UNDERWATER COLOR CORRECTION

SITI NOR IFFAH BINTI NORZAMRI

BACHELOR OF ELECTRICAL ENGINEERING WITH HONORS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

UNDERWATER COLOR CORRECTION

SITI NOR IFFAH BINTI NORZAMRI

A report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electrical Engineering with Honours

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled "UNDERWATER COLOR CORRECTION is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name :

Date :

APPROVAL

I hereby declare that I have checked this report entitled "UNDERWATER COLOR CORRECTION" and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Electrical Engineering with Honours

Signature :

....

Supervisor Name :

....

Date :

....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

All praises to almighty Allah for His kind gesture, love, protection, guidance, endurance, wisdom and privilege given to me to accomplish this research successfully. Hereby, I want to express my gratitude and appreciation to my parents who have supported me morally and financially thru my degree life. I would like to record my deepest thanks to my supervisor, Mr. Lim Wee Teck who has contributed valuable suggestions and great help to coordinate my research work. Without his guidance, precious advices and encouragement, this research might not be completed as presented here. My further gratitude goes to Universiti Teknikal Malaysia Melaka (UTeM) for providing the necessary infrastructure throughout my degree studies.

Last but certainly not least, my sincere appreciation to my parents, relatives and friends for their continuous support, patience, unfailing love, prayers and faith in my perseverance over the entire period of my studies.

ABSTRACT

This report presents the color correction for underwater images. Recovering correct or at least realistic colors of underwater scenes is a very challenging issue for imaging techniques. The crucial reason for this problem is due to the density of the seawater and the penetrating of light. Since the light will be absorbed and scattered by small particles when travels in water, underwater imaging exists in difficulties of color cast. It gives impact to the limitation of visibility in the sea water. The need to correct colors of underwater images is an important task required in all image-based applications like underwater research such as coral bleaching and navigation as it is vital for the researchers to monitor and tracking the corals's health and pathway for the underwater transportation. To restore the color of the images, the method that is used in this project is Unsupervised Color Correction method and it was technically simulated in MATLAB by using 36 numbers of input images that categorised to; bluish, greenish and normal images respectively. In UCM method, it mainly has two stages to solve the problems mentioned above. First, equalization of RGB colors to achieve equal color values of the RGB component. Second, contrast to RGB color models to improve the contrast of the images. The result is evaluated visually and quantitatively by using Sobel edge detector to detect the edge and proved that whether the color of the images is improved by using Unsupervised Color Correction method. From the results obtained, it shown that the input images that is bluish and greenish improved significantly rather than the normal images that having a deterioration in terms of visual and the number of edges.

ABSTRAK

Laporan ini membentangkan pembetulan warna untuk imej bawah air. Memulihkan warna yang betul atau sekurang-kurangnya warna realistik di bawah air adalah isu yang sangat mencabar untuk teknik pencitraan. Alasan penting untuk masalah ini adalah kerana ketumpatan air laut dan penembusan cahaya. Oleh kerana cahaya akan diserap dan tersebar oleh zarah-zarah kecil apabila bergerak dalam air, pengimejan bawah air wujud dalam kesulitan warna cast. Ia memberi impak kepada penglihatan di air laut. Keperluan untuk membetulkan warna gambar bawah air adalah satu perkara penting yang diperlukan dalam semua aplikasi berasaskan imej seperti penyelidikan bawah air seperti pemutihan dan pelayaran karang kerana penting bagi penyelidik untuk memantau dan menjejaki kesihatan karang dan laluan untuk pengangkutan bawah laut. Untuk memulihkan warna imej, kaedah yang digunakan dalam projek ini adalah kaedah Pembetulan Warna Tidak Teratur dan secara teknikalnya disimulasikan dalam MATLAB dengan menggunakan 36 bilangan imej input yang dikategorikan; imej biru, kehijauan dan normal masing-masing. Dalam kaedah UCM, ia mempunyai dua peringkat untuk menyelesaikan masalah yang disebutkan di atas. Pertama, penyamaan warna RGB untuk mencapai nilai warna yang sama komponen RGB. Kedua, kontras dengan model warna RGB untuk meningkatkan kontras imej. Hasilnya dinilai secara visual dan kuantitatif dengan menggunakan detektor kelebihan Sobel untuk mengesan tepi dan membuktikan sama ada warna gambar itu diperbaiki dengan menggunakan kaedah Pembetulan Warna Tak Bertanda. Dari hasil yang diperoleh, ia menunjukkan bahawa imej input yang kebiruan dan kehijauan bertambah baik dengan ketara dan bukannya imej biasa yang mengalami kemerosotan dari segi visual dan bilangan tepi.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

RGB	Red, Green, Blue
HSI	Hue, Saturation, Intensity
HE	Histogram Equalization
UCM	Unsupervised Color Correction Method

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter introduces underwater color correction and the cause of color distortion of underwater images. It consists of research background, problem statement, motivation, objectives and project scope.

1.2 Research Background

Images have become a significant way to acquire and record information replacing traditional sensors and the application of vision system are becoming more and more vast. Due to speedy development and advancement of technology, people have started to explore the ocean which consists of abundant energy, luxuriant mineral resources and biological supply. As ocean exploration increased, the area of underwater image processing has drawn many attentions over the last years. However, color distortion will affect the images in underwater environment and this caused the difference in the perspective of the objects at the underwater environment compared to their appearance in reality perspective. Image distortion occur due to seawater density which is 800 times denser than air [1]. Water surface divides the moving light from air to water into reflected light and penetrating light. When light rays move from the air to the water, it is partly reflected back into the air and partly enters the water.

Additionally, the penetrating light that enters the water reduced as going deeper and deeper towards the bottom of the sea. This is because the water molecules absorb a certain amount of light [2] which resulted in darker images as the depth increases. As shown in the Figure 1-1, not only the amount of light rays is reduced when it deepens but the colors also will disappear gradually depending on the wavelength of the light. At the depth of 3m red color disappears, then orange color begins to disappear. Yellow color will fades away at the depth of 5m then followed by the green and purple color will goes off at further depth. Figure 1-2 shows that blue color is able to travel to the furthest depth due to its short visible wavelength. This makes the underwater images having been dominated mainly by blue color. In addition to excessive amount of blue color, the images also lost its brightness and contrast. The need to correct colors of underwater images is significant task required in all image-based applications like underwater research such as coral bleaching and the navigation. Therefore, underwater color correction is needed to improve the distorted images.

