

# Thresholding: A Pixel-Level Image Processing Methodology 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 is thresholding. 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. Thresholding is one of the preprocessing methods discussed in this paper.

In thresholding, the color-image or gray-scale image is reduced to a binary image.

## 2. Thresholding



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

---

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

## 2.1 Definition

Thresholding is a process of converting a grayscale input image to a bi-level image by using an optimal threshold.

## 2.2 Purpose

The purpose of thresholding is to extract those pixels from some image which represent an *object* (either text or other line image data such as graphs, maps). Though the information is binary the pixels represent a range of intensities. Thus the objective of binarization is to mark pixels that belong to true foreground regions with a single intensity and background regions with different intensities.

## 2.3 Thresholding algorithms

For a thresholding algorithm to be really effective, it should preserve logical and semantic content. There are two types of thresholding algorithms

- 1 Global thresholding algorithms
- 2 Local or adaptive thresholding algorithms

In global thresholding, a single threshold for all the image pixels is used. When the pixel values of the components and that of background are fairly consistent in their respective values over the entire image, global thresholding could be used.

In adaptive thresholding, different threshold values for different local areas are used.

### 2.3.1 Quadratic Integral Ratio (QIR) algorithm

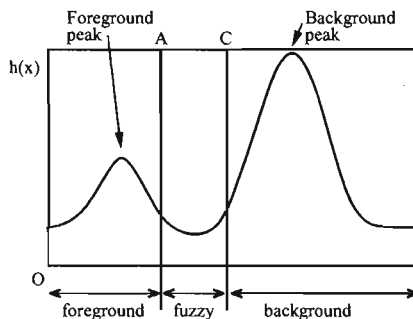


Fig. 2: Three sub images of QIR method

**Method:** QIR is a global two stage thresholding technique that uses intensity histogram to find the threshold.

The first stage of the algorithm divides an image into three subimages: foreground, background, and a fuzzy subimage where it is hard to determine whether a pixel actually

belongs to the foreground or the Background. Two important parameters that separate the subimages are  $A$ , which separates the foreground and the fuzzy subimage, and  $C$ , which separate the fuzzy and the background subimage. If a pixel's intensity is less than or equal to  $A$ , the pixel belongs to the foreground. If a pixel's intensity is greater than or equal to  $C$ , the pixel belongs to the background. If a pixel has an intensity value between  $A$  and  $C$ , it belongs to the fuzzy subimage and more information is needed from the image to decide whether it actually belongs to the foreground or the background.

The strategy is to eliminate all pixels with intensity level in  $[0,A]$  and  $[C,255]$ . Thus produce a range of promising threshold values delimited by the parameter  $A$  and  $C$  ( $T \in [A,C]$ ).

**Performance (with respect to our experiments):** QIR performed well as it generally was able to separate definite foreground (dark) pixels and definite (background pixels). The uncertain or fuzzy pixels were clearly defined and required further processing to determine appropriate assignment to background or foreground.

### 2.3.2 OTSU algorithm

**Method:** This type of thresholding is global thresholding. It stores the intensities of the pixels in an array. The threshold is calculated by using total mean and variance. Based on this threshold value each pixel is set to either 0 or 1. i.e background or foreground. Thus here the change of image takes place only once.

The following formulas are used to calculate the total mean and variance.

The pixels are divided into 2 classes,  $C_1$  with gray levels  $[1, \dots, t]$  and  $C_2$  with gray levels  $[t+1, \dots, L]$ .

The probability distribution for the two classes is:

$$C_1: p_1/w_1(t), \dots, p_t/w_1(t) \text{ and}$$

$$C_2: p_{t+1}/w_2(t), \dots, p_L/w_2(t)$$

$$\text{Where } w_1(t) = \sum_{i=1}^t p_i$$

$$\text{and } w_2(t) = \sum_{i=t+1}^L p_i$$

Also, the means for the two classes are

$$\mu_1 = \sum_{i=1}^t ip_i/w_1(t)$$

and

$$\mu_2 = \sum_{i=t+1}^L ip_i/w_2(t)$$

Using Discriminant Analysis, Otsu defined the between-class variance of the thresholded image as

$$\sigma_B^2 = w_1 (\mu_1 - \mu_T)^2 + w_2 (\mu_2 - \mu_T)^2.$$

For bi-level thresholding, Otsu verified that the optimal threshold  $t^*$  is chosen so that the between-class variance  $\sigma_B$  is maximized; that is,

$$t^* = \underset{1 \leq t \leq L}{\text{Arg Max}} \{ \sigma_B^2(t) \}$$

**Performance (with respect to our experiments):** Otsu works well with some images and performs badly with some. The majority of the results from Otsu have too much of noise in the form of the background being detected as foreground. Otsu can be used for thresholding if the noise removal and character recognition implementations are really good. The main advantage is the simplicity of calculation of the threshold. Since it is a global algorithm it is well suited only for the images with equal intensities. This might not give a good result for the images with lots of variation in the intensities of pixels.

### 3. Results



Fig. 3: An input image before thresholding (Pedestal)

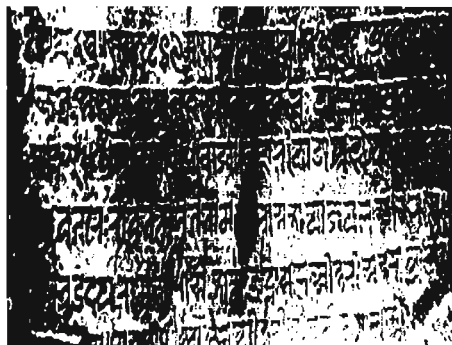


Fig. 4: The output image as a result of applying QIR thresholding algorithm to Fig.3

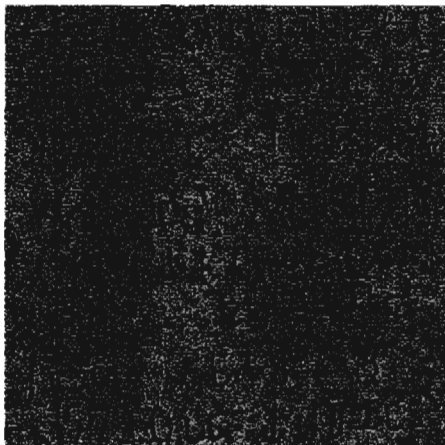


Fig. 5: An input image before thresholding

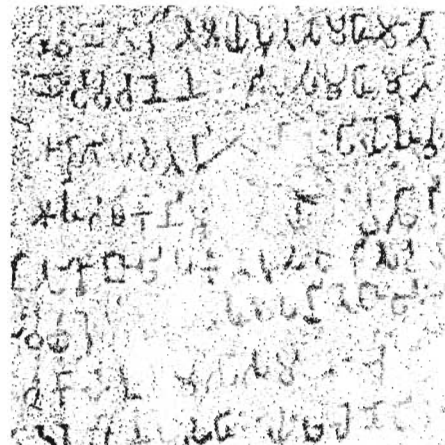


Fig. 6: The output image of applying OTSU algorithm to Fig. 5

#### **4. Conclusions and Future Enhancement**

The preprocessing algorithms discussed so far give fairly average 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 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. Anitha J., Ms. Dimple Kolhapure, Mr. Prasanna Venkatesh and Mr. Santosh Kabbur for their contribution during the execution of the program.

#### **References**

- The Multi-stage Approach to Grey-Scale Image Thresholding for Specific Applications*, Yan Solihin and C.G. Leedham
- Document Image Analysis* by Rangachar Kasturi, Louis Lam, Seong - Whan Lee & Ching Y. Suen.