Adaptive Histogram Thresholing

CIS 465 | Multimedia Programming

Michael Fasko Jr
Department of Electrical Engineering and Computer Science
Cleveland State University
Cleveland, Ohio
mafaskojr@gmail.com

***Abstract*—In many grayscale images, there is usually a significant difference between the gray levels of the focus of the image and the image's background, this could be caused by the photo taken at a different time or day, or other reason for different lighting conditions. By using a statistical thresholding method, it is capable to construct a new, binary image, which can enhance the viewing of two distinct classes. By computing an image histogram, it is possible to segment the image into dark and light pixels, to compute an interclass difference. This report examines the difference between two different image thresholding techniques: Otsu's adaptive thresholding technique, and adaptive progressive thresholding (APT).**

Keywords—Otsu, Adaptive Thresholding, Image Manipulation, Histogram Manipulation, Python

I. Introduction

Information obtained from digital image processing is being used across many industries, and for many different purposes today. Health professionals are using it for conducting medical procedures, as well as being used by artificial intelligence systems, and being used to obtain various analytics [1]. To demonstrate the capabilities of such processing techniques, the Otsu adaptive thresholding technique as well as the adaptive progressive thresholding technique are used to construct grayscale, binary images. Each respective technique recursively can calculate and obtaining a threshold value, for which a new image is constructed, based upon the gray levels of the source image.

Otsu's adaptive thresholding technique is used to compute an 'optimum threshold' value. This value is calculated by computing the inner-class variance, which is found by creating a grayscale image, and using that grayscale image's histogram. For the best effect, it is assumed that the input image has a distinct difference between the focus of the image, and the image's background

Piggybacking off Otsu's algorithm, the APT algorithm utilizes the previously computed optimum threshold value to normalize the original gray image histogram

II. Methodology

The most important part of these adaptive thresholding can find a threshold that can segment the image into two classes, dark and light. Firstly, we look at the Otsu method. However, there is an involved statistical process to do before reaching that point [2]. Firstly, the colored image needs to be converted to a grayscale image, then, then access to the image's histogram is used. Then, for all the pixel values, divide them into two classes, the first method, the first method iteratively calculates the value of the intensity, and multiplies it by that numbers occurrence (1). The second method iteratively takes the value from the first method and uses it to subtract from the total image size (2).

$$q_1(t) = \sum_{i=1}^{t} P(i), \quad (1) \qquad q_2(t) = \sum_{i=t+1}^{I} P(i). \quad (2)$$

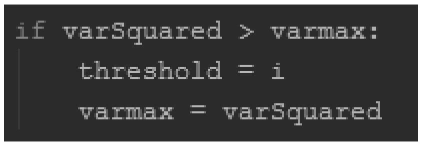
Both methods are then used to calculate their individual values for the means (3) and (4).

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)}, \quad (3) \qquad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}. \quad (4)$$

Then, once the means are calculated, they are used to calculate the interclass variance for the current state (5).

$$\sigma_b^2(t) = q_1(t)q_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (5)$$

The interclass variance is used to find the largest value for the threshold, the threshold obtains the value of 'i' when a new variance succeeds the old one. 'varmax' represents the largest threshold value at the current time, if that value is succeeded, it is replaced by the current squared variance:



```
if varSquared > varmax:
    threshold = i
    varmax = varSquared
```

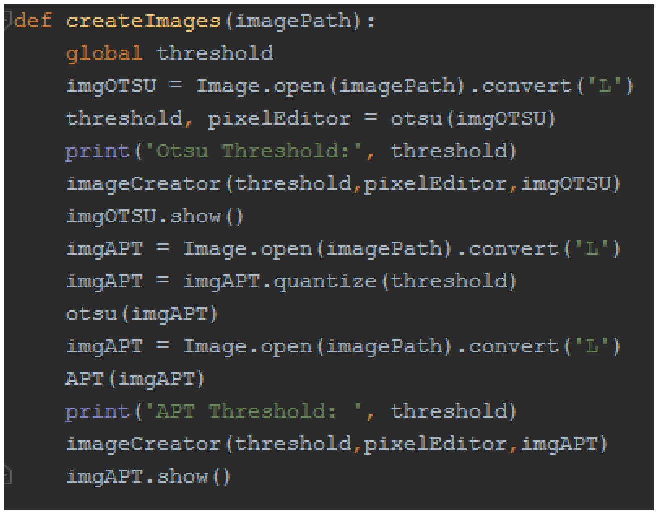
(6)

