Thinning: A Preprocessing Technique for an OCR System for the Brahmi Script

H. K. Anasuya Devi*

Abstract

In this paper we study the methodology employed for preprocessing the archaeological images. We present the various algorithms used in the low-level processing stage of image analysis for Optical Character Recognition System for Brahmi Script. The image preprocessing technique covered in this paper include Thinning method. We also try to analyze the results obtained by the pixel-level processing algorithms.

1. Introduction

Optical scanning of the rock inscription yields an image (file of pixels) that forms the raw input to the Optical Character Recognition System. The output is the set of recognized characters.

Preprocessing is the first phase of document analysis. The purpose of preprocessing is to improve the quality of the image being processed. It makes the subsequent phases of image processing like recognition of characters easier. Thinning is one of the preprocessing methods discussed in this paper.

In thinning, the image regions are reduced to one-pixel width characters.

2. Thinning

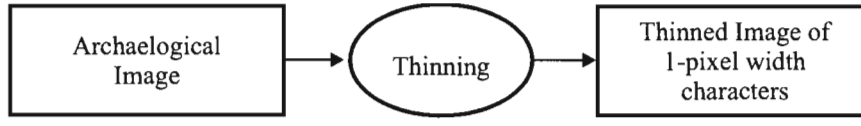

Fig. 1: The process of thinning along with its inputs and outputs

*Fellow, National Institute of Advanced Studies IISc Campus, Bangalore-12
(hka@nias.iisc.ernet.in)

2.1 Definition

Thinning is an image preprocessing operation performed to make the image crisper by reducing the binary-valued image regions to lines that approximate the skeletons of the region.

2.2 Purpose

Thinning cleans the image so that only reduced amount of data needs to be processed in the next image processing stage. Shape analysis could be done easily.

2.3 Thinning algorithms

Thinning algorithms should perform thinning effectively by successive deletion of dark points (i.e. changing them to white points) along the edges of the pattern until it is thinned to a line.

An effective thinning algorithm is one that can ideally compress data, eliminate local noise without introducing distortions of it own. But the key goal is to retain significant features of the pattern. There are two types of thinning algorithms

1. Sequential thinning algorithms
2. Parallel thinning algorithms

In [1], result of n^{th} iteration depends on result of $(n-1)^{th}$ iteration as well as pixels already processed in the n^{th} iteration.

In [2], deletion of pixels in of n^{th} iteration depends only on the result that remains after $(n-1)^{th}$ iteration. We consider only the type [2] algorithms here.

2.3.1 ZS Thinning algorithm

P1	**P2**	**P3**
P8	**Pi**	**P4**
P7	**P6**	**P5**

Fig. 2: 3x3 pixel window under consideration

Method: This algorithm performs sub-iteration twice.

In first step, the pixels satisfying following conditions are erased

1. All pixels whose number of value 1 in 8-neighbour pixels are in the range 2 to 6
2. The number of zero to one patterns in 8-neighbour pixels is one
3. P2 * P4 * P6 = 0
4. P4 * P6 * P8 = 0

In second step, condition [3] and [4] is replaced with

3^{1} P2 * P4 * P8 =0
4^{1} P2 * P6 * P8=0

Performance: Loses pixels in slanting lines. Slanting lines with 2-pixel width are erased.

2.3.2 LW Thinning algorithm

Method: Only the condition [1] of ZS is replaced by a condition

1. All pixels whose number of value 1 in 8-neighbour pixels are in the range 3 to 6

Performance: It can shrink horizontal lines well, but doesn't have one pixel width in slanting lines.

2.3.3 WHF Thinning algorithm

Method: A new condition is added to LW algorithm

$$(P3 \oplus P5) \,.\, (P2 \oplus P6) \,.\, P4 \,.\, P8 = 1$$

Performance: It can make one pixel width but also makes needless tress sometimes.

2.3.4 Enhanced Parallel Thinning Algorithm

Method: Condition 1 has step one and two same as ZS algorithm. A new condition is added which following steps:

1. P1 * P8 * P6 = 1 & P3=0
2. P3 * P4 * P6 = 1 & P1=0
3. P5 * P6 * P8 = 1 & P3=0
4. P4 * P6 * P7 = 1 & P1=0

Condition 1 used to erase all pixels in the input image. Condition 2 finds to erase the remaining lines with 2-pixel width.

Performance: It can extract 1-pixel slim lines. It can also settle the end point shrinking phenomenon and has perfect 8-connectivity.

2.3.5 Arabic Parallel Thinning Algorithm

In each pass, the input pattern is scanned from the upper left hand corner to the lower right hand corner.

A dark point p1 is flagged if at least one of each sub-iteration is satisfied.

Method: Each of the sub-iterations is divided into 4 conditions. First and second conditions are same for all sub-iterations which is

1. All pixels whose number of value 1 in 8-neighbour pixel are in the range 2 to 6
2. The number of zero to one pattern in 8-neighbour pixels is one.

