parameters are summarized for devnagri numeral script.

The remainder of paper is organized as follows: Three thinning algorithms¹⁰⁻¹² are discussed in section 3. Thinning parameters used for performance evaluation are described in section 4. Experimental results are presented and discussed in section 5. Conclusion and future scope is given in section 6.

Algorithm: Thinning Procedure for Pixel Deletion

Input: An offline handwritten binary numeral image.

Output: A thin arc like binary numeral image that is of unit pixel width.

Begin

Repeat following steps in given order until no more pixels are flagged in a particular iteration:

1. Examine each foreground pixel of image based on pixel deletion criteria of thinning algorithm.
2. Flag the pixels if conditions of pixel deletion criteria are satisfied.
3. Delete all the pixels flagged in step 2.

End

Figure 2. Thinning Procedure for Pixel Deletion.

3. Materials and Methods: Algorithms for Thinning

In this section, algorithmic details for pixel deletion are provided in different thinning algorithms. In the scope of present paper, three algorithms¹³⁻¹⁵ are reviewed and applied on devnagri numeral script to analyze performance parameters. Thinning algorithm based on two subiterations¹⁰ is one of the fundamental algorithms for thinning of binary images. This algorithm has been in use as a base for proposing new thinning conditions by different researchers or evaluating performance of any proposed algorithm. Many researchers^{8,16,17} used the algorithm for proposing their methods. Further the issue of information loss and two pixel wide lines is addressed on handwritten number images¹⁷. Adaptive thinning framework^{18,19} is based on Gaussian blurring is proposed for noisy sketch images.

3.1 Algorithm 1: Thinning Algorithm Based on Two Subiterations¹⁰

The algorithm consists of two sub iterations for flagging and deleting a pixel. Each sub iteration consists of four conditions¹⁰. The algorithm makes use of 3x3 masks as shown in Figure 4.

A pixel is marked for deletion in particular sub iteration if all given conditions in particular sub iteration are satisfied. The detailed algorithm is described in Figure 5.

P₉	P₂	P₃
P₈	P₁	P₄
P₇	P₆	P₅

Figure 4. A 3X3 Pixel Mask Based on 8-Neighbors.

First sub iteration removes the south-east boundary points and the north-west corner points which do not belong to an ideal skeleton. Second sub iteration removes only the north- west boundary points and the south - east corner points which do not belong to an ideal skeleton.

3.2 Algorithm 2: Thinning Algorithm Based on Information Loss and Two Pixel Width¹⁷

The algorithm is based on 3x3 neighborhoods and uses elimination rules. Further, this algorithm adds contour

information to overcome possible loss of information¹⁷. The algorithmic procedure is described in Figure 6.

Table 1. History of Thinning Algorithms Applied to Different Datasets

Year	Name of the Author	Data Set
1979	Dyer C R and Rosenfeld A ⁹	Grayscale Pictures
1984	Zhang T Y and Suen C Y ¹⁰	Different Digital Patterns
1988	Abdullah W H et. al ¹¹	Hand Written Characters
1991	Kundu M. et. al. ³	Different types of images
1992	Lam L et. al. ¹²	Binary Image Dataset
1992	Jang B K and Chin R T ¹³	Binary Images
1994	Datta A and Parui S K ¹⁴	Different Patterns
1994	Ng G S et. al. ¹⁵	Chinese Signatures
1995	Nagendraprasad M V et. al. ¹⁶	Character Database extracted NIST
2003	Huang L et. al. ¹⁷	Handwritten number images
2008	Bai X. et. al. ¹⁸	Different Shapes from MPPEG-7 database
2010	Saeed K et. al. ⁶	Isolated Letters, Handwritten words and Graphical symbols
2010	Padole G V and Pokle S B ⁷	Different Images
2008	TARABEK P ¹	Test case images
2011	Bag S and Harit G ⁴	Printed English, Bengali, Hindi, and Tamil characters
2011	Latecki J et. al. ¹⁹	Binary Images
2012	Ali M A ²⁰	Arabic Handwritten Images
2012	Chen W et. al. ²¹	Pony Drawing Image
2013	Abu-Ain W et. al. ²	MPEG-7 Shape Dataset
2014	Chatbri H, Kameyama K. ⁸	Noisy Sketch Images

3.3 Algorithm 3: Adaptive Thinning Algorithm Based on Gaussian Filtering²⁰

Adaptive Thinning algorithm is based on 3x3 neighborhood of pixels that perform thinning in two stages:

Algorithm 1: Thinning Algorithm Based on Two Sub- Iterations¹⁰

Begin

Sub iteration 1

1. Number of non-zero neighbors of a pixel should be between 2 and 6.
2. Number of 01 patterns in the sequence of eight neighbors of a pixel under consideration are 1
3. At least one of the neighbors from P2, P4, or P6 is zero.
4. At least one of the neighbors from P4, P6, or P8 is zero.

