Performance Comparison of ZS and GH Skeletonization Algorithms

Ritika Luthra
Research Scholar
Chandigarh University

Gulshan Goyal
Associate Professor
Chandigarh University

ABSTRACT
Skeletonization is a crucial step in many digital image processing applications like medical imaging, pattern recognition, fingerprint classification etc. The skeleton expresses the structural connectivities of the main component of an object and is one pixel in width. Present paper covers the aspects of pixel deletion criteria in the skeletonization algorithms needed to preserve the connectivity, topology, sensitivity of the binary images. Performance of different skeletonization algorithms can be measured in terms of different parameters such as thinning rate, number of connected components, execution time etc. Present paper focuses on thinning rate, number of connected components, execution time on Zhang and Suen algorithm and Guo and Hall algorithm.

Keywords
Skeletonization, Optical character Recognition, ZS, GH, ZSM

1. INTRODUCTION
Skeletonization is the process of extracting skeletons by deleting unwanted pixels from an image. It is morphological operation that deletes black foreground pixels iteratively layer by layer until one pixel width skeleton is obtained. It is a procedure of reducing an object to its minimum size[2].Skeletonization is usually applied on binary images which consist of black (foreground) and white (background) pixels. It takes input to be a binary image, and produces another binary image as output as shown in fig1.

For a skeletonization algorithm to be effective, it should reduce the images into thin like objects and should retain the topological and geometric properties as well. However, a good skeletonization algorithm must have the following features:

1. The resulting skeletons should maintain connectivity.
2. The resulting skeletons should be of unit pixel width.
3. No excessive deletion of pixels should takes place.
4. It should perform better in terms of execution time.
5. It should ideally compress the data.

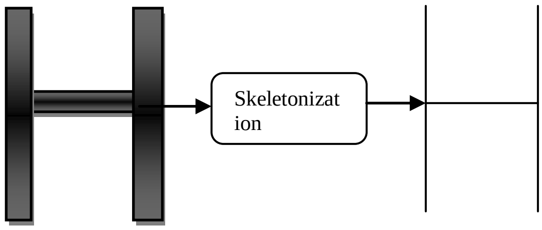


Fig 1: General concept of Skeletonization

1.1 Need of Skeletonization
Skeletonization is a significant step in many image processing applications such as Pattern recognition [2], Optical character recognition [1], and fingerprint classification [3] etc. Therefore, it is an active area of research.

So, there is always a need for good skeletonization algorithms in reference to following parameters:

1. The data that is required to be processed is less [3].
2. To reduce processing time.
3. Extraction of important features such as critical end-points, junction-points, and connection among the components can be helpful in many applications [3].
4. When we reduce an object to a skeleton, unimportant features and image noise can be filtered out.
5. Skeletonization is commonly used for the higher degree analysis and recognition for applications such as diagram understanding, OCR, feature detection, and fingerprint analysis.

1.2 Applications of Skeletonization
Skeletonization has been used for variety of image processing applications like:

1. Optical character recognition (OCR) [1,4]
2. Pattern recognition[2]
3. Fingerprint classification[3]
4. Biometric authentication[4]
5. Signature verification[4]
6. Medical imaging[3]

