Comparing Thinning Based Morphological operation on Sindhi Printed Dataset.

**Nasreen Nizamani†, Bhawani Shankar Chowdhry††, Mujtaba Shaikh††† Abdul Sattar Saand†††
Ehsan Ali†††**

†PhD student at Department of Electronic Engineering and ††Dean faculty of FEECE, MUET, Jamshoro Pakistan
†††Department of Electronic Engineering, QUEST, Nawabshah, Pakistan
††††Department of Electrical Engineering, QUEST, Nawabshah, Pakistan

Summary
Thinning is an essential step in many character recognition techniques. Thinning is a process, used at a pre-processing stage for recognizing character. In this paper Z.-Suen and H-W-Lui algorithms are enhanced to Sindhi Optical Character Recognition (OCR). Using aforementioned algorithms the thinning of 4704 Sindhi characters is analyzed. The four thinning algorithm are selected and applied on Sindhi characters data set. The obtained results are analyzed based on the shape and text connectivity. The performance is measure on the basis of text connectivity, and calculating the processing time.

Key words:
Sindhi OCR, Thinning process, Sindhi character.

1. Introduction

In optical character recognition (OCR) the extraction of thinned images is most often used as a pre-processing step for recognition system. Thinning is a process to reduce the components of the character image. In the process if thinning several pixel character thickness is reduced into single pixel shape. The thinning of a set A by a structuring element B is defined by an operator $\otimes$ as shown in Fig.1.

$C = A \otimes B = A - (A \otimes B)$ where $A \otimes B$ is Hit or Miss transform, $\{B\} = \{B^1, B^2, B^3 .....B^n\}$

$A \otimes \{B\} = ((.....((A \otimes B') \otimes B^2).....) \otimes B^n)$

Thinning process is utilized in many applications for the recognition of text, finger print analysis and chromosome [1]-[4]. it is widely used in OCR system. Generally there are few steps in OCR process, pre-processing, feature extraction and classification. The thinning was broadly utilized to support any feature at extraction and classification stage [5]. Numerous thinning methods have been suggested in [3] for different objectives. Z.-Suen thinning algorithm is based on morphological operation and skeletonization of morphological operation and H.-W.-Lui algorithms are to apply for the thinning on Sindhi OCR and to analyze its impact on printed dataset. In this work, two main problems addressed are; the execution time of algorithm and the shape of the thinned image considering these four methods (see Appendix).

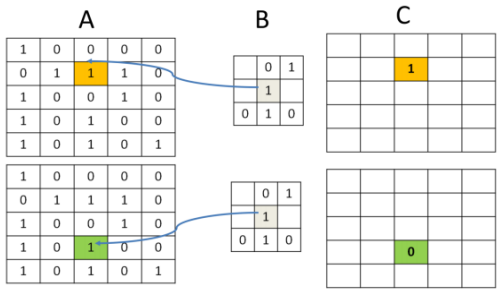


Fig. 1 Thinning hit-or-miss operation.

The paper is organized as follows. The details of a thinning algorithm are presented in section 2. The results obtained by using four selected algorithm are presented in section 3 which also contains a comparison of these thinning algorithms. The conclusion is presented in Section 4.

2. Thinning Methods

t_5	t_6	t_7
t_4	P	t_0
t_3	t_2	t_1

Fig. 2: A point mentioned as P and its neighbors. (t0 ,.....,t7)

In this section, at first the four different thinning methods are discussed.

2.1 Z.-Suen Thinning Algorithm

This algorithm is included into parallel thinning algorithm [7] in which new pixels values are the function of only previous iterations and the whole point representing the image are processed simultaneously. First of all the thinning method of Z.-Suen (see algorithm1 in

Manuscript received February 5, 2018
Manuscript revised February 20, 2018

APPENDIX) is applied for the thinning of the Sindhi character. For an input image the black pixels are describing the shape whereas the white pixels represent the background. So that: $(x, y), (x+1, y), (x, y-1), ...(x, y+1)$ represents the set of 4 neighbors, and the 8 pixels in the 3x3 window as shown in Fig.2.

2.2 Thinning based morphological operation (TMBO)

The TMBO method is selected for the thinning of the Sindhi character images. The procedure to apply thinning based morphological operation is named algorithm2 as given in APPENDIX.

2.3. Skeletonization based morphological operation (SBMO)

This method extracts the skeleton of Sindhi character images by using algorithm3 shown in APPENDIX.