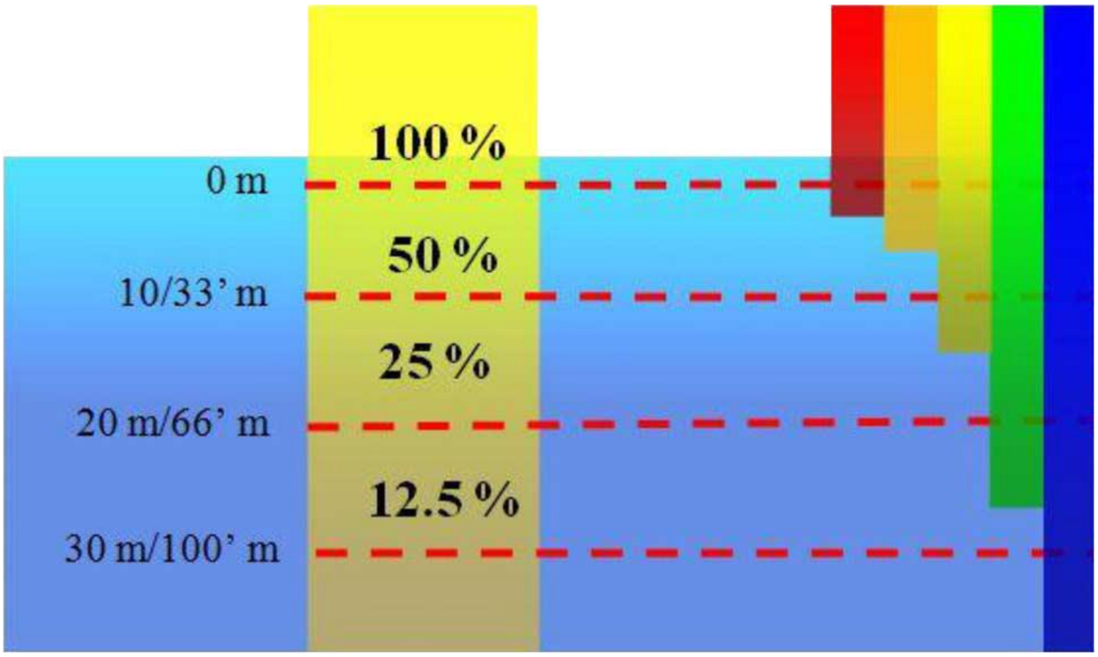


Figure 1-1: Different wavelengths of light are absorbed at different depths [1]

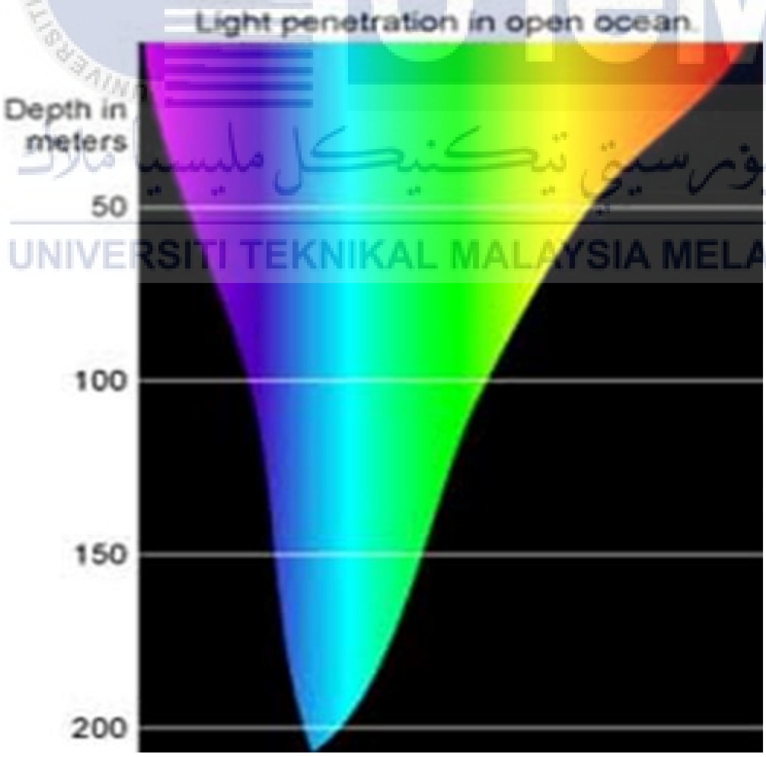


Figure 1-2: Color Penetration Pattern [2]

1.3 Problem Statement and Motivation

Coral reefs are important to protect the coastline from the damaging effects of the waves and tropical storm. Other than that, it provides shelter for many marine organisms and also it is the source of nitrogen and other essential nutrients for marine food chains. Due to developing activities and the increasing number of the coastal populations, the habitat of coral reefs are threatened. 60% of the world's coral reefs are at risk [3] due to human action such as coastal advancement, devastating fishing and aquatic pollution. Corals are bleached when there is pollution – This causes the color of the corals to fade away and change to white color because the algae that live in the corals will leave them when the surrounding environment was polluted.

In order to get data of surveys on coral reefs and fish population, one method used by scientist are the satellite. From the satellite, large-scale images can be obtained but because of the large size, the details of the data may not be accurate. So, another method is to dive into the sea and to take pictures of the corals. It is not practical for scientist to dive and gather the data alone, so they hire or request volunteers to help them. But the volunteers may use different camera and method, thus the quality of the image will become varies too. It is also known that the underwater view will looks in bluish or greenish tone. To regain and improve their colors to have a good visibility for research purpose, this project will use Unsupervised Color Correction method.

The motivation of this project is to spread the awareness about the coral bleaching problems and to obtain the natural colors of any objects in underwater as it is vital to underwater research such as survey on corals and fish. By using Unsupervised Color Correction method, the quality of the underwater images will be improved.

1.4 Objectives

The main aim of this project is to improve the color distorted of the underwater images by using Unsupervised Color Correction method. The objectives for this project are as follows:

i) To improve the color of images in the underwater environment.

ii) To obtain an equally distributed histogram output to show that the images have undergo the color improvement.

1.5 Project Scope

The project scopes included:

i) The technique used is simulated in MATLAB.

ii) The 12 of image dataset was referred to [4], [5], [6], [7], [8] and [9] be tested on the underwater images corresponding to image that have bluish or greenish tone and poor contrast.

1.6 Conclusion

The color of the images needs to be improved so that it can be used to provide accurate information to marine scientists. Thus, the conditions of coral reefs can be more closely monitored and suitable actions can be performed quickly to ensure the sustainable environment for future generations.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter presents related literature concerning basic theory of vision system and underwater image color correction techniques and why the UCM is the most suitable method for this project compared to the others method.

2.2 Basic Concept Of Digital Images

Digital image consists of picture elements that is called as pixel [10]. Commonly, pixels are sort in an ordered rectangular array. Dimension of the arrays represents the size of an image. The image height is determined by the number of rows in the array and the image width is the number of columns (M×N) matrix [11]. The coordinate system of image matrices described x as increasing from left to right and y as increasing from top to bottom. For each of the pixels, it has their own intensity value or brightness [12].