2. SURVEY OF RELATED WORK

Table 1: Survey of Related Work

S.No	Name of the author	Description
1.	Zhang T.Y et al.[5]	**Pros**: a. Preserves connectivity b. Contour noise immunity. c. Efficient in terms of execution time **Cons**: The resulting skeletons are not of unitary thickness.
2.	Guo Z et al. [15]	**Pros:** a. Parallel speed is superior. b. Produces very thin medial curves. **Cons:** Produces noisy branches in the skeleton
3.	Zhou R.W et al. [6]	**Pros**: a. Fast b. Reliable c. High immunity to boundary noise. **Cons**: It takes more computation time.
4.	Ahmed M et al. [7]	**Pros**: a. Effective b. Fast c. Can thin any symbol in any language, irrespective of the direction of rotation. **Cons**: Unable to thin two-pixel width lines.
5.	Rockett P.I [9]	**Pros**: a. No excessive erosion b. Produce thin skeletons. **Cons**: Ran 18% slower than A-W algorithm, more execution time.
6.	Tarabek P. [14]	**Pros:** **a.** Z-S algorithm preserves connectivity of the skeletons. **b**. Shows better results in Noise sensitivity measurements. **Cons:** Thinning rate i.e. calculated is not good for vectorization of roads.
7.	Padole G.V [2]	**Pros**: a. preserve the connectivity b. Produce thin skeletons. **Cons**: Edge based iterative thinning algorithm is time consuming as compared to optimized thinning algorithm.
8.	Saeed K. et al. [11]	**Pros**: a. It produces a unit-pixel-wide skeleton. b. Better connectivity in output skeletons. **Cons**: As K3M is iterative thinning algorithm, so it requires much more computing power than other algorithms.
9.	Jagna A. et al. [12]	**Pros**: a. Skeletons are perfectly 8-connected b. Does not results in excessive erosion **Cons**: Not efficient in terms of execution time.
10.	Abu-Ain W. [1]	**Pros:** Performance is high in terms of execution time. **Cons**: Topology problem. Cannot preserve shape sometimes.
11.	Kwon J. [16]	**Pros:** a. One-pixel wide skeleton b. No excessive erosion. **Cons:** Requires more number of iterations.

3. OVERVIEW OF SKELETONIZATION ALGORITHMS

In general, all the skeletonization algorithms can be classified into two categories as shown in fig 3:

1. Iterative thinning algorithms: Iterative (pixel based) [1,2] thinning algorithms examine the individual pixels in a binary image and deletes the boundary pixels of the pattern until a skeleton remains. Iterative thinning algorithms are further divided into two categories: sequential thinning algorithms and parallel thinning algorithms [1,2].
   a. In sequential algorithms, the points selected for deletion are chosen in a predetermined order and this can be possible by either raster scanning i.e. from line to line scan or by contour following [3]. Sequential thinning algorithms preserves the connectivity of obtained skeletons but sometimes they results into unacceptably thick skeletons.
   b. On the other hand a parallel thinning algorithm [1, 2] selects pixels for deletion purpose based on the result produced by the previous iterations. For these reasons, these parallel thinning algorithms are appropriate for implementation for parallel processing [3]. Parallel thinning algorithms faces the difficulty of preserving the connectivity of the skeletons. These algorithms are better in terms of computing time or the number of iterations or sub iterations used.
2. Non iterative thinning algorithms [1, 2]: Non iterative [1,2] thinning algorithms do not scan individual pixels one by one. Instead, they produces a median line or some centerline of the pattern and then take a decision whether to delete that particular boundary pixel or not. Some popular non-pixel based methods are medial axis transforms, distance transforms, and determination of centerlines by line following [2]. These algorithms show much efficiency in terms of computation time but at the same time these are responsible for creating noise branches in a skeleton.

4. EXISTING ALGORITHMS

4.1 Zhang and Suen Algorithm

Input Image: Pre-processed Image

Output Image: Skeleton of pre-processed image

Assumption: Pixel to be examined is black.

Sub-iteration 1:

1. 2 < = $B(P1)$ < = 6
2. A(P1) = 1
3. At least one of P2 and P4 and P6 is white
4. At least one of P4 and P6 and P8 is white. [5]

Sub-iteration 2:

1. 2 < = $B(P1)$ < = 6
2. A(P1) = 1
3. At least one of P2 and P4 and P8 is white
4. At least one of P2 and P6 and P8 is white

After checking all the conditions, the pixel is deleted otherwise not [5].

4.2 Guo and Hall Algorithm

It uses 3*3 templates for pixel deletion. Let C(P) be the number of 8-connected components of 1's in its neighbourhood.