When this process is finished, we have obtained the threshold value for the current image. We then iterate through the original image, accessing it's pixels. If the value of the image's current pixels is larger than the threshhold value, it obtains the max value for white, 255, else, it is 0 for black.

$$g(x,y) = \begin{cases} 1, & f(x,y) \geq T \\ 0, & x < T \end{cases}$$

(7)

Now this is all the is required for the Otsu method. Now, to do the APT method, you must normalize the histogram around the threshold value, this is known as quantization. Then, for the quantized image, you must repeat the steps (1-7) and compute a new threshold value, it should be significantly lower than the threshold value calculated for the Otsu's method. This new threshold value will then go through the whole process, equations (1-8) to give you the APT image.



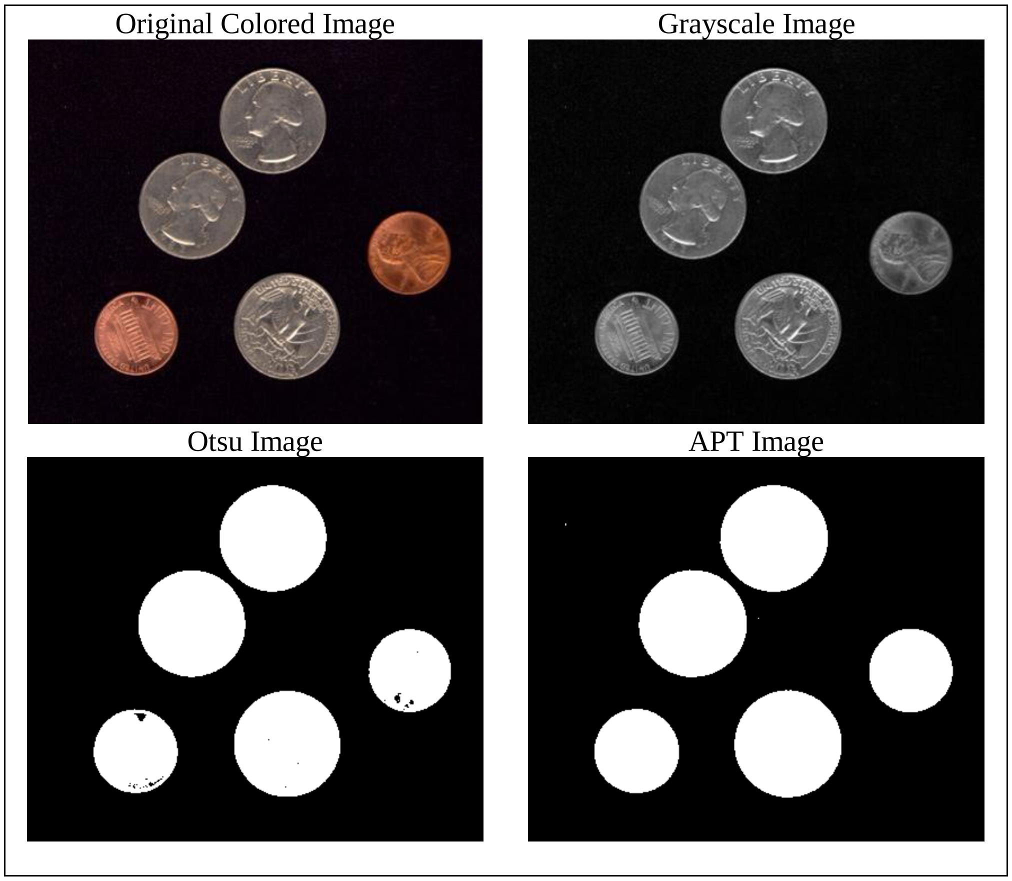
```
def createImages(imagePath):
    global threshold
    imgOTSU = Image.open(imagePath).convert('L')
    threshold, pixelEditor = otsu(imgOTSU)
    print('Otsu Threshold:', threshold)
    imageCreator(threshold,pixelEditor,imgOTSU)
    imgOTSU.show()
    imgAPT = Image.open(imagePath).convert('L')
    imgAPT = imgAPT.quantize(threshold)
    otsu(imgAPT)
    imgAPT = Image.open(imagePath).convert('L')
    APT(imgAPT)
    print('APT Threshold: ', threshold)
    imageCreator(threshold,pixelEditor,imgAPT)
    imgAPT.show()
```