For grey images, it only has one color frame and its intensity is from darkest grey (black) to lightest grey (white) and it’s ranged from 0-255 as shown in Figure 2-1. For color images, it has three color frame, Red, Green and Blue. Images represented in the RGB color model include three component images, one for each primary color respectively. The three primary colors can produce a broad array of colors if it combines in various ways such as in Figure 2-2.

Figure 2-1: The Intensity Range [13]

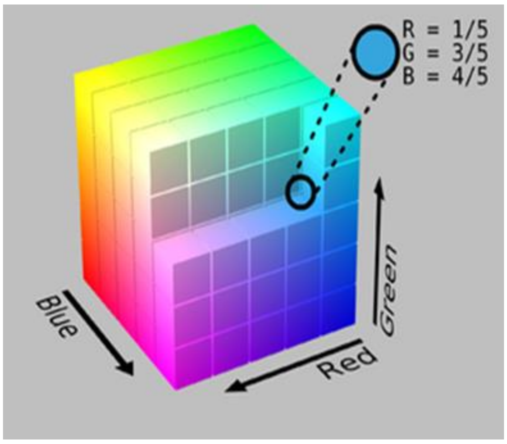


Figure 2-2: RGB Color Model [14]

Hue is a color attribute that describes a pure color, saturation gives a measure of the degree to which pure color is diluted by white light and intensity is brightness. Other than the RGB color model, HSI model is also an important basic theory in the vision system. Based on Munsell color model in Figure 2-3, it represented as a three-dimensional cylindrical shape that equals to hue (H), saturation (S), and intensity (I) [15], and it was the first model that isolates the three color components into disciplinary independent, regular, and three dimensional space.

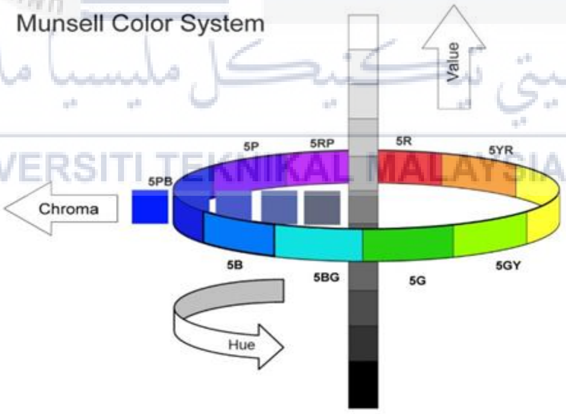


Figure 2-3: Munsell Color Model [15]

In the vision system topic, the histogram of an image usually refers to a graph that displaying the number of pixels in an image (on the y-axis) at each contrary intensity value (on x-axis) found in that image as shown in Figure 2-4 [16]. There are 255 different feasible intensities for an 8-bit grayscale image, and so the histogram will graphically show 255 numbers displaying the dispersal of pixels amongst those grayscale values. Histograms can also be taken of color images which is individual histograms of red, green and blue channels or a 3-D histogram can be produced, with the three axes representing the red, blue and green channels, and intensity at each point representing the pixel count. The left side of the graph represents the blacks or shadows, the right side represents the highlights or bright areas and the middle section is mid-tones.

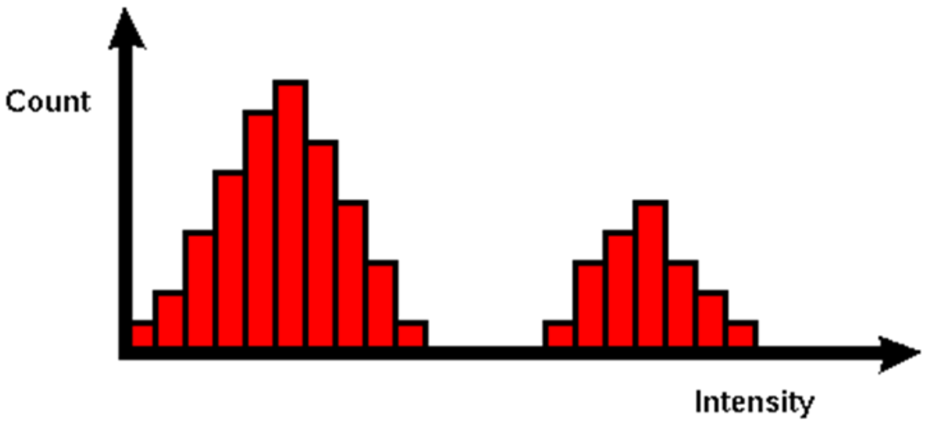


Figure 2-4: Image Histogram [16]

When the input image has a good contrast and no disturbance occur, the histogram of the image will be equally distributed. If it is in high intensity, the graph will more on the right side and if it is in low intensity the graph will more at the left side as shown in Figure 2-5. The histograms has wide application in image brightness. Not only in brightness, but histograms are also used in adjusting contrast of an image. Another important use of histogram is to equalize an image so that the image will have a better output in term of color.

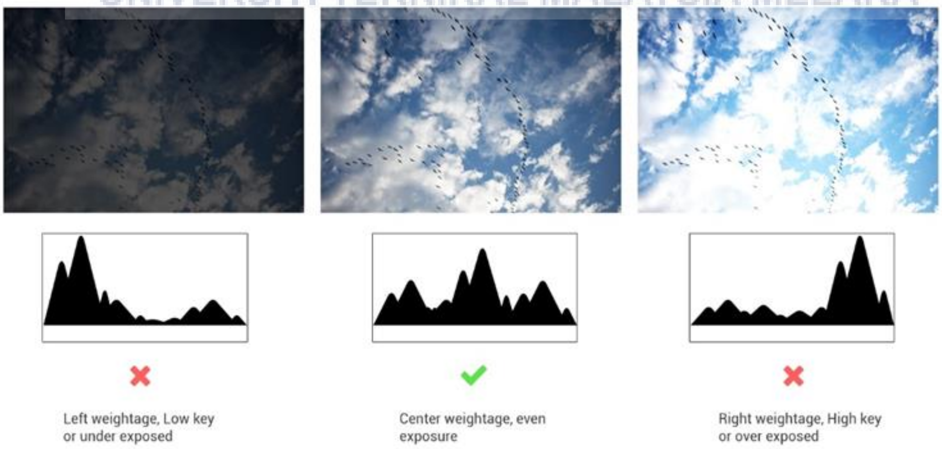


Figure 2-5: Comparison of the Histogram Image [17]

2.3 Previous Research

The underwater color correction technique proposed by various researcher will be explained in this part.