Sub iteration 2

1. Number of non-zero neighbors of a pixel should be between 2 and 6.
2. Number of 01 patterns in the sequence of eight neighbors of a pixel under consideration are 1
3. At least one of the neighbors from P2, P4, or P8 is zero.
4. At least one of the neighbors from P2, P6, or P8 is zero.

End

Figure 5. Thinning Algorithm Based on Two Sub Iterations.

Algorithm 2: Thinning Algorithm Based on Information Loss and Two Pixel Width¹⁷

Begin

1. Apply pixel elimination rules¹⁷ simultaneously to each pixel.
2. If the pixel under consideration is eliminable, check the width of pixel.
3. If the pixel under consideration is not two pixel wide then delete it
4. If the pixel under consideration is two pixel wide then use preservation templates¹⁷ to decide whether the pixel should be deleted or not.

End

Figure 6. Thinning Algorithm Based on Elimination Rules and Preservation Templates.

Algorithm 3: Adaptive Thinning Algorithm Based on Gaussian Filtering²³

Begin

Pre-Thinning Steps:

1. Apply 5x5 gaussian filter for different values of sigma ranging from 1 to 17 to produce blurred images.
2. The value of threshold is selected by experimentation corresponding to best topology.

Stage 1: A pixel is flagged for deletion based on following conditions²³:

1. Number of non-zero neighbors of a pixel should be between 2 and 6.
2. Number of black to white transitions in the sequence of eight neighbors of a pixel under consideration are 1
3. Pixel under consideration have at least one non-border pixel.

Stage 2: Pixels belonging to two pixel wide strokes are examined

1. Apply elimination mask²³
2. Delete the pixel when matching with any of templates²³

End

Figure 7. Adaptive Thinning Algorithm Based on Gaussian Filtering.

- a) A parallel thinning stage for removing border pixels based on conditions of stage 1 of Algorithm²¹. The output skeleton generated is one or two pixel wide.
- b) This is a sequential stage in which pixels belonging to two pixel wide strokes are eliminated by applying the elimination masks²².

The thinning results obtained by direct application of algorithm are not good and contains different types of noise in output skeletons. In order to overcome the noise problem authors²³ proposed adaptive thinning framework using Gaussian filter and then binarizing the image using mean threshold value. The authors choose the skeleton by calculating noise sensitivity for each generated skeleton. A final skeleton is selected based on minimum value of sensitivity^{8,23,24}. In present study the adaptive thinning methodology is slightly modified in terms of Gaussian filter and threshold value. In present paper, Gaussian filter of size 5x5 are used. The values of sigma are ranging from 1 to 17 and the threshold value is chosen based on experimentation. The results showing best skeleton are chosen by visual inspection. The present algorithm is described in Figure 7.

4. Performance of Thinning Algorithms

Performance evaluation is an important criterion for applicability of any algorithm in a particular application. A large number of thinning algorithms and techniques available so far involves trade-offs among one, or more of the topological and geometrical requirements. Thinning algorithms can be evaluated on the basis of following parameters:

- Number of pixels in output image
- Compression Ratio
- Pixel Removal Parameter
- Connectivity
- Unit Pixel Width
- Information Loss
- Topology measure

The parameters are described as given below:

a). Number of Pixels in Output Image: This measure counts the total number of pixels in the generated skeleton. An ideal skeleton should have nearly half the number of pixels present in contour of original image^{8,17,23}. However, it does not that skeletons having number of pixels more or less than half are not good skeletons. The variations should not be unreasonably large.

b). Compression Ratio: Compression Ratio can be defined as ratio of number of object pixels count in original image to object pixel count in thinned image. Mathematically, Compression Ratio CR can be calculated as:

$$CR = \frac{N_o}{N_{th}}$$

Where N_o = No. of object pixels in original image, and
 N_{th} = No. of object pixels in thinned image.