The abstracted view of running the Otsu and APT process

III. IMPLEMENTATION RESULTS

For some input images where the background is clearly distinguished, the Otsu and APT method work great, the object(s) are clearly distinguishable, while some images where the range of colors are greatly distributed do not look so great. When comparing the Otsu and APT images side by side, the APT images will always be a bit lighter, and have a lower threshold, due to the nature of the APT equation, where you limit the amount of possible colors in a certain range from the threshold obtained during the Otsu method. Below we examine the images in depth.

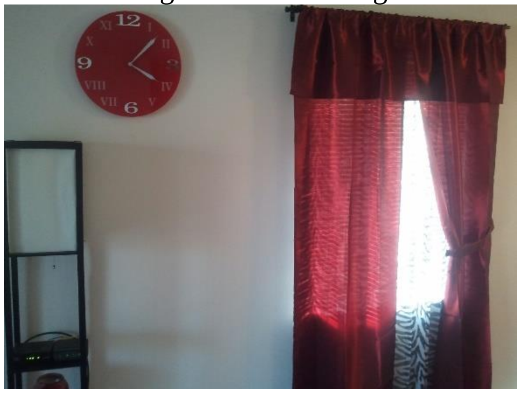
Case 1: Coins
Otsu Threshold: 58
APT Threshold: 34



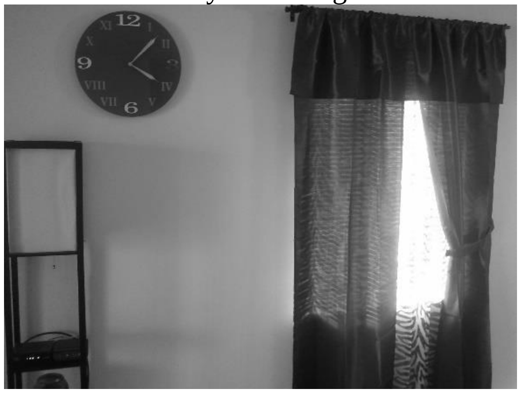
For the case of the coins, since the objects are clearly distinguishable from the background, the adaptive imaging algorithms run with great success in modifying the images. Each coin is made white, due to their values when compared to the dark background. Since the threshold for the APT image is significantly smaller than the Otsu image, it is more capable to remove some the ‘darker’ bits, which is why the circles are completely white.