2.3.1 Dark Channel Prior method

Capturing images underwater is very difficult and it is due to haze caused by light that is reflected from a surface and is deflected and dispersed by water particles, and color change due to altering degrees of light attenuation for dissimilar wavelengths. K.He stated that in his article [18], haze is caused by suspended particles such as sand, minerals, and plankton that live in lakes, oceans, and rivers. From article that was written by J. Y. Chiang and Y. C. Chen [19], a portion of the light meets these suspended particles when the light reflected from objects propagates toward the camera. In Figure 2-6 it shows that the effects caused by the light scattering and color change in the underwater images and the result by using Dark Channel Prior method is shown in Figure 2-7.

Figure 2-6: The original image [19]

Figure 2-7: The enhanced image after using Dark Channel Prior method [19]

By using this removing haze method, it can remarkably increase the visibility of the scene and the color shift is corrected. The haze removal can create depth information and give advantages to many vision algorithms and advanced image editing. Haze or fog can be a useful depth clue for view understanding. The bad haze image can be improved. However, haze removal is a challenging issue because the haze is dependent on the unknown depth information [18]. Typically, in the underwater environment there are some pixels (called “dark pixels”) that have very low intensity in at least one color (RGB) channel. In the haze image, the intensity of these dark pixels in that channel is mostly caused by the particles, plankton and sand. Thus, these dark pixels can directly provide accurate estimation of the haze’s transmission.

R. T. Tan [20] observes that the haze-free image must have higher contrast compared with the input haze image and he removes the haze by maximizing the local contrast of the restored image. The results are visually captivating, but it is not physically valid. From [18], the researcher stated that his approach also has its own restriction. The dark channel prior may be invalid when the scene object is intuitively like the underwater environment over a large local area and no shadow is cast on the object. Although the approach works well for most of haze images, it may fail on some extreme cases.

2.3.2 The Gray World Assumption

The Gray World Assumption method that was proposed by Buchsbaum [21], it used the assumption where the illuminant is uniform over the entire image. It is a well-known method to improve color of images at the underwater environment. In its simplest version where it assumes that the information in each channel of the image (RGB) is the representative gray level. It estimates the illuminant by calculating the average global space color. The fact that the illuminant can be estimated by calculating the average light reaching the observer was previously known. Buchsbaum was the first person to formalized the method.

M. Vision and M. Ebner [22] assumed that on average the world is gray. He estimates the illuminant by assuming that a certain standard spatial spectral average exists for the entire visual field. This average can be used to estimate the illuminant. Once the illuminant is estimated, the reflectance can be computed. Quite many algorithms have been proposed that use the gray world assumption in one way or another. The main disadvantage of gray-world method is that they will not valid and fail to produce correct colors if there are no sufficiently large numbers of hues present in the scene. The result by using this method is shown in Figure 2-9.

Figure 2-8: The original image [22]

Figure 2-9: The enhanced image after using The Gray World Assumption method [22]

2.3.3 Red Channel Method

The Red Channel method can be interpreted as a variant of the Dark Channel method used for atmospheric images surrounded by haze. The first step in this method to estimate is the color of the water. A pixel with the maximum depth was selected in relation to the camera. The degradation of image is supposed to depends on the location of pixel. The result by using this method is shown in Figure 2-10.

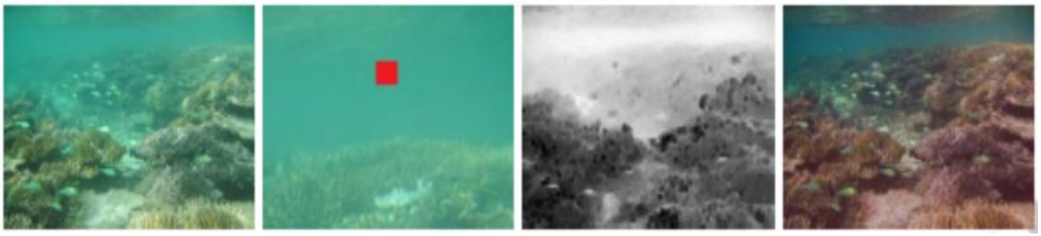


Figure 2-10: The enhanced image after using Red Channel method [23]

The analogous method suggested by K.He [18] would be take the top 10% brightest pixels in the input image and among this set of pixels, the one that is brightest in the degraded image. This pixel that was chosen does not need to be the brightest in the Red Channel. In the underwater case, the researcher observed that picking among that 10% of pixels the one that has lower red component in it obtained the good result.

2.3.4 Color Image Histogram Equalization

Histogram equalization (HE) that proposed by R. Jaiswal, A. G. Rao, and H. P. Shukla [24], it is an image intensity and contrast adjustment technique with the image's histogram. Every pixel in color image has three color channels, that is RGB (red, green, blue). The equalization of the histogram is valid in pictures with backgrounds and frontal areas both bright or both dim. Histogram equalization can depict hidden details in an image by stretching out the contrast of local regions and thus creating the differences in the regions more observable. This intensify the dynamic range of gray pixel values, so it can improve the image contrast. The basic idea of histogram equalization is shifting the histogram of input image to equally distributed form. Low contrast images have small histogram while for the wide histogram distributions represent high contrast images. This technique is distinctly helpful in images with large related areas of tone such as an image with a very light background and dark foreground. It enhances the contrast of image by modifying the values in an intensity image so that the resulted image almost matches a specified histogram (uniform distribution by default). If histogram equalization used to process the original images directly, it will lose a lot of color information and the resulted image become gray. J. H. Lee [11] in his article stated that the image processed in the experiment is colorful, so color image histogram equalization is used to realize color correct.

Each channel is histogram equalized by a uniformly increasing gray level transform later then the three images is combined after histogram equalization respectively. The output image will have high contrast and many color information. While, this method is based on independent equalization without consider the relationship among RGB three channels, which will result little color distortion. J. Y. Chiang and Y. C. Chen [19] said that the vital achievements are the short time of processing it and the simple algorithm.

Histogram Equalization works by stretching and equalizing the histogram through the range of intensities and is a graph which shows the frequency of data occurrence in the whole data set. This is an easy and straightforward technique and it assist to equalize the color contrast in the images and addresses the problem of lighting. From article proposed by B. Hu and B. Zheng [25], it has a drawback that is it amplifies the background noise that present in the image and lead to decrease in the useful signal. So, it will have unrealistic effects. In Figure 2-11, it shows the original image and Figure 2-12 shows the enhanced image by using Histogram Equalization method.

Figure 2-11: The original image [24]

Figure 2-12: The enhanced image after using Histogram Equalization method [24]

2.3.5 Unsupervised Color Correction (UCM)

In this method, it based on the contrast of RGB and HSI color model. This technique eliminates the bluish color cast, increase the low red and low illumination problem in order to achieve high quality images for the output result. It has 3 stages that are; balancing an image, removal of the color cast and improvement of the intensity and increase of real color of the input image [26]. In Figure 13 (a), it shows the original image and (b) shows the enhanced image.