2.4. Huang Wan Liu (H.-W.Liu) algorithm

H.-W.-Liu presented the parallel thinning algorithm in [3] which is executed in 2 pass modes. The principle of this algorithm is similar to other thinning methods, it preserve the connectivity by taking into account the unique pixel neighbor, as 3x4, 4x3 and 4x4 windows are shown in Fig.3. Algorithm4 given in APPENDIX is presented by H.-W.-Liu.

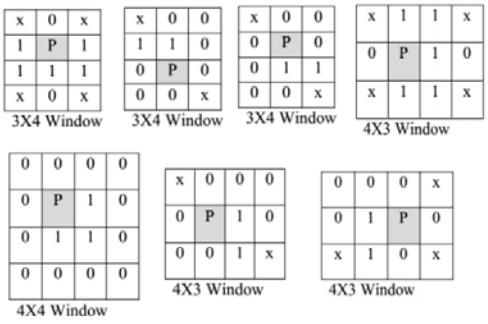


Fig. 3 The marked points shows the connectivity.

3. Experimental Results

The Z.-Suen, TBMO, SBMO and H-W-Lui thinning algorithms have been applied on Sindhi printed dataset. Fig.4 shows the result of the thinning operation performed on an image (i.e. sindhi character pronounced as Bha) when these four methods are applied. It can be said that, to effectively perform thinning method on Sindhi text may need a few requirements like Checking text connectivity, preserving shape information, preserving dots and speed efficiency.

A) Checking text connectivity: The text connectivity can be measured by looking at the components connected in both images (the original and thinned image). If the connected components are same in both(original and thinned) are equal in numbers, then it maintains the text connectivity otherwise, no connectivity can be considered. In Fig. 4, It can be seen that H.-W.-Liu method has generated thinning of the image that preserves the connectivity of its components. On the other hand Z.-Suen, TBMO& SBMO algorithms don't contain the similar number of connected components. Fig.5 shows that, H.-W.-Lui is very efficient to preserve the 100% text connectivity of Sindhi printed dataset.

Table 1: Using four methods connectivity of images is represented.

Thinning Method	Preserved the text connectivity out of 4704 images	Performance (%)
Z.-Suen	3690	78.4%
TBMO	4222	89.75%
SBMO	4226	89.83%
H.-W.-Liu	4704	100%

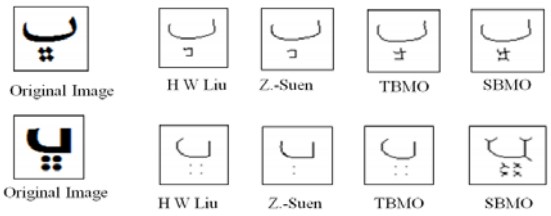


Fig. 4 Obtained thinned image using four algorithms

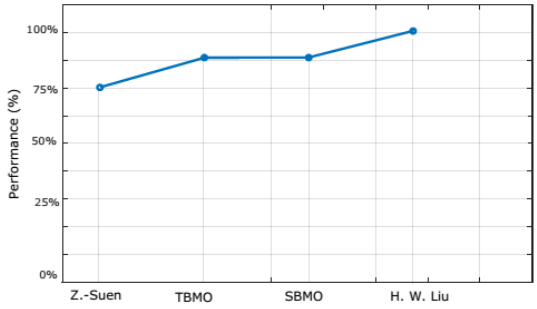


Fig. 5 The performance chart of measuring connectivity by applying four thinning methods on the Sindhi printed data set.

B) Speed

The time (in seconds) is an essential parameter to run any algorithms, so that execution efficiency of any algorithm can be determined. The overall processing time taken by these four algorithms for thinning of 4704 images of Sindhi printed dataset are as given in table.3.

Table 2: Comparing the execution time taken by these four thinning methods for dataset of 4704.

Thinning methods	Z.-Suen	TBMO	SBMO	H. W. Liu
Time (s)	230	317	367	620

From Tab.2, it is clear that, Z.-Suen thinning method is faster than other method and H. W Liu method is slower among these four thinning methods.

4. Conclusion

In this paper, the four thinning methods (Z.-Suen, TBMO, SBMO, H. W. Liu) were selected and applied on Sindhi printed dataset. The results are obtained based on the following criteria; check text connectivity and execution speed. The H. W. Liu's method produces the best thinned images comparing other methods(Z.-Suen, TBMO, and SBMO). It produces 100% skeleton that preserve the connectivity for Sindhi printed dataset. Z.-Suen thinning algorithm is faster among other methods, however, it lacks in preserving the connectivity, therefore, H. W-Liu method is more effective for thinning of Sindhi printed dataset. In Future work, measurement of effectiveness of thinning methods that can perform other requirements; stroke thickness, preventing spurious tails and avoiding necking problem will be discussed

Acknowledgment

The authors would like to express their cordial thanks to Dr. Ahsan Ahmad Ursani for his guidance, and Quaid-E-Awan University of Engineering, Since and Technology Nawabshah for providing funds to carry out this research.