N (P) =Minimum of (N1(P), N2(P))

N1 (P) = (p1or p2)+(p3or p4)+(p5or p6)+(p7or p8)

N2 (P) = (p2or p3)+(p4or p5)+(p6 or p7)+(p8or p1)

An edge point in the image will be deleted if it satisfies following two conditions:

a. Number of distinct 8 connected components should be one.
b. Number of non zero neighbours should be between 2 and 3 .
c. Apply one of the following:
   i. (P2vP3vP5)*P4=0 in odd iterations;
   ii. (P6vP7vP8)*P8=0 in even iterations Where "v" expresses the logic "OR" operation. C (P)=1 means P is 8-simple [15].

In other words, there is only one group of 8-connected 1's around P. Under this condition, deletion of P will not break the connectivity of the elements in the 3*3 window under processing. Condition (a) guarantees P is not a break point. The GH algorithm is better in detecting the end points than the ZS algorithm. The use of N (P) allows one to identify the end points whether or not they have one or two 1's 8-neighbours [15].

5. PERFORMANCE MEASURES

There are number of performance measures on the basis of which we can measure various skeletonization algorithms. Some of them are described below:

1. Connectivity Measurement CM [6]: It is used to measure the connectivity in the skeletons that are produced as outputs. This is given by:

$$CM=\sum_{x=0}^{n}\sum_{y=0}^{m} S(P[x][y]) \quad [6]$$

Where

$$S(P[x][y]) = \begin{cases} 1, & if\ CN(P[x][y] < 2 \\ 0, & otherwise \end{cases} \quad [6]$$

Where CN is defined as current neighborhood function and is defined as follows:

$$CN(P_0) = \sum_{x=1}^{8}(P_x * Q_x) \quad [6]$$

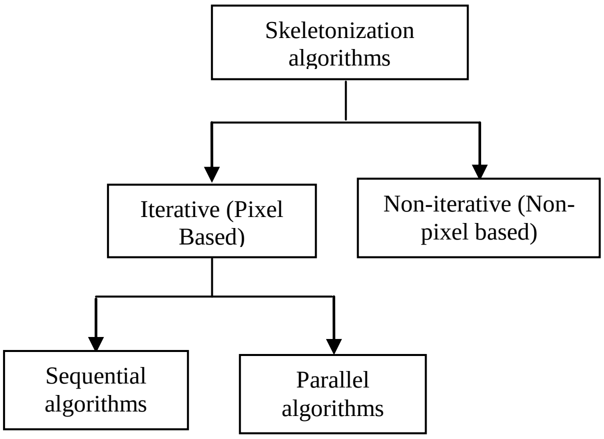


Fig 3: Taxonomy of skeletonization algorithms

Connectivity can be measured in terms of:

1. Number of Connected Components [14]: It basically counts the total number of separated regions or components as shown in fig 4. It is used to measure whether obtained skeleton is connected or not.
2. Thinning Rate (TR): The degree to which an image can be thinned or completely thinned can be possibly measured in terms of thinning rate as shown in fig5 [15].

The thinning rate is given by the equation:

$$TTC=\sum_{x=1}^{n}\sum_{y=1}^{m} TC\ (P[x][y]) \qquad [14]$$

Where:

TTC means total triangle count.

n , m are dimensions of input image.

P[x] [y] are the black pixels with coordinates x,y

The TR is defined as follows:

$$TR=1-\frac{TTC_T}{TTC_O} \qquad [14]$$

Where:

TTCO indicates total triangle count of original image

TTCT indicates total triangle count of thinned image

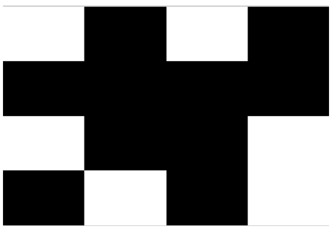


Fig 4: Image with 4 components (NOC=4)