First sub-iteration

3. P2 * P4 * P6 = 0
4. P4 * P6 * P8 = 0

Second sub-iteration

3. P2 * P6 * P8 = 0
4. P4 * P6 * P8 = 0

Third sub-iteration

3. P2 * P4 * P8 = 0
4. P2 * P6 * P8 = 0

Fourth sub-iteration

3. P2 * P4 * P6 = 0
4. P2 * P4 * P8 = 0

Performance: Preserves shape of original image well

2.3.6 Matching algorithm

Method: It utilizes 8 templates(shown below) and 2 images i.e. the current image and a working image (of the same size which is updated when templates are matched).

0 0 ★	★ 0 0	★ 1 ★	★ 1 ★
0 1 1	1 1 0	1 1 0	0 1 1
★ 1 ★	★ 1 ★	★ 1 0	0 0 ★
A1	A2	A3	A4
0 0 0	1 ★ 0	★ 1 1	0 ★ ★
★ 1 ★	1 1 0	★ 1 ★	0 1 1
1 1 ★	★ ★ 0	0 0 0	0 ★ 1
B1	B2	B3	B4

Fig. 3: Templates for matching

Pixels having a value 1 are removed by comparing each pixel and its neighbors in the current image with a set of templates. The algorithm works as follows:

1) Initially the current image and the working image are identical copies of the original input image.
2) Template a1 is compared with all pixels having a value 1 and their neighbors in the current image.
3) If the match is obtained, then the central pixel is removed in the working image.
4) After preprocessing with template a1, the current image is discarded and the working image becomes the new current image, and the new working image is obtained by copying the new current image.
5) The process is repeated with templates A2, A3,...,B4 forming a complete cycle until no more pixels are removed (i.e. the skeleton is obtained).

Performance: Preserves shape of original image well and preserves connectivity.

3. Results

Fig. 4: An input image before thinning (Pedestal)

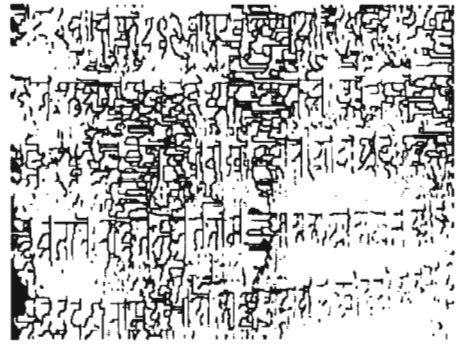

Fig. 5: The output image as a result of applying ZS thinning algorithm to Fig.4

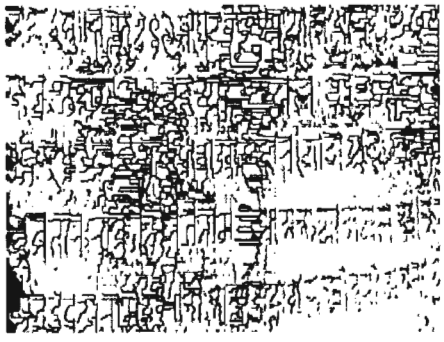

Fig. 6: The output image as a result of applying LW thinning algorithm to Fig. 4

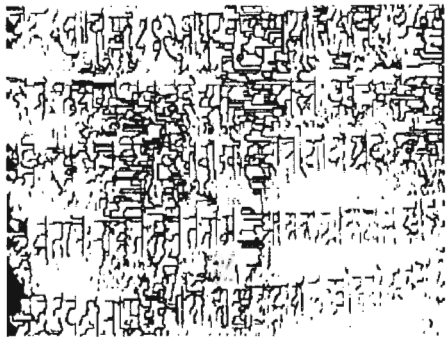

Fig. 7: The output image as a result of applying Enhanced Thinning Algorithm to Fig. 4

4. Conclusions and Future Enhancement

The preprocessing algorithms discussed so far give considerably good results. A cascaded approach wherein various thinning and thresholding algorithms are successively applied on the input image can yield better results. Hybrid preprocessing algorithms can be tried out wherein new methods can be designed to perform effective thinning and thresholding. Preprocessing techniques like Filtering (to remove distortions and noise) could be incorporated.

Acknowledgements

The author wishes to thank Mr. Bipin Suresh, Ms. Adithi Sampath, Ms. Dimple Kolhapure, Mr. Prasanna Venkatesh and Mr. Santosh Kabbur for their contribution during the execution of the program.

References

Thinning Algorithms for Arabic OCR (M. Tellache, M.A.Sid Ahmed & B.Abaza) Communications, Computers and Signal Processing, 1993., *IEEE Pacific Rim Conference* on , Volume: 1 , 19-21 May 1993 Pages:248 - 251 vol.1

An enhanced thinning algorithm using parallel processing (Jun-Sik Kwon., Jun- Woong Gi' and Eung-Kwan Kang) *Image Processing, 2001. Proceedings. 2001* International Conference, Volume: 3 , 7-10 Oct. 2001 Pages:752 - 755 vol.3

Thinning Methodologies -A Comprehensive Survey (Louis Lam, Seong - Whan Lee & Ching Y. Suen)

Document Image Analysis by Rangachar Kasturi