References

[1] Al-Shatnawi AM, Omar K, Zeki AM. Challenges in thinning of arabic text. In ICGST International Conference on Artificial Intelligence and Machine Learning (AIML-11), Dubai. UAE, pp. 127-133, 2011

[2] Melhi, M., Stanley S. Ipson, and William B. A novel triangulation procedure for thinning hand-written text. Pattern Recognition Letters 22.10): pp.1059-1071, (2001

[3] Lam, Louisa, Seong-Whan Lee, and Ching Y. Suen. "Thinning methodologies-a comprehensive survey." IEEE Transactions on pattern analysis and machine intelligence 14.9, pp. 869-885, (1992):

[4] J Al-Shatnawi, Atallah M., et al. Skeleton extraction: Comparison of five methods on the Arabic IFN/ENIT database." Computer Science and Information Technology (CSIT), 2014 6th International Conference on. IEEE, 2014.

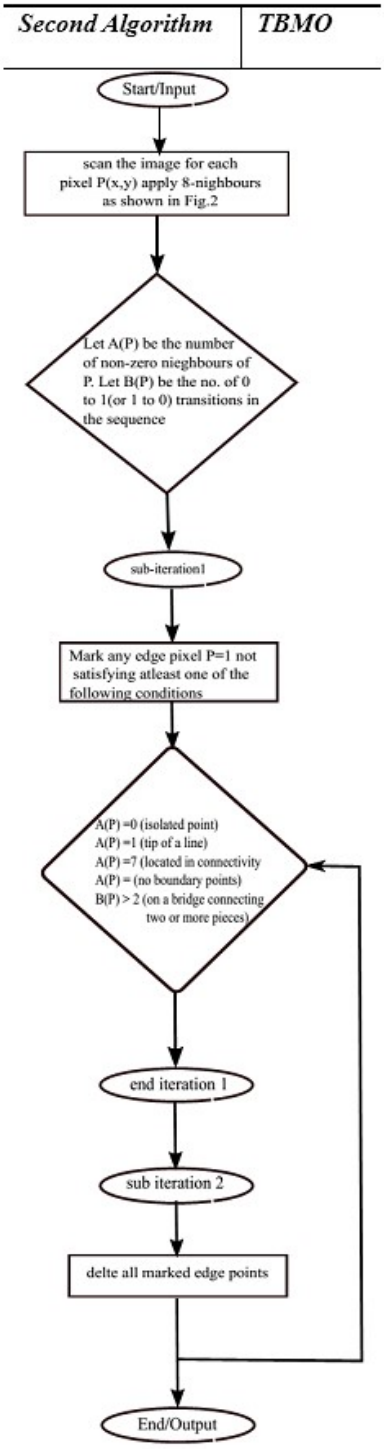
[5] Mori S, Nishida H, Yamada H. Optical character recognition. John Wiley & Sons, Inc.; 1999

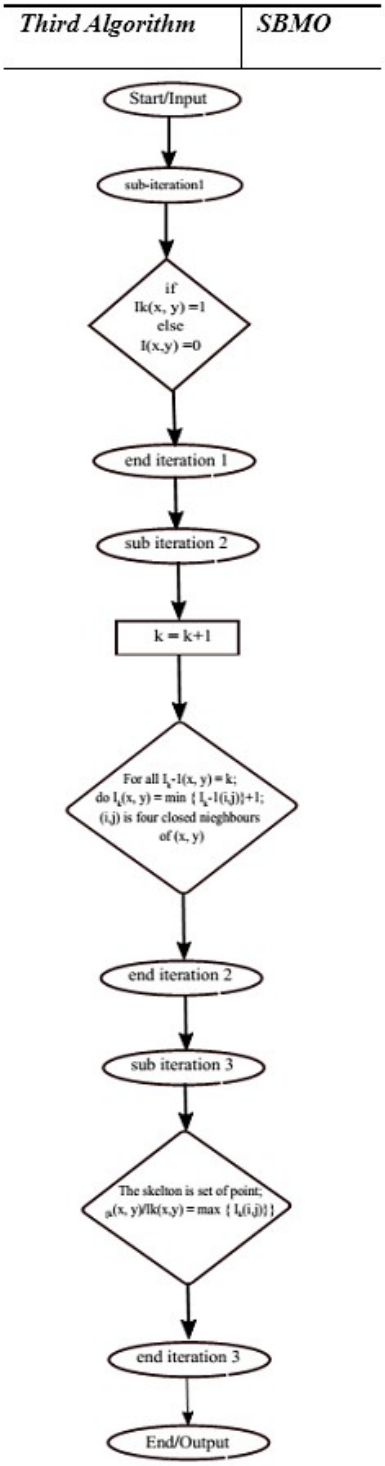
[6] Cowell, John, and Fiaz Hussain, Thinning Arabic characters for feature extraction, In Information Visualisation. Proceedings. Fifth International Conference on, pp. 181-185. IEEE, 2001.