Figure 2-13: The original image [26]

Figure 2-14: The enhanced image after using UCM method [26]

After doing several readings to choose which method that is proper to use for this project, Unsupervised Color Correction method will be used to obtain a good result of the image at the underwater environment because it is an effective method to balancing the color of the image, removal of the color cast, it can improve the illumination and improve the true color. Thus, it can produce a better result. So, this method has 3 stages; equalization of RGB colors, contrast correction of RGB color model and contrast correction of HSI color model.

Stage 1:

A. Equalization of RGB colors

In the underwater situation, images are not in correctly color balanced. In order to equalise the RGB value, the first step of the proposed approach is to calculate the maximum values. Let I_R(i, j), I_G(i, j) and I_B(i, j) be respectively the red, green and blue components of an RGB image of size M x N pixels, Where i = 1, ..., M; j = 1, ...N. The maximum pixel values of each color component R_{max}, G_{max} and B_{max} are calculated:

$$R_{max} = \max_{i,j} I_R(i, j) \quad (2\text{-}2)$$

$$B_{max} = \max_{i,j} I_B(i, j) \quad (2\text{-}1)$$

$$G_{max} = \max_{i,j} I_G(i, j) \quad (2\text{-}3)$$

Then the average values of each color component R_{avg},G_{avg} and B_{avg} are calculated:

$$R_{avg} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} I_R(i, j) \quad (2\text{-}4)$$

$$B_{avg} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} I_B(i, j) \quad (2\text{-}5)$$

$$G_{avg} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} I_G(i, j) \quad (2\text{-}6)$$

$$A = \frac{B_{avg}}{R_{avg}} \quad (2\text{-}7)$$

$$B = \frac{B_{avg}}{G_{avg}} \quad (2\text{-}8)$$

$$R' = A \times R \quad (2\text{-}9)$$

$$G' = B \times G \quad (2\text{-}10)$$

Where R and G are original pixels values in the image and R' and G' are the adjusted pixel values.

Stage 2:

B. Contrast Correction of RGB Color Model

Stretching the range of intensity values in order to extent the desired range of values. Before performing contrast correction, a determination is made for the upper and lower limits of the image for contrast correction of each band. Normally, the range of values in 8-bit color channel is 0-255.

$$P_o = (P_i - c)\frac{(b-a)}{(d-c)} + a \tag{2-11}$$

Where,

- P_o is the contrast corrected pixel value;
- P_i is the considered pixel value;
- a is the lower limit value which is 0;
- b is the upper limit value which is 255;
- c is the minimum pixel value currently present in the image;
- d is the maximum pixel value currently present in the image;

In order to obtain better results, it is proposed that the contrast correction method must be applied to an image to the upper side, lower side and both sides as explained below.

i) Contrast Correction to Upper Side: The lowest color value component is selected; that is normally the red color component which is the first color that disappears underwater within 3 meters.

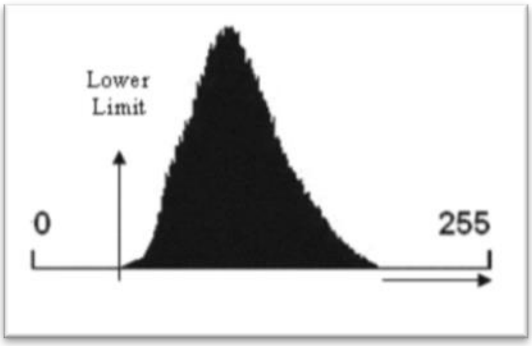


Figure 2-15: Upper side component

(Upper Limit – Minimum of Red) / (Maximum – Minimum)

Lower Limit = Minimum of Red, Upper Limit = 255

Lower limit is set to minimum of Red instead of zero

ii) Contrast Correction to Lower Side: The prominent color cast component is selected which has heavy color cast and in underwater images that is the blue color.

Figure 2-16: Lower side component

(Maximum of Blue - Lower Limit) / (Maximum - Minimum)

Lower Limit = 0, Upper Limit = Maximum of Blue

Upper limit is set to Maximum of Blue instead of 255.

iii) Contrast Correction to Both Sides: To find the color component which has the value between lower and higher color components. The adjustments are made toward both directions, minimum and maximum side to help the histogram equally distributed well to both directions as shown.

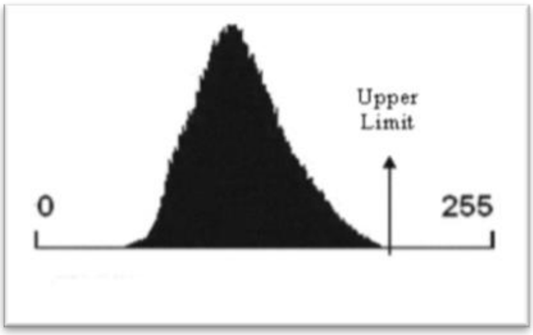


Figure 2-17: Both sides component

(Upper Limit - Lower Limit) / (Maximum - Minimum)

Lower Limit = 0, Upper Limit = 255

2.4 Conclusion

It is important to do the literature review as it is the way to understand more about Unsupervised Color Correction method that used for this project to improve the color of the images. From [26], UCM is chosen because it can balancing the RGB values of an image. Underwater images have higher blue and green color compared to other colors and this high value are used to increase the low value color in order to make the image balanced. UCM also can remove the color cast by stretching the blue histogram towards the minimum side. The red and green color similarly has been increased by stretching the red towards the maximum side and the green color to the both sides of histogram. The use of these three different types of contrast correction really helps the image to retain the true colors of the images. The core reason why UCM method is better because it considered the image's properties and enhanced the image based on its characteristics.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This section explains about the implementation of this project. The flow of this project will be briefly explained and illustrate using flowchart. Also, the dataset that will be use and the validation method to quantify the results. The Gantt Chart in the Appendix 1 section is the timeline for this project.

3.2 Overall Project Flow

This project follows the flowchart in Figure 3-1 to make sure that a systematic task can be done. It starts with the literature review section. This part is very important in this project because it needs to be done to gain knowledge in order to improve the understanding about the basic vision systems, to study about the current research for underwater environment and how the Unsupervised Color Correction method works. For the next step is to develop the algorithms for the UCM in the MATLAB. As explained in the literature review section, UCM has 2 stages which is equalization and contrast correction of RGB color model. Algorithms for these 2 stages is developed based on the formula that given in [26]. For UCM, it was tested using the dataset selected; bluish, greenish and normal image that will be further explained on 3.3. Then, the result will be analyzed and validate using the Sobel edge detection technique. After all the step is completed, a report will be submitted.

