Threshold and binarization for document image analysis using otsu’s Algorithm

Dr. Neeraj Bhargava¹, Anchal Kumawat², Dr. Ritu Bhargava³

¹Associate Professor, ²Research Scholar, School of Engineering and Systems Sciences, Department of Computer Science, MDS University, Ajmer India
³Lecturer, Department of Computer Science, Government Women’s Engineering College, Ajmer India

Abstract—This paper deals with a binarization of image using a global threshold. Otsu calculates a global threshold by accepting the existence of two classes, foreground and background pixels, and chooses the threshold that minimizes the interclass variance of the threshold black and white pixels.

Converting a gray scale image to binarize is a common image processing task. In this paper, image processing methods are presented for automatically finding the optimum global threshold value.

Keywords—Binarization, otsu, gray scale image, intensity histogram, global threshold.

I. INTRODUCTION

Image binarization is the process of separation of pixel values into two groups, white as background pixel and black as foreground or object pixel. Thresholding plays a major role in binarization of images. Otsu’s thresholding method is the most commonly used method for image binarization using a global threshold [1].

Otsu's method, named after its inventor Nobuyuki Otsu, is one of the binarization algorithms. Image Binarization is an important and effectual area of image processing and pattern recognition. Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels that either falls in foreground or background. The aim is to find the optimum threshold value where the sum of foreground and background spreads is at its minimum. It is important in image processing to select an adequate threshold of gray level for extracting objects from their background. In an ideal case, the intensity histogram is a bi-modal which has a deep and sharp valley between two peaks representing foreground & background respectively. In such cases, the threshold separating those two classes can be chosen at the bottom of this valley [2].

A. Grayscale Image

A gray scale image is an image in which the value of each pixel is a single sample, that is, it carries only luminance or intensity information. Such sort of image is also known as black and white, are composed exclusively of shades of gray. A gray scale color varies from the weakest intensity level (black) to the strongest intensity level (white) as shown in Fig.1.

![Fig. 1 Gray scale shades](image-url)

In MATLAB, we can convert a truecolor image RGB to grayscale image using the inbuilt function ‘rgb2gray(RGB)’. This function rgb2gray converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance or intensity.

B. Intensity histogram

In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. It is a graph showing the number of pixels in an image at each different intensity value found in that image as in Fig. 2 (c). For an 8-bit gray scale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those gray scale values.

The horizontal axis of the graph represents the intensity variations, while the vertical axis represents the number of pixels in that particular intensity. The graph can be divided into three regions along the horizontal axis. The left side of the horizontal axis represents the pure black and dark areas, the middle represents medium and intermediate gray and the right side represents...
bright and pure white areas. The number of pixels that is captured in each one of these intensity levels is represented along the vertical axis. Therefore, the histogram for a darker image will have the majority of its data points on the left side and center of the graph. On the contrary, the histogram for a very bright image will have most of its data points on the right side and center of the graph.

For a given gray scale image having 8 bits/pixel with L=256 gray levels of intensity, the intensity level histogram is defined by a function $h(g)$, for each intensity level $g \in [1 - L]$, the number of pixels in the image that have intensity equal to $g$ is given by:

$$h(g) = N_g$$  

(1)

Where $N_g$ represents the number of pixels in the image at a given intensity $g$.

In MATLAB, we can calculate the intensity histogram of an image using the inbuilt function ‘imhist(I)’. This function displays a histogram for the intensity image I whose number of bins are specified by the image type. For example, in case of a gray scale image, imhist uses 256 bins as a default value.

II. METHODOLOGY

A. Algorithm

The Otsu’s binarization algorithm consists of the following steps.

1. Read a gray scale image.
2. Calculate image histogram.
3. Select a threshold and referred as t,
   3.1 Calculate foreground variance.
   3.2 Calculate background variance.
4. Calculate Within-Class variance.
5. Repeat steps 3 and 4 for all possible threshold value.
6. Final global threshold $T = \text{threshold in MIN(Within-class variance)}$
7. Binarize Image = gray scale image > T

B. Formulation

Let the pixels of a given gray image be represented in $L$ gray levels $[1, 2, 3, \ldots, L]$. The number of pixels at level $I$ is denoted by $n_I$ and the total number of pixels by $N = n_1 + n_2 + n_3 + \ldots + n_L$.[2]

Now suppose we divide the pixels into two classes $C_b$ and $C_f$ (background and foreground) by a threshold at level $t$; then $C_b$ denotes pixels with levels $[1, 2, 3, \ldots, t]$ and $C_f$ denotes pixels with levels $[t+1, t+2, \ldots, L]$.