Case 2: Room with clock
Otsu Threshold: 96
APT Threshold: 48

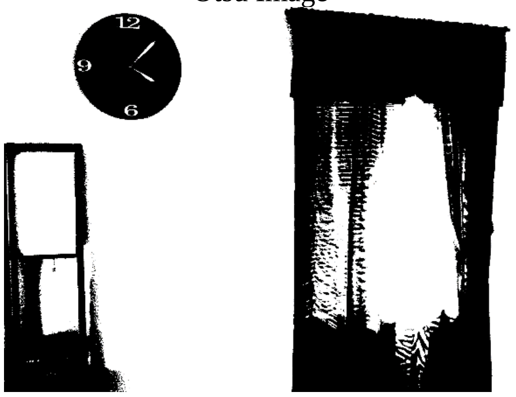
Original Colored Image



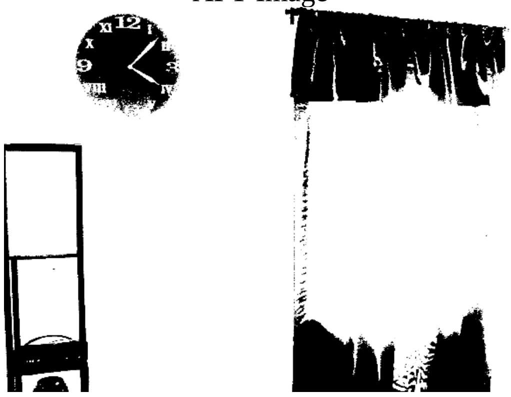
Grayscale Image



Otsu Image

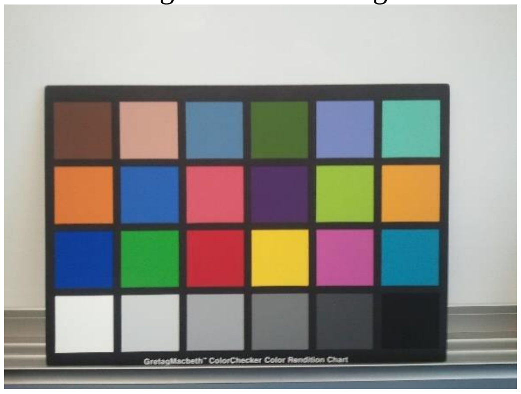
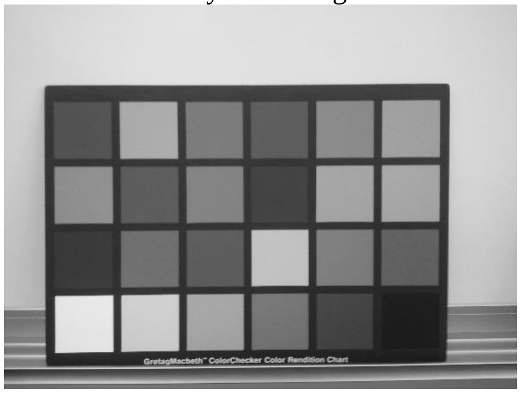
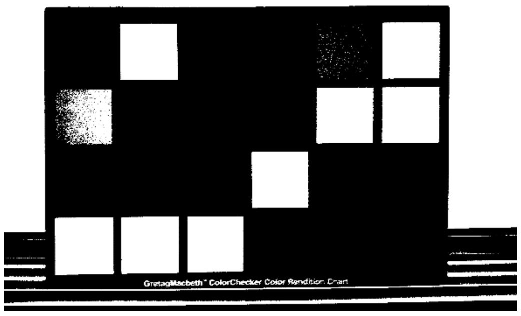
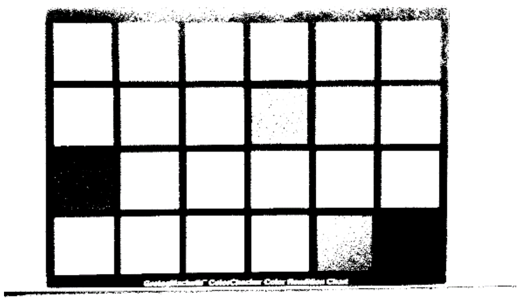