c) Pixel Removal Parameter: This parameter measures the extent to which pixels from original image pattern are deleted. Mathematically, pixel removal parameter PR is formulated as:

$$PR = \left(1 - \frac{N_{th}}{N_o}\right)$$

The value of PR can be multiplied with 100 to get the percentage of pixels removed during thinning process.

d) Connectivity: It checks the connectivity of pixels in output skeleton. If number of connected components in the skeleton are 1, it indicates that all the pixels in thinned image are connected.

e) Unit Pixel Width: In order to check the width or thinness of skeleton, following two parameters based on triangle counts are considered:

i) Thinning Rate¹: This measure counts the No. of triangle counts in original and thinned image and gives an idea about the pixels width. Triangle Thinning rate TTR is calculated as given below:

$$TTR = \left(1 - \frac{TTC_{th}}{TTC_o}\right)$$

Where TTC_{th} = Total triangle count in thinned image
 TTC_o = Total triangle count in original image.

If the value of TTR is 1, skeleton is perfectly thin and if the value of TTR is 0, image is not thinned at all.

ii) Thinness Measurement¹⁵: The thinness measurement (TM) parameter also measures the extent to which a pattern in the image is thinned. Mathematically, TM can be calculated as:

$$TM = \left(1 - \frac{TM1}{TM2}\right)$$

Where $TM1$ = Total triangle count in thinned image
 $TM2 = 4 * [\max(\text{height}, \text{width}) - 1]^2$

The measure TM will have values between 0 and 1. When the value of TM is 1, the pattern is completely unit pixel wide.

Table 2. Pixel Count in Input Image, Contour Image and Thinned Image

Numeral	Pixel Count (Input Image)	Pixel Count (Image Contour)	Pixel Count (Thinned Image)		
			Algo1	Algo2	Algo3
०	553	255	141	133	124
१	461	195	113	110	94
२	433	184	110	102	83
३	412	176	99	101	78
४	551	245	140	116	121
५	380	157	80	71	68
६	503	212	107	109	92
७	387	168	98	55	69
८	426	192	109	87	98
९	415	172	99	96	77

f) Information Loss¹⁷: Authors described a parameter R to judge whether while applying thinning algorithm there is loss of information or not. The mathematical formulation for R is:

$$R = \frac{\text{Area of skeleton}}{\text{Area of contour}}$$

The value of R is compared with a chosen threshold value. The threshold value of R is chosen as 0.4 by authors¹⁷. If the value of R for a thinned image is considerable smaller than the chosen threshold, the contour of image is retained.

g) Topology Preservation Measure^{8,23}: Topology preservation means that thinned image should deliver the same visual information as the original image. The topology preservation measure T is defined as:

$$T = 1 - \left| \frac{1}{2} - \frac{\text{No. of Object Pixels in Thinned Image}}{\text{No. of Contour Pixels in Original Image}} \right|$$

As the skeleton generated should be near medial axis, it is expected that number of contour pixels to be nearly two times as that in skeleton. The value of T near 1 express topology preservation while the values near 0 indicates

Table 3. Compression Ratio and Pixel Removal Parameter.

Numeral	Compression Ratio (CR)			Pixel Removal Parameter (PR)		
	Algo1	Algo2	Algo3	Algo1	Algo2	Algo3
०	3.92	4.16	4.45	0.745027	0.759494	0.775769
१	4.08	4.19	7.20	0.754881	0.761388	0.796095
२	3.94	3.94	5.21	0.745928	0.764434	0.808314
३	4.16	4.08	5.28	0.759709	0.754854	0.810680
४	3.94	5.35	4.55	0.745916	0.789474	0.780399
५	4.75	5.35	5.58	0.789474	0.813158	0.821053
६	4.70	4.61	5.46	0.787276	0.783300	0.817097
७	3.95	7.03	5.60	0.746770	0.857881	0.821705
८	3.91	4.89	4.34	0.744131	0.795775	0.769953
९	4.19	4.32	5.38	0.761446	0.768675	0.814458
AVG	4.15	4.79	5.30	0.758056	0.784843	0.801552



Figure 8. (a) Devnagri Numeral Data Set (b) Contour of Devgari Numerals (c) Results using Algorithm 1 (d) Results using Algorithm 2 (e) Results using Algorithm 3.

distortion. The Topology measure values for three algorithms is shown in Table 1.