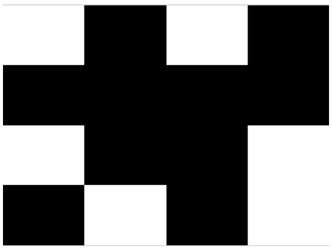


Fig 5: Image with Euler number= 4

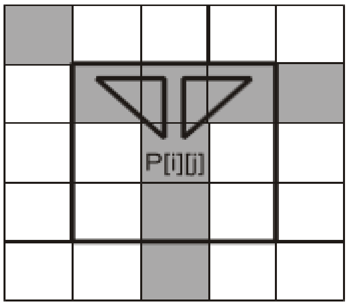


Fig 6: TC=2 [14]

TTCO stands for total triangle count of original image

3. Mean Square Error [16]: Mean square error calculates the difference between the original input image and the skeletonized image. For example: If we have two images and that images are identical in every aspect then MSE between the images is considered to be zero. Lesser the MSE better is the quality of the image. MSE can be defined as follows:

$$MSE=\frac{1}{mn}\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}[I(x,y)-K(x.y)]^2 \qquad [16]$$

4. Peek Signal to noise ratio (PSNR)[16]: Peek signal to noise ratio is usually abbreviated as PSNR, it is used to measure the quality of reconstructed image or we can say skeletonized image [16].

If the MSE in case of any images is zero, then PSNR value reaches to infinite.

Larger the PSNR better is the quality of the image. PSNR is defined by:

$$PSNR= 20*\log_{10}(MAX_I) -10*\log_{10}(MSE) \qquad [16]$$

5. Execution Time: It is the total time taken to execute a program or code completely. Execution time can be found by tic-toc command. Place tic; before the first line of code and toc; after the last line of the code.
6. Noise Sensitivity (NS): This criteria measures how much as skeleton is immune to noise. It is very significant feature of skeletonization algorithms because it determines both topology and shape preservation of the skeleton [14]. NS can be defined as follows:

$$NS=\sum_{x=1}^{n}\sum_{y=1}^{m} N(P[x][y]) \qquad [14]$$

$$N(P[x][y])=\begin{cases}1 & \text{if } CN(P[x][y])>2\\ 0 & \text{otherwise}\end{cases} \qquad [14]$$

Where:

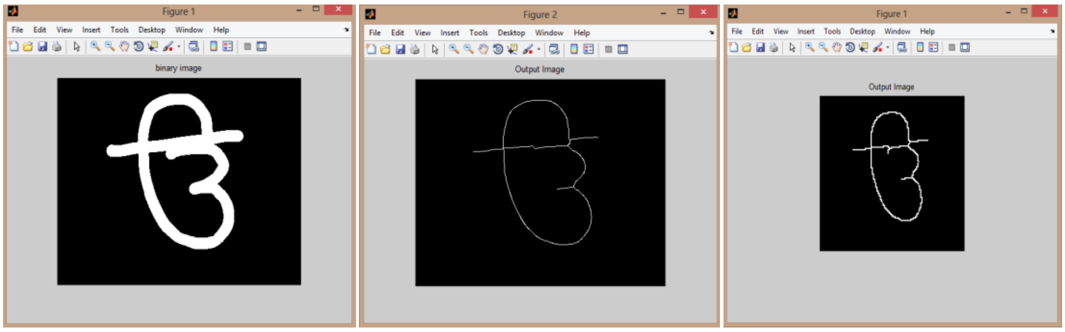
P[x] [y] are black pixels with coordinates x,y

n, m are dimensions of an image

CN is connectivity number which counts the total number of color in neighbourhood of pixel P[x][y] [14].

6. SKELETONIZATION ALGORITHM

In the present paper we are applying Zhang and Suen algorithm , Guo and Hall algorithm in order to measure performance evaluation parameters such as thinning rate, PSNR ,MSE, Execution time and number of connected components etc. Present section describes some of the outputs of Zhang and Suen and Guo and hall algorithm skeletonization is shown in fig a, b, c, d respectively.