The calculations for finding the background and foreground variances for a single threshold $t$ are shown below.

For background pixel class $C_b$,

$$\text{Weight } W_b = \sum_{i=1}^{t} \frac{n_i}{N}$$

(2)

$$\text{Mean } \mu_b = \frac{\sum_{i=1}^{t} i \cdot n_i}{\sum_{i=1}^{t} n_i}$$

(3)

$$\text{Variance } \sigma_b^2 = \frac{\sum_{i=1}^{t} (i - \mu_b)^2 \cdot n_i}{\sum_{i=1}^{t} n_i}$$

(4)

For foreground pixel class $C_f$,

$$\text{Weight } W_f = \sum_{i=t+1}^{L} \frac{n_i}{N}$$

(5)

$$\text{Mean } \mu_f = \frac{\sum_{i=t+1}^{L} i \cdot n_i}{\sum_{i=t+1}^{L} n_i}$$

(6)

$$\text{Variance } \sigma_f^2 = \frac{\sum_{i=t+1}^{L} (i - \mu_f)^2 \cdot n_i}{\sum_{i=t+1}^{L} n_i}$$

(7)

The next step is to calculate the ‘Within-Class Variance’ which is simply the sum of the two variances multiplied by their associated weights.

$$\text{Within Class Variance } \sigma_w^2 = W_b \sigma_b^2 + W_f \sigma_f^2$$

(7)

This final value is the ‘sum of weighted variances’ for the threshold value $t$. This same calculation needs to be performed iteratively for all the possible threshold values $1$ to $L$.

Finally we select a threshold $T$, which has the lowest ‘sum of weighted variances’ to be the final globally selected threshold. All pixels with a level less than $T$ are background, all those with a level greater than or equal to $T$ are foreground.
III. Discussion

A. Analysis of Otsu’s method

The algorithm assumes that the image to be threshold contains two classes of pixels or bi-modal histogram (e.g. foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal.

In global threshold, the threshold value is held constant throughout the image. It also assumes high-intensity pixels are of interest and low intensity pixels are not.

Finding the threshold that minimizes the weighted within-class variance is computationally intensive. A faster approach is to calculate the ‘Between-Class Variance’, which is much faster to calculate. In this approach, the threshold with the maximum between-class variance also has the minimum within-class variance. So, it can also be used for effectively finding the optimum threshold and therefore due to being simpler is a much better and quicker approach to use. It is given by:

\[ B_{\text{Between Class Variance}} = \sum_{k=1}^{K} w_k (\mu_k - \mu_T)^2 \]  

B. Transforming a gray scale image to binary image

A binary image is an image which has only two possible colors. In MATLAB, a binary image has only two values i.e. 0 or 1. A value of 1 (white) for all pixels in the input image with luminance or intensity greater than \( T \) and 0 (black) for all other pixels. Thus the transformation of an input image \( A \) into a binary image \( B \) at a selected threshold \( T \), can be represented as follows:

(a) \( b_{ij} = 1 \) for \( a_{ij} > T \)
(b) \( b_{ij} = 0 \) for \( a_{ij} < T \)

Here \( b_{ij} = 1 \), for the object or foreground pixels and \( b_{ij} = 0 \), for the background pixels.

IV. Results

Several examples of experimental results are shown in Figs. 2-4. Throughout these figures, (a) and (d) are original gray-level image; (b) and (e) are binarize image at threshold \( T \); (c) is the gray-level histogram.

Fig. 2 Application to fingerprint images

Image binarization is an important image preprocessing step in fingerprint recognition. This helps in extracting the minutiae point during the feature extraction process. In the above example, Fig 2. (a) and (d) is a full and partial fingerprint images; (b) and (e) are their respective binarize image.
In the above examples, the binarization clears the background, and preserves all the thin details which are of interest. In Fig. 4 (b) and (e), the binarize image preserve only the area of interest.

V. CONCLUSION
In this paper, we have presented a method for automatically selecting an optimal threshold from a gray-level histogram. From the experimental result we conclude that the binarization process based on this method is best suitable for images with bimodal histogram. Most of the calculated threshold value resides in the valley between two peaks.

The suggested method in this paper may be recommended as the most simple and standard one for automatic threshold selection that can be applied to various practical problems.

REFERENCES