5. Experimental Results and Discussion

The algorithms^{10,17,23} are implemented using C programming language. Dataset for experimentation consist of handwritten devnagri numeral script. Experimental results are illustrated in Figure 8.

The results shown in Figure 8 are evaluated based on the performance parameters described in previous section. The number of object pixels in original image, contour and thinned image are summarized in Table 2. The values of CR and PR are listed in Table 3.

The graphical representation of PR values is shown in Figure 9.

Average values of CR are 4.15, 4.79 and 5.30 respectively for Algo1, Algo2 and Algo3. Average value of PR in terms of percentage is varying between 75% to 80%. In Algo1, Algo2 and Algo3 average 75.80%, 78.48% and 80.15% pixels are deleted respectively during thinning process. In terms of connectivity, Algo1 and Algo2 shows good results and skeletons are connected. However, as clear from visual inspection, in Algo3, some disconnections are detected resulting in more PR value. The triangle counts measure in input and thinned images are summarized listed in Table 4.

Based on Triangle counts, Triangle Thinning Rate and Thinness Measure are calculated and summarized in Table 5.

From the Table 5, it is observed that average triangle thinning rate using Algo1, Algo2 and Algo3 are 97.65%,

99.45% and 99.24% respectively. Further, thinness measure using algo1, algo2 and algo3 are 99.92%, 99.98% and 99.97% respectively indicating that skeletons are almost one pixel wide. The values of Information loss parameter R and topology measure T are represented in Table 6. In present study, with reference to threshold value of 0.40, it is observed that there is no significant loss of information. The topology measure shows the extent to which topology is retained using the three algorithms.

Table 4. Triangle Count in Input and Thinned Images

Numeral	Triangle Count (Input Image)	Triangle Count (Thinned Image)		
		Algo1	Algo2	Algo3
0	1633	30	6	11
१	1367	35	9	13
२	1288	34	10	8
३	1216	33	14	8
४	1621	37	9	18
५	1159	17	1	10
६	1506	28	4	6
७	1146	30	10	4
८	1244	34	2	20
९	1236	34	7	4

Table 5. Triangle Thinning Rate and Thinness Measure

Numeral	Triangle Thinning Rate (TTR)			Thinness Measure (TM)		
	Algo1	Algo2	Algo3	Algo1	Algo2	Algo3
०	0.981629	0.996326	0.993264	0.999219	0.999844	0.999714
१	0.974396	0.993416	0.990490	0.999089	0.999766	0.999662
२	0.973602	0.992236	0.993789	0.999115	0.999740	0.999792
३	0.972862	0.988487	0.993421	0.999141	0.999636	0.999792
४	0.977175	0.994448	0.988896	0.999037	0.999766	0.999531
५	0.985332	0.999137	0.991372	0.999557	0.999974	0.999740
६	0.981408	0.997344	0.996016	0.999271	0.999896	0.999844
७	0.973822	0.991274	0.996510	0.999219	0.999740	0.999896
८	0.972669	0.998392	0.983923	0.999115	0.999948	0.999479
९	0.972492	0.994337	0.996764	0.999115	0.999818	0.999896
AVG	0.976539	0.994540	0.992445	0.999188	0.999813	0.999735

Table 6. Information Loss and Topology Measure

Numeral	Information Loss (R)			Topology Measure (T)		
	Algo1	Algo2	Algo3	Algo 1	Algo2	Algo3
०	0.552941	0.521569	0.486275	0.947059	0.978431	0.986274
१	0.579487	0.564103	0.482051	0.920513	0.935897	0.982051
२	0.597826	0.554348	0.451087	0.902174	0.945652	0.951087
३	0.562500	0.573864	0.443182	0.937500	0.926136	0.943182
४	0.571429	0.473469	0.493878	0.928571	0.973469	0.993878
५	0.509554	0.452229	0.433121	0.990446	0.952229	0.933121
६	0.504717	0.514151	0.433962	0.995283	0.985849	0.933962
७	0.583333	0.457381	0.410714	0.916667	0.827381	0.910714
८	0.567708	0.453125	0.510417	0.932292	0.953125	0.989583
९	0.575581	0.558140	0.447674	0.924419	0.941860	0.947674

Graphical representation of T is shown in Figure 10. Average values of CR, PR, TTR and TM are summarized in Table 7. Average CR is plotted in Figure 11 and average values of PR, TTR and TM are plotted in Figure 12.