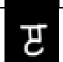
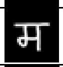
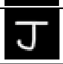
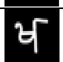
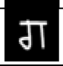
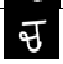
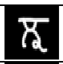
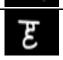


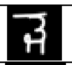
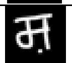
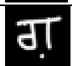
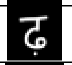
(a) Input Image **(b) ZS Skeletonization** **(c) GH Skeletonization**

7. RESULTS OF ZS IN COMPARISON TO GH ALGORITHM

Present section describes outputs of Zhang and Suen algorithm and Guo and Hall algorithm in terms of following parameters such as: number of connected components, execution time, PSNR, MSE.

Table 2: Performance Evaluation of ZS and GH algorithm

PARAMETERS								
	PSNR		MSE		EXECUTION TIME		NUMBER OF CONNECTED COMPONENTS	
Algo	ZS	GH	ZS	GH	ZS	GH	ZS	GH
	33.90	32.85	33.25	32.88	0.90	45.7	1	1
	32.85	32.81	33.42	32.77	0.91	34.2	1	1
	32.95	32.92	34.91	33.01	0.74	34.2	1	1
	32.94	32.90	32.61	32.64	0.81	34.9	1	1
	32.90	32.85	32.96	32.91	0.68	37.3	1	1
	32.98	32.96	32.98	32.97	0.81	38.3	1	1
	32.96	32.94	32.98	32.80	0.79	32.1	1	1
	33.07	33.04	32.83	33.00	0.76	39.5	1	1
	32.90	32.86	33.19	32.94	0.79	38.1	1	1
	32.90	32.89	32.95	32.94	0.74	39.6	1	1
	32.90	32.90	32.95	32.82	0.83	36.6	1	1
	32.90	32.90	32.84	32.89	0.76	33.2	1	1
	33.09	33.08	32.90	33.58	0.78	31.9	1	1
	33.10	33.08	33.59	32.93	0.76	41.2	1	1
	33.79	33.79	32.97	33.01	0.84	33.7	1	1
	32.80	32.79	33.05	32.81	0.84	38.7	1	1
	32.91	32.87	32.83	32.93	0.76	37.0	1	1

	32.93	32.92	32.88	32.87	0.80	35.5	1	1
	32.94	32.96	32.90	32.89	0.78	38.7	1	1
	32.92	32.90	32.94	32.92	0.66	32.33	1	1
	32.84	32.82	32.85	32.83	0.67	40.17	1	1
	32.87	32.85	32.96	32.95	0.71	40.16	1	1
	32.87	32.85	32.79	32.78	0.70	37.93	1	1
	32.88	32.84	33.01	32.98	0.77	36.07	1	1
	32.91	32.90	32.89	32.83	0.76	30.48	1	1
	33.03	33.03	32.78	32.76	0.69	28.67	1	1
	32.95	32.92	32.92	32.90	0.69	27.57	1	1
	32.86	32.86	32.97	32.94	0.70	31.29	1	1
	32.86	32.84	32.78	32.77	0.71	33.02	1	1
	32.90	32.91	32.92	32.89	0.68	33.18	1	1
	32.85	32.84	32.87	32.84	0.68	30.93	1	1
	33.12	33.10	32.85	32.82	0.71	25.50	1	1
	33.03	33.01	32.80	32.78	0.69	32.77	1	1
	32.96	32.94	32.78	32.75	0.72	29.92	1	1
	32.95	32.96	32.89	32.87	0.70	33.35	1	1
	32.90	32.89	32.90	32.85	0.74	34.78	2	2
	32.97	32.96	32.77	32.75	0.68	32.07	2	2
	32.94	32.92	32.90	32.86	0.69	28.84	2	2
	32.94	32.93	32.85	32.82	0.68	38.47	2	2
	32.98	32.97	32.95	32.91	0.74	25.06	2	2
	32.87	32.85	32.88	32.78	0.74	32.00	2	2

From the above results, we can conclude that Zhang and Suen algorithm is better than Guo and Hall algorithm in terms of connectivity, mean square error, peek signal to noise ratio and execution time.