APT Image



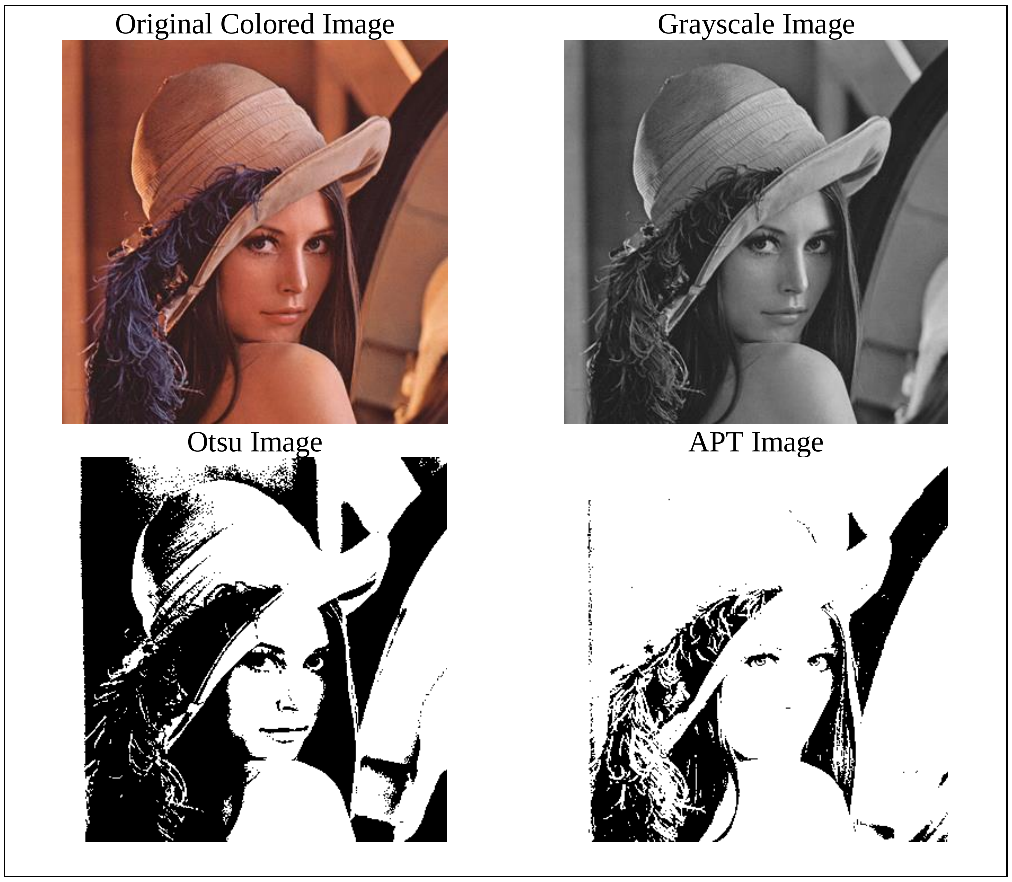
In this image, there are many distinguishable objects, the clock, curtains, and stand. For the Otsu image, it is noticeable that the threshold is very large, this is likely due to the light colors created by the light coming through the window, and the light-colored walls. It is noticeable that specific numbers are visible on the clock, likely due to that they are bolded, and represents a clear shade of white. Also, it is noticeable that certain parts of the sheets over the window are displayed, this is likely due to the values being very close to the threshold values. When looking at the APT image, the threshold becomes lowered, this makes lighter objects white, and which is why it is possible to see the numbers on the clock. It is also noticeable that the sheets hanging over the window have been written as white, because they are close to the threshold value.

Case 3: Color Palette
Otsu Threshold: 142
APT Threshold: 64

Original Colored Image	Grayscale Image
	
Otsu Image	**APT Image**
	

In this image, we have a variety of different colors. It is a bit difficult to analyze this image, because the main factor that comes into this image is the method used to create the grayscale image. However, one thing that is easy to point out in the colored image is that the orange square on the left side, is sort of half white, and half black for Otsu method. This is likely due to the lighting conditions making an impact on the pixel values; this means the grayscale values for the pixels in the orange sector come close to the threshold of 142. For the APT image, many of the colors have been converted to white, it is noticeable that the 2 squares that have remained black are the dark blue, from the left side, and the black, in the left corner. It is likely these are the only two that stay dark because of their dark shades.

Case 4: Lena
Otsu Threshold: 93
APT Threshold: 47



For this specific image these adaptive thresholding algorithms are not particularly useful, specifically APT. There are a variety of different colors throughout this image, leaving a great mix of white and black pixels; these algorithms are great when the difference between the colors are noticeable to the human eye. The pixels that were under the threshold were the predominately light parts of the image, the hat, skin, and reflection in the mirror. APT makes the already ‘light’ parts of the image lighter, resulting in a very white, very abstracted image. The reason I believe this APT method is not useful here is because it removes so many details, on a very complex image; you can no longer tell the facial features of Lena from the source image.

IV. Conclusion

Adaptive thresholding algorithms prove to be most useful in cases where there is a distinct background of an image, where there is a clear, distinguishable difference from the main focus of the image. For the Otsu, and APT method, there is an involved computational process to obtain a threshold value, where images are then segmented into dark and light pixels. When compared, the Otsu method performs well in situations where there are clear distinctions between light, and dark pixels but performs extremely well when pixels are scattered (Lena). However, Otsu method is succeeded by APT where there is a clear distinction between light and dark pixels (Clock, Coins), but looks very poor in images where the pixels are not distinguished between light and dark (Lena). Both algorithms have their advantages and disadvantages when it comes to a specific type of image.

References

[1] Asari, Vijayan. A fast and accurate segmentation technique for the extraction of gastrointestinal lumen from endoscopic images. Medical engineering & physics. (2000) 22. 89-96.

[2] Juan Pablo Balarini, and Sergio Nesmachnow, A C++ Implementation of Otsu’s Image Segmentation Method, Image Processing On Line, 6 (2016), pp. 155–164.