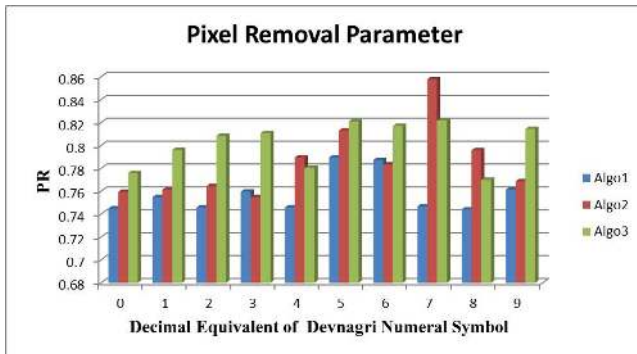


Figure 9. Graphical Representation of PR Values.

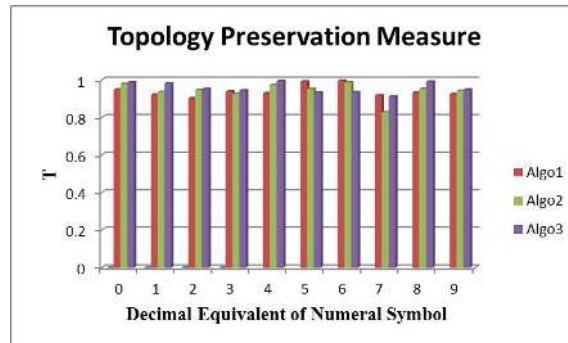


Figure 10. Graphical Representation of T.

6. Conclusion and Future Scope

In present paper, thinning algorithms based on their strengths are studied and implemented for handwritten devnagri numeral script. Experimental approach is



Figure 11. Graphical Representation of Average values of CR for Three Algorithms.

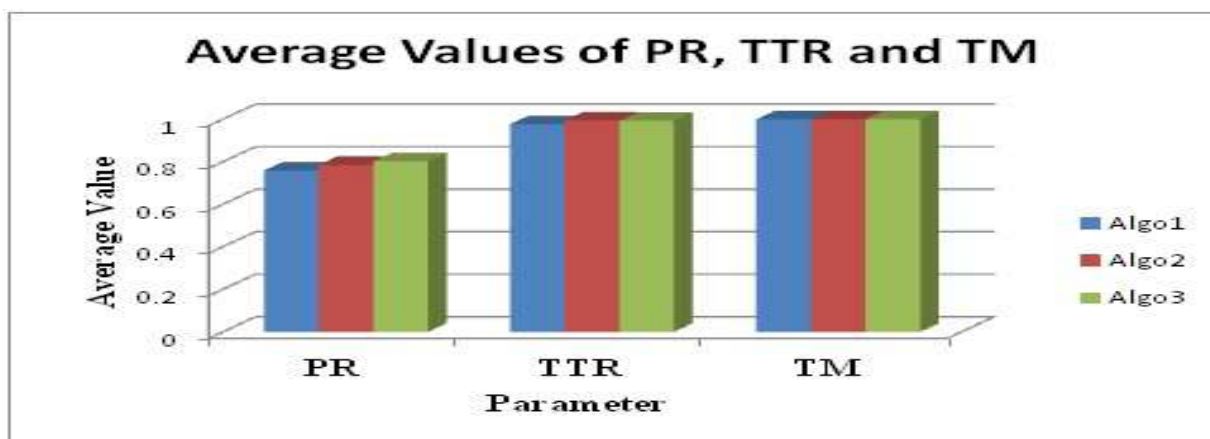


Figure 12. Graphical Representation of Average values of PR, TTR and T for Three Algorithms.

used to analyze performance of thinning algorithms on multiparameter set. From the experimental results, it is observed that algorithm 3 uses adaptive thinning using Gaussian filtering and shows good parametric results for noisy images. However, there are detected disconnections. Further extra processing in terms of Gaussian blurring is required. Algorithm 2 addresses the information loss issue but requires extra processing in terms of preservation templates. Algorithm 1 based on two sub iterations produce nearly comparable results with no extra processing for information loss or inherent noise. Therefore, processing time is comparatively less. In future, keeping in mind the performance evaluation parameters and problem under consideration, new algorithm can be proposed that can give better performance than reviewed algorithms.