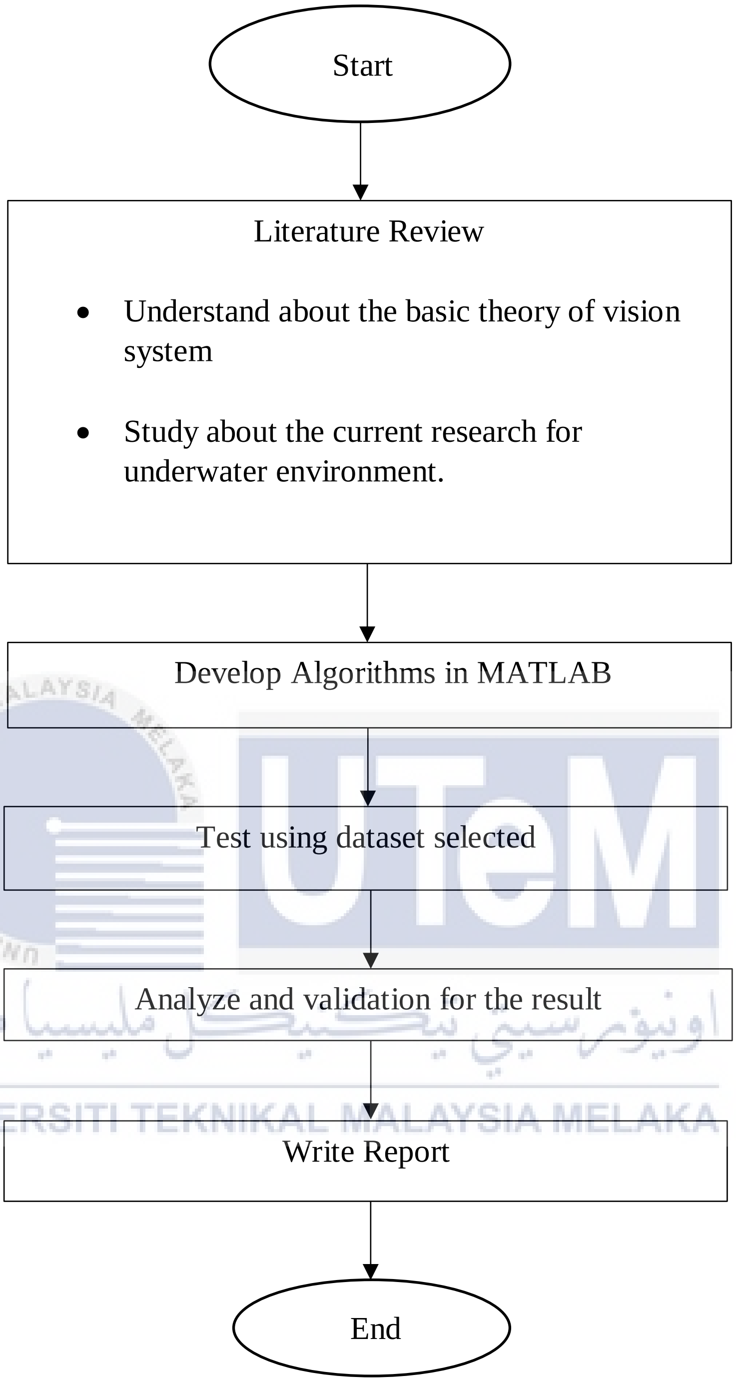


Figure 3-1: Flowchart of the project

3.3 Validation

To evaluate whether the dataset image really undergo color improvement or not, other than visually looking at the results, Sobel edge detection method will also be used to produce a measurable quantitative result. Sobel edge detection method is an image processing technique to find the boundaries of objects within images and it works by detecting discontinuities in the brightness of the image. This method will detect and returns the total number of edges found in that images as feedback. The algorithms for this validation method are developed by using MATLAB. In [26], it is determined that to use 85-175 as the threshold value to detect the edges. The images that have the maximum number of edges means that it has high feature content and clear image. Thus, it will be proved that the dataset images are successfully improved the color by using UCM.

3.4 Dataset of the project

This project will be evaluated using 12 images each of bluish, greenish and normal datasets that is taken from six websites. To choose the image, it must be in bluish, greenish color and also consist of corals and fish to be detected for the edge detection. To know which the input images belongs between those three, it will be explained at the section below. These images are taken from the Adventures in Loom-Science [4], Picture of Gebel El Rosas, Marsa Alam – TripAdvisor website [5], dpreview website [6], Wordatlas website [7], CoralTriangle website [8] and Github website [9]. Figure 3-2 is an example of bluish image. Figure 3-3 is an example of greenish image while Figure 3-4 is an example of normal image. The complete image dataset can be found in Appendix B.

Figure 3-2: Input bluish image

Figure 3-3: Input greenish image

Figure 3-4: Input normal image

3.4.1 The criteria for the input images

The result is specifically separated to three types of data; the input image that is in bluish, greenish and normal. A total of 12 images is choose for this project according to its category respectively. In this result, it is important to determine the criteria of input image that suitable to use Unsupervised Color Correction method. The median for Red, Green and Blue component for each of the bluish, greenish and normal images was found and tabulated in the table respectively. In Table 3-1, it shows the median for each of 12 bluish images. After the average and ratio is calculated, it shown that the suitable types of bluish images that its color can be improved by UCM method is the images that have median number of pixels approximately in Red = 21.67, Green = 125.92 and Blue = 152.25. The ratio of RGB component to choose the input image that categorized as bluish is based on median's average that is 7:42:51. The highest ratio among these three components is the blue component. So, if any image that has this type of ratio, it is considered as bluish image.

For the greenish images, the median is found out as shown in Table 3-2. After the suitable average median and ratio for the input images is calculated, the images that have median number of pixels approximately in Red = 57.92, Green = 135.50 and Blue = 89.17 and for the ratio, 20:48:32 is accepted. As can be seen from the ratio of the RGB components, the red and blue component have the lowest ratio than the green component. For the normal images, the average median in number of pixels is approximately to have Red = 83, Green = 108.33 and Blue = 100.08 and the median is show in Table 3-3. For the ratio, it is 28:37:35. It cannot be decided to choose the normal image based on the median ratio, so next is to find the ratio of maximum values in the images. From Table 3-4, Table 3-5 and Table 3-6, it shows the maximum values of RGB components for the bluish, greenish and normal images. The ratio for the bluish images is 26:36:38, greenish images is 27:40:33 and for the normal images is 33:33:33. From the ratio for bluish and greenish it means that if the input images have higher ratio of blue or green component, it is categorized as these two criteria. For the normal images, it has equal ratio between the RGB components so all the dataset images that had been tested to have these criteria, it is categorized as normal images. Ratio for median and maximum that was calculated are the variables to choose and categorized the suitable input image for the color improvement using UCM in this project.