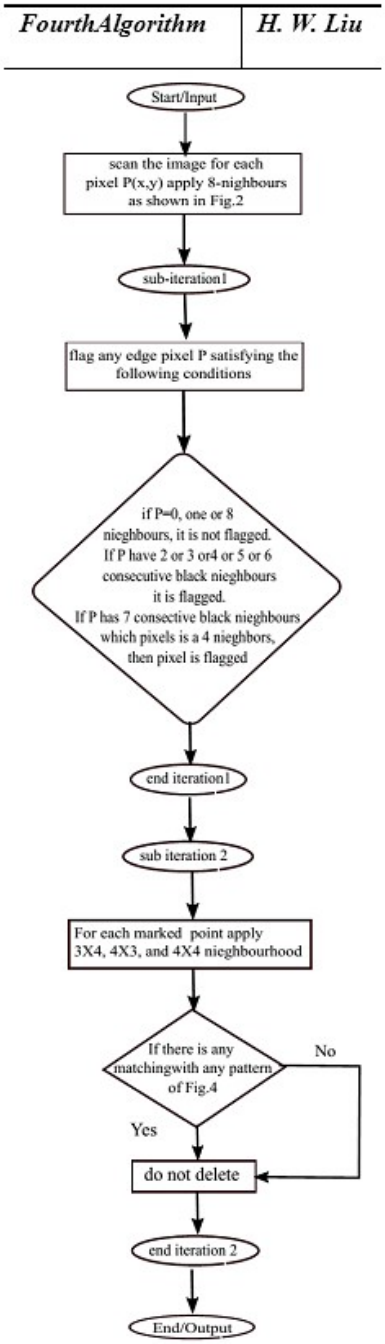
[7] Zhang, T. Y., and Ching Y. Suen. "A fast parallel algorithm for thinning digital patterns." Communications of the ACM 27, no. 3, pp. 236-239, (1984).

[8] L. Huang, G. Wan and C.Liu, An improved Parallel Thinning Algorithm. Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), (2003), 780-783

APPENDIX


Second Algorithm
TBMO
Start/Input
scan the image for each pixel P(x,y) apply 8-nighbours as shown in Fig.2
Let A(P) be the number of non-zero nieghbours of P. Let B(P) be the no. of 0 to 1(or 1 to 0) transitions in the sequence
sub-iteration1
Mark any edge pixel P=1 not satisfying atleast one of the following conditions
A(P) =0 (isolated point)
A(P) =1 (tip of a line)
A(P) =7 (located in connectivity
A(P) = (no boundary points)
B(P) > 2 (on a bridge connecting two or more pieces)
end iteration 1
sub iteration 2
delte all marked edge points
End/Output


Third Algorithm
SBMO
Start/Input
sub-iteration1
if Ik(x, y) =1 else I(x,y) =0
end iteration 1
sub iteration 2
k = k+1
For all Ik-1(x, y) = k; do Ik(x, y) = min { Ik-1(i,j)}+1; (i,j) is four closed nieghbours of (x, y)
end iteration 2
sub iteration 3
The skelton is set of point; Sk(x, y)/Ik(x,y) = max { Ik(i,j)}
end iteration 3
End/Output


FourthAlgorithm
H. W. Liu
Start/Input
scan the image for each pixel P(x,y) apply 8-nighbours as shown in Fig.2
sub-iteration1
flag any edge pixel P satisfying the following conditions
if P=0, one or 8 nieghbours, it is not flagged.
If P have 2 or 3 or4 or 5 or 6 consecutive black nieghbours it is flagged.
If P has 7 consective black nieghbours which pixels is a 4 nieghbors, then pixel is flagged
end iteration1
sub iteration 2
For each marked point apply 3X4, 4X3, and 4X4 nieghbourhood
If there is any matchingwith any pattern of Fig.4
No
Yes
do not delete
end iteration 2
End/Output