7. References

1. Tarabek P. Performance Measurements of Thinning Algorithms. *Journal of Information, Control and Management Systems*. 2008; 6(2):125-32.
2. Abu-Ain W, Abdullah SNSH, Bataineh B, Abu-Ain T, Omar K. Skeletonization Algorithm for Binary Images. 4th International Conference on Electrical Engineering and Informatics. 2013; 11:704-09.
3. Kundu M, Chaudhuri B, Majumder DD. A Parallel Graytone Thinning Algorithm (PGTA). *Pattern Recognition Letters*. 1991; 12(8):491-96.
4. Bag S, Harit G. An Improved Contour Based Thinning Method for Character Images. *Pattern Recognition Letters*. 2011; 32(14):1836-42.
5. Sangeetha V, Vaithyanathan V, Sivagami R, Divyalakshmi K, Sundar KJA, Ahmed MI. Skeletonization Approaches for Determining Paths in an Image: A Review. *Indian Journal of Science and Technology*. 2015; 8(35):1-7.
6. Saeed K, Tabezki M, Rybnik M, Adamski M. K3M: A Universal Algorithm for Image Skeletonization and a Review of Thinning Techniques. *International Journal of Applied Mathematics and Computer Science*. 2010; 20(2):317-35.
7. Padole G, Pokle SB. New Iterative Algorithms for Thinning Binary Images. Goa: Third International IEEE conference on Emerging Trends in Engineering and Technology. 2010; p. 166-71.
8. Chatbri H, Kameyama K. Using Scale Space Filtering to make Thinning Algorithms Robust against Noise in Sketch Images. *Pattern Recognition Letters*. 2014; 42(1):1-10.
9. Dyer CR, Rosenfeld A. Thinning Algorithms for Grayscale Pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1979; 1(1):88-90.
10. Zhang TY, Suen CY. A Fast Parallel Algorithm for Thinning Digital Patterns. *Communications of ACM*. 1984; 27(3):236-39.
11. Abdulla WH, Saleh AOM, Morad AH. A Preprocessing Algorithm for Hand-Written Character Recognition. *Pattern Recognition Letters*. 1988; 7(1):13-18.
12. Lam L, Suen CY. An Evaluation of Parallel Thinning Algorithms for Character Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992; 17(9):914-19.
13. Jang BK, Chin RT. One Pass Parallel Thinning: Analysis, Properties and Quantitative Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992; 14(11):1129-40.
14. Datta A, Parui SK. A Robust Parallel Thinning algorithm for Binary Images. *Pattern Recognition Letters*. 1994; 27(9):1181-92.
15. Ng GS, Zhou RW, Quek C. A Novel Single Pass Thinning Algorithm. *IEEE Transactions on System, Man and Cybernetics*. 1994; p. 215-22.
16. Nagendraprasad MV, Wang PSP, Gupta A. An Improved Algorithm for Thinning Binary Digital Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1995; 14(11):386-89.
17. Huang L, Wan G, Liu C. An Improved Parallel Thinning Algorithm. *Proceedings of Seventh IEEE International Conference on Document Analysis and Recognition*. 2003; p. 780-85.
18. Bai X, Yang X, Latecki L, Xu J, Liu WY. Computing Stable Skeletons with Particle Filters. Springer: PRICAI 2008: Trends in Artificial Intelligence. 2008; p. 30-41.
19. Latecki LJ, Li QN, Bai X, Liu WY. San Antonio, TX: Skeletonization using SSM of the distance transform. 2011; 5:349-52.
20. Ali MA. An Efficient Thinning Algorithm for Arabic OCR Systems. *International Journal on Signal and Image Processing*. 2012; 3(3):31-38.
21. Chen W, Sui L, Xu Z, Lang Y. Improved Zhang-Suen Thinning Algorithm in Binary Line Drawing Applications. *IEEE International Conference on Systems and Informatics (ICSAI)*; 2012; p. 1947-50.
22. Jayadevan R, Kolhe SR, Patil PM, Pal U. Offline Recognition of Devanagari Script: A Survey. *IEEE Transactions on systems, man and cybernetics-part C: Applications and Reviews*. 2011; 41(6):782-96.
23. Chatbri H, Kameyama K. An Adaptive Thinning Algorithm for Sketch Images Based on Gaussian Scale Space. *Technical Report of IEICE*. 2012; 111(442):33-38.
24. Hannah JG, Gladis D. Feature Extraction with Thinning Algorithms for Precise Cretoscopy. *Indian Journal of Science and Technology*. 2015; 8(29):1-7.