Table 3-1: Median for RGB component in bluish images

Input image	Median for Red component	Median for Green component	Median for Blue component
1	23	120	135
2	8	166	186
3	25	132	157
4	130	130	128
5	3	119	162
6	14	121	149
7	1	131	143
8	5	106	131
9	26	109	143
10	13	149	178
11	4	113	160
12	8	115	155
TOTAL	260	1,511	1,827

Median average of pixels for:

Red component: 260 / 12 = 21.67

Green component: 1,511 / 12 = 125.92

Blue component: 1,827 / 12 = 152.25

Ratio: 7:42:51

Table 3-2: Median for RGB component in greenish images

Input image	**Median for Red component**	**Median for Green component**	**Median for Blue component**
1	2	160	111
2	93	138	85
3	50	147	95
4	22	111	69
5	33	72	43
6	73	158	67
7	101	158	128
8	96	158	116
9	25	124	91
10	94	171	132
11	1	80	37
12	105	149	96
TOTAL	695	1,626	1,070

Median average of pixels for:

Red component: 695 / 12 = 57.92

Green component: 1,626 / 12 = 135.50

Blue component: 1,070 / 12 = 89.17

Ratio: 20:48:32

Table 3-3: Median for RGB component in normal images

Input image	**Median for Red component**	**Median for Green component**	**Median for Blue component**
1	75	105	103
2	67	115	112
3	113	137	118

Input image	Median for Red component	Median for Green component	Median for Blue component
4	110	115	98
5	88	78	92
6	83	94	60
7	120	84	49
8	36	148	143
9	74	34	35
10	79	178	197
11	102	95	67
12	49	117	127
TOTAL	996	1,300	1,201

Median average of pixels for:

Red component: 996 / 12 = 83.00

Green component: 1,300 / 12 = 108.33

Blue component: 1,201 / 12 = 100.08

Ratio: 28:37:35

Table 3-4: Maximum for RGB components in bluish images

Input image	Maximum for Red component	Maximum for Green component	Maximum for Blue component
1	95	205	209
2	42	172	255
3	108	218	243
4	181	184	198
5	107	217	247
6	131	228	255
7	207	205	255
8	220	235	255
9	197	224	255
10	201	221	255
11	150	215	255
12	184	190	255
TOTAL	1,823	2,514	2,682

Maximum average:

For Red component: 1,823 / 12 = 151.92

For Green component: 2,514 / 12 = 209.50

For Blue component: 2,682 / 12 = 223.50

Ratio = 26:36:38

Table 3-5: Maximum for RGB components in greenish images

Input image	Maximum for Red component	Maximum for Green component	Maximum for Blue component
1	82	255	230
2	220	245	179
3	162	255	232
4	186	255	209
5	126	163	141
6	181	255	172
7	171	211	195
8	161	240	207
9	181	255	245
10	194	255	194
11	46	173	98
12	166	200	148
TOTAL	1,876	2,762	2,250

Maximum average:

For Red component: 1,876 / 12 = 156.33

For Green component: 2,762 / 12 = 230.17

For Blue component: 2,250 / 12 = 187.50

Ratio = 27:40:33

Table 3-6: Maximum for RGB components in normal images

Input image	Maximum for Red component	Maximum for Green component	Maximum for Blue component
1	255	255	255
2	255	255	255
3	255	255	255
4	255	255	255
5	255	252	255
6	255	255	255
7	255	255	255
8	253	255	255
9	255	255	255
10	255	255	255
11	255	255	255
12	255	255	255
TOTAL	3,058	3,057	3,060

Maximum average:

For Red component: 3,058 / 12 = 254.83

For Green component: 3,057 / 12 = 254.75

For Blue component: 3,060 / 12 = 255.00

Ratio = 33:33:33

3.4.2 Bluish image

From the dataset image Figure 3-2, the histogram for each of the RGB component is displayed in Figure 3-5. In the histogram, the x-axis shows the level intensities and the y-axis shows the frequency of these intensities. It can be seen that, this histogram is not equalized thus the images are not balanced in color. To equalize the RGB value by using UCM, the first step is to find the maximum Equation (2-1), (2-2) and (2-3) and average value Equation (2-4), (2-5) and (2-6) for RGB and from these maximum values, the prominent color cast is found. In UCM, it keeps the dominant color cast channel constant since in this case, it has high blue color than red and green. Therefore, the dominant color value is used to increase the other colors to make the image balanced in color. From the dominant color cast that is blue color, two gain factors can be found out Equation (2-7) and (2-8) . The blue color is set as the target mean and the red and green component acts as multiplier to the target mean in order to achieve the balanced color image. UCM used these two colors channel to reduce the blue cast on the affected image. Then the original blue value, new red and green values is recombined. The result from this action can be seen in Figure 3-6 that represent the histogram.

Next step is to do contrast correction to the images in order to improve the contrast of the images. Clipping for each of RGB component (cut off few pixels values from both sides of the histogram) is used to remove the single outlying pixel that having too high or too low value of pixels to obtain a clearer image. In the UCM, it selects 0.2% and 99.8% [26] as determinants to do the clipping and the contrast correction only be applied to the remaining pixel value. Figure 3-7 is the histogram after clipping process. In order to obtain better results, the contrast correction was applied to an image to the upper, lower and both sides.

For the contrast correction to upper side, the lowest color value component is selected; normally the red color because it is the first color that disappears underwater within 3 meters. For the contrast correction to lower side, the dominant color cast that is blue is selected and was applied to the minimum side to solve the problem of bluish color cast. The last step in the contrast correction is the contrast correction to the both sides to help the histogram spreads well.

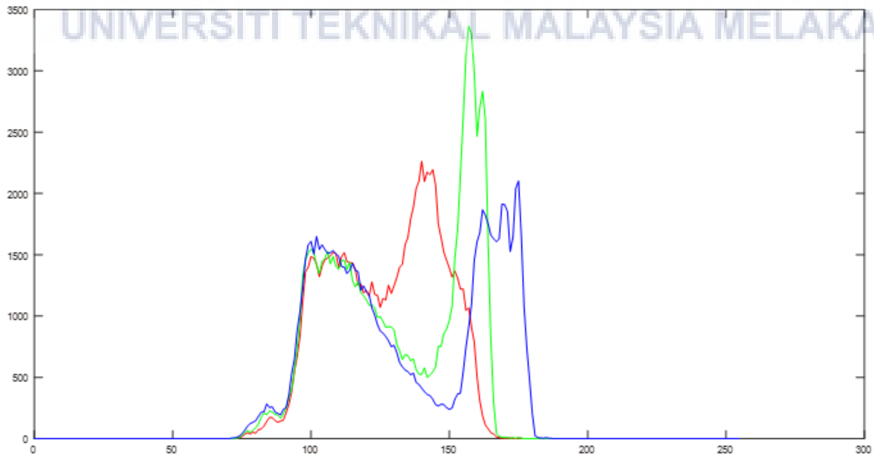


Figure 3-5: The separation of RGB histogram

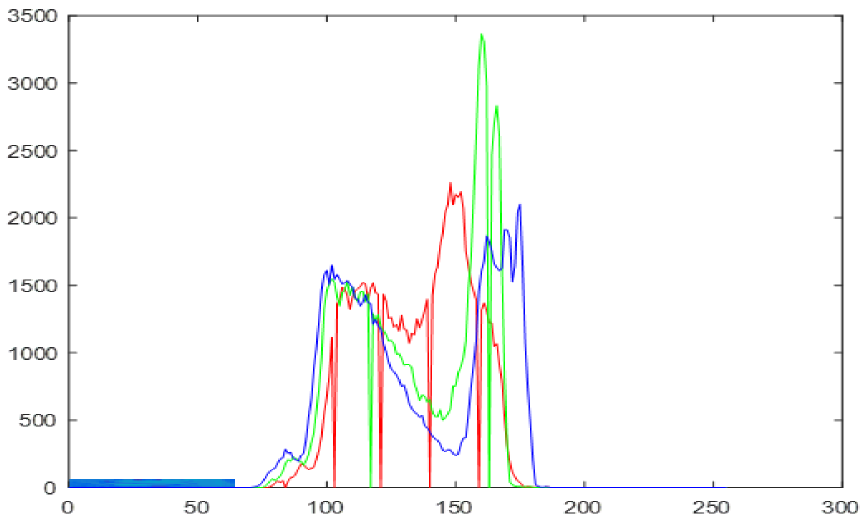


Figure 3-6: The histogram after recombined the new color channel

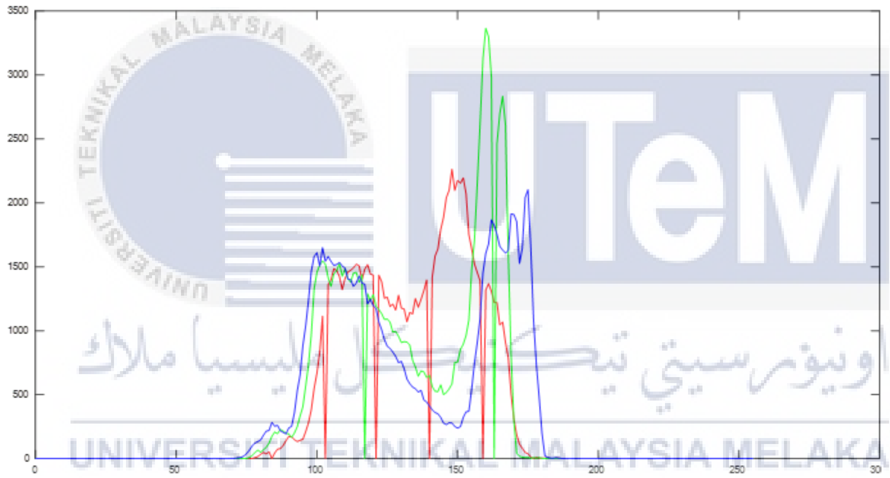


Figure 3-7: Histogram of each channel after Clipping

3.4.3 Greenish image

From histogram shown in Figure 3-8, it can be seen that the pixels for green color have higher intensity. For the greenish input image as in Figure 3-3, it has same but slightly different in the method as the bluish input image. It is different in terms of dominant color cast. After found out the values of maximum number of pixels in RGB component, it shows that green component has the higher number of pixels than red and blue.

For the equalization of RGB, the green color will replace blue color as the dominant cast in the images. From this green dominant color cast, two gain factors can be calculated. The green color is set as the target mean and the red and blue component acts as multiplier to the target mean in order to achieve the balanced color image. UCM used these two colors channel to reduce the green cast on the affected image. Then the original green value, new red and blue values is recombined resulting as shown in Figure 3-9. In Figure 3-10, it shows the histogram after clipping process. To increase the contrast of the images, it used the same ways as explained for the bluish input image.

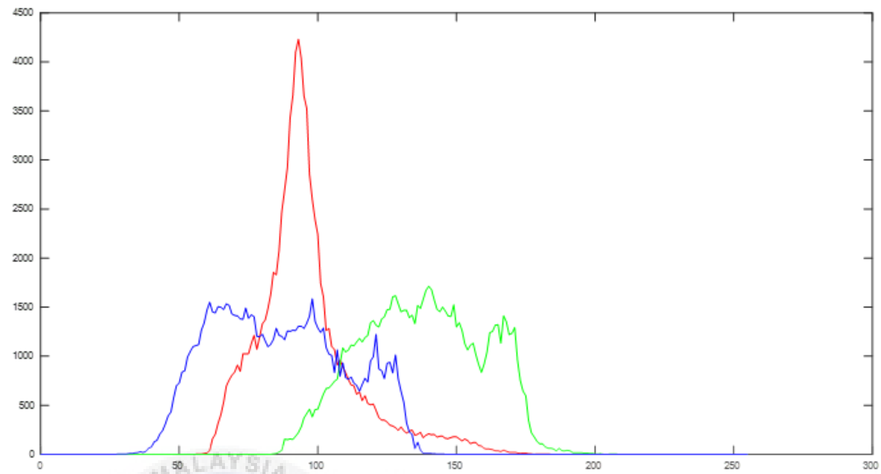


Figure 3-8: The separation of RGB histogram

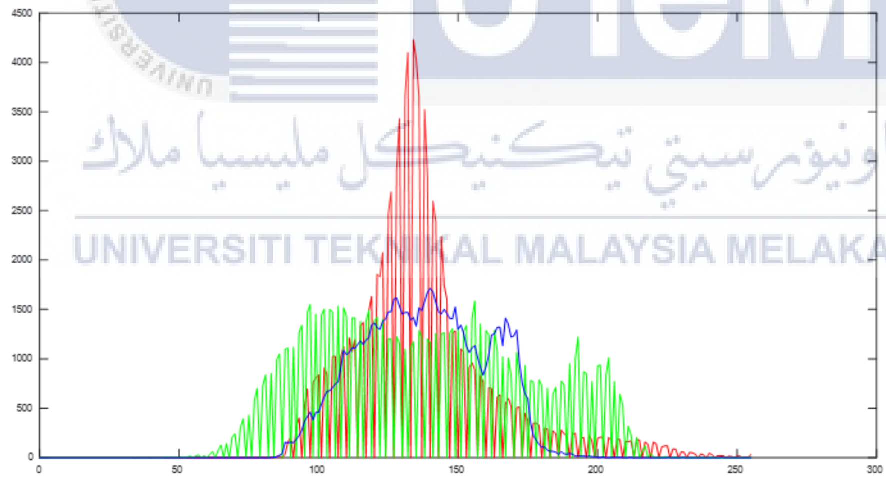


Figure 