8. CONCLUSION

Huge number of skeletonization algorithms has been proposed by different authors till now but due to some complicated nature of skeletonization sometimes it is difficult to understand that how these different approaches are related to one another in terms of the algorithms processing quality and execution time. So far we have discussed Zhang and Suen and Guo and Hall algorithm in terms of connectivity, PSNR, MSE and execution time. Experimental results show that ZS modified is better in terms of Peak Signal to Noise Ratio (PSNR), Mean Square Error(MSE) , Connectivity and execution time than GH algorithm.

In future, Results can be tested on other datasets also. Conditions for noise robustness can be included. Possibility of using neural networks for performing skeletonization can be explored.

9. REFERENCES

[1] Abu-Ain W., Bataineh B., Abu-Ain T. and OmarK., "Skeletonization Algorithm for Binary Images", Fourth International Conference on Electrical Engineering and Informatics(ICEEI) Elsevier, Vol. 11, pp.704-709, 2013.

[2] Padole G.V. and Pokle S. B., "New Iterative Algorithms for Thinning Binary Images" IEEE Third International Conference on Emerging Trends in Engineering and Technology, Vol. 7, pp. 166-171, 2010.

[3] Lam L, Lee S.W. and Suen C.Y., "Thinning methodologies-A comprehensive survey", IEEE transactions on pattern analysis and machine intelligence, Vol. 14, No. 9, pp. 869-885, 1992.

[4] Chatbri H. and Kameyama K., "Using Scale Space Filtering to Make Thinning Algorithms Robust Against Noise in Sketch Images", International Conference on Pattern Recognition letters Elsevier, Vol. 42, pp. 1-10, 2014.

[5] Zhang T.Y. and Suen C.Y., "A Fast Parallel Algorithm for Thinning Digital Patterns", Communications of the Association of Computer Machinery (ACM), Vol. 27, No. 3, pp. 236-239, 1984.

[6] Zhou R.W., Quek C. and Ng G.S., "A Novel Single-Pass Thinning Algorithm and an Effective Set of Performance Criteria", International Journal of Pattern Recognition Letters Elsevier, Vol. 16, Issue 12, pp. 1267-1275, 1995.

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

[8] Rockett P.I., "An Improved Rotation-Invariant Thinning Algorithm", IEEE Journal on Pattern Analysis and Machine Intelligence, Vol. 27, No. 10, pp.1671-1674, 2005.

[9] Huang L., Wan G. and Liu C., "An Improved Parallel Thinning Algorithm", IEEE Seventh International Conference on Document Analysis and Recognition, Vol. 10, pp. 780-783, 2003.

[10] Saeed K., Tabedzki M., Rybnik M. and Adamski M., "K3M: A Universal Algorithm for Image Skeletonization and A Review of Thinning Techniques", International Journal of Applied Mathematics & Computer Science, Vol. 20, No. 2, pp. 317–335, 2010

[11] Jagna A. and Kamakshiprasad V., "New parallel binary image thinning algorithm" ARPN Journal of Engineering and Applied sciences, Vol. 5, No. 4, pp. 64-67, 2010.

[12] Choudhary A., Rishi R. and Ahlawat S., "Off-Line Handwritten Character Recognition using Features Extracted from Binarization Technique", American Applied Science Research Institute (AASRI) Conference on Intelligent Systems and Control, Vol. 4, pp. 306-312, 2013.

[13] Tarabek P., "Performance Measurements of Thinning Algorithms", Journal of Information, Control and Management Systems, Vol. 6, No.2, pp. 125-132, 2008.

[14] Lin X. et al. " A proof of image Euler number formula" Springer June 2006, Volume 49, Issue 3, pp 364-371

[15] Guo Z. and Hall R.W., "Parallel Thinning with Two-Sub Iteration Algorithms", Communications of the Association of Computer Machinery (ACM) Image Processing and Computer Vision, Vol. 32, No. 3, pp. 359-373, 1989.

[16] Kwon J., "Improved Parallel Thinning Algorithm to Obtain Unit -Width Skeleton", The International Journal of Multimedia & Its Applications (IJMA), Vol.5, No.2, pp. 1-14, 2013.