

Exploring Otsu's Method

Sotheanith Sok

Department of Engineering

California State University of Long Beach

Sotheanith.Sok@student.csulb.edu

I. INTRODUCTION

In computer vision, image segmentation is the process of dividing an image into multiple segments and these segments tend to simplify the image which will make it easier to analyze such image for other purposes such as boundary locating, object finding, and many more. One of the simpler types of image segmentation is splitting a gray-scale image between a foreground area and a background area by picking a pixel intensity threshold that separates the two areas. Unfortunately, picking the pixel intensity threshold manually is not efficient as any given threshold can only work for a selective group of images. Thus, a more adaptive approach is necessary and Otsu's method is one of them.

Otsu's method, named after Nobuyuki Otsu, is an algorithm for automatically determining a pixel intensity threshold that separates a background area from a foreground area of any gray-scale image. This is possible under the assumption that there is a bi-modal distribution, with each peak represents a background area or a foreground area, when examining the histogram generated from pixel intensities and pixel counts of a given image [8].

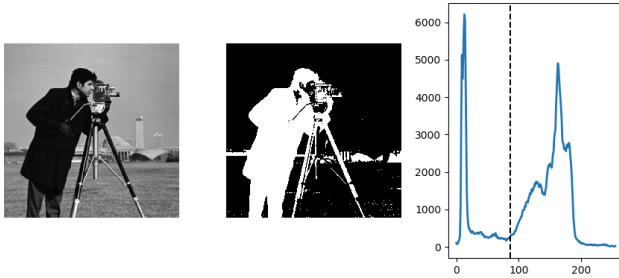


Fig. 1. Original image, binary image, and histogram used or generated by Otsu's method [6].

II. THE ALGORITHM

In order to find the pixel intensity threshold, Otsu's method will perform an exhaustive search of all pixel intensity values and select a value that maximizes the inter-class variance. For any given image, the algorithm starts by generating a histogram representing all pixel intensities and the numbers of pixels that have such intensities [7]. Then, according to [3], an inter-class variance can be represented as

$$\sigma_w^2 = W_f \sigma_f^2 + W_b \sigma_b^2$$

where:

- σ_w^2 is the inter-class variance.
- W_f is the weight of the foreground area.
- σ_f^2 is the variance of the foreground area.
- W_b is the weight of the background area.
- σ_b^2 is variance of the background area.

. Weights of the foreground area and the background area can be calculated with the following formula

$$W_b = \frac{\sum_{i=0}^{t-1} P(i)}{\sum_{i=0}^{all} P(i)}$$

$$W_f = \frac{\sum_{i=t}^{all} P(i)}{\sum_{i=0}^{all} P(i)}$$

where:

- W_b is the weight of the background area.
- W_f is the weight of the foreground area.
- t is the pixel intensity threshold.
- $P(i)$ is the pixel count at intensity i .

The remaining factors, which are variances of both the background area and the foreground area, can be calculated from the mean of each area respectively with the following formula

$$\mu_b = \frac{\sum_{i=0}^{t-1} iP(i)}{\sum_{i=0}^{t-1} P(i)}$$

$$\sigma_b^2 = \frac{\sum_{i=0}^{t-1} (i - \mu_b)^2 \times P(i)}{\sum_{i=0}^{t-1} P(i)}$$

$$\mu_f = \frac{\sum_{i=t}^{all} iP(i)}{\sum_{i=t}^{all} P(i)}$$

$$\sigma_f^2 = \frac{\sum_{i=t}^{all} (i - \mu_f)^2 \times P(i)}{\sum_{i=t}^{all} P(i)}$$

where:

- μ_b is the mean of the background area.
- μ_f is the mean of the foreground area.
- σ_b^2 is the variance of the background area.
- σ_f^2 is the variance of the foreground area.
- t is the intensity threshold.
- $P(i)$ is the pixel count at i intensity.

The entire process is repeated for all pixel intensities and the one that produces the largest inter-class variance will be picked as the final pixel intensity threshold.

III. LIMITATION

While Otsu's method performs well on most gray-scale images, it still has some problems when processing images that do not produce the bi-modal distribution with high peaks and deep valleys [4]. To start with, any image that has a

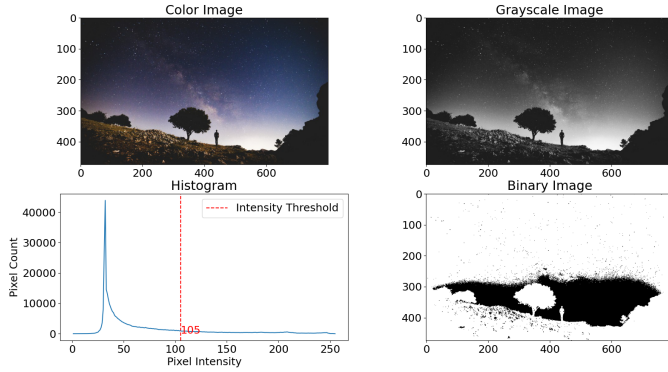


Fig. 2. Otsu's method performance on an image with a small foreground area [1].

foreground area significantly smaller than a background area tends to not do well with Otsu's method due to the lack of pixels count in a foreground area which is often necessary for generating a high peak in the image's histogram. Then, there

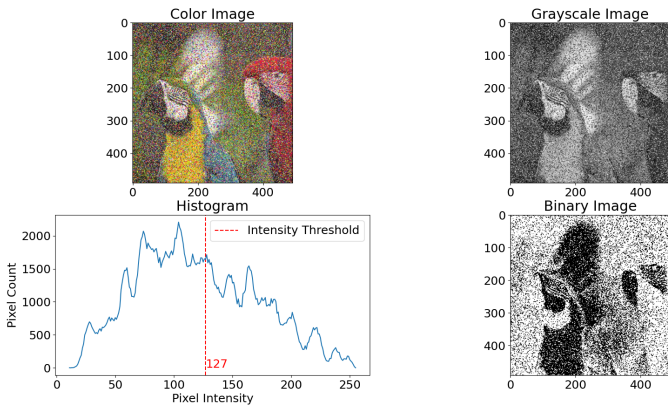


Fig. 3. Otsu's method performance on an image that contains a lot of noise [2].

are images with a lot of noise. This type of image also does not perform well with Otsu's method because noises in such images often prohibit the formation of bi-modal distribution in the histogram. Last but not least, the Otsu's method performs poorly on images with non-uniform illumination because it uses a single global threshold to separate the background area and the foreground area. To simply put, for such an image, a section of the background area might have the same illumination as another section of the foreground area and vice versa.

IV. FINAL THOUGHT

Otsu's method is a great choice for finding a global intensity threshold for splitting a gray-scale image into a foreground

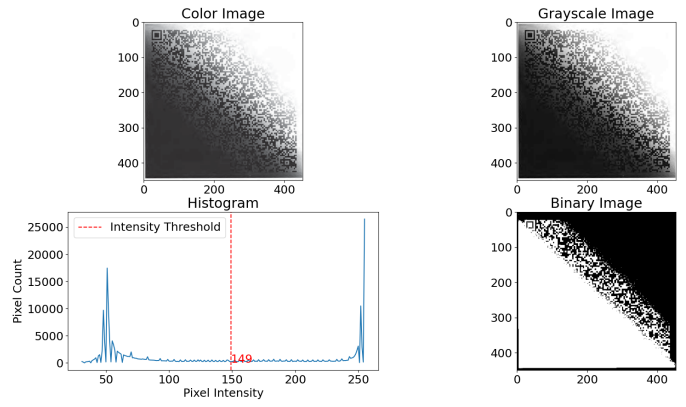


Fig. 4. Otsu's method performance on an image with non-uniform illumination [5].

area and a background area. However, due to the nature of the algorithm which relies heavily on an image's intensities having a bi-modal distribution, it should not be applied universally to all gray-scale images. To overcome the limitations of Otsu's method, preprocessing techniques such as image denoising and others should be used before applying Otsu's method onto an image.

REFERENCES

- [1] Altphotos. *Man Watching the Amazing Starry Sky at Night*. July 2017. URL: <https://altphotos.com/photo/man-watching-the-amazing-starry-sky-at-night-1759/>.
- [2] Rajamani Arumugham et al. "A new denoising approach for the removal of impulse noise from color images and video sequences". In: *Image Analysis and Stereology* 31 (Nov. 2012), pp. 185–191. DOI: 10.5566/ias.v31.p185-191.
- [3] Andrew Greensted. *Otsu Thresholding - The Lab Book Pages*. June 2010. URL: <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>.
- [4] Kang and Atul. *Limitations of otsu method*. July 2019. URL: <https://theailearner.com/tag/limitations-of-otsu-method/>.
- [5] Roman Lishchuk, Volodymyr Kucheruk, and Igor Kurnytnik. "Adaptive binarization with non-uniform image illumination". In: *Pomiary. Automatyka. Robotyka* 6 (June 2014), pp. 72–75. DOI: 10.14313/PAR_208/72.
- [6] Scipy_Lectures. 3.3.9.7. *Otsu Thresholding*. 2020. URL: http://scipy-lectures.org/packages/scikit-image/auto_examples/plot_threshold.html.
- [7] Jian Tay. *Otsu's Method*. Sept. 2020. URL: <https://www.youtube.com/watch?v=jUUKMaNuHP8>.
- [8] Wikipedia_Contributors. *Otsu's method*. Aug. 2020. URL: https://en.wikipedia.org/wiki/Otsu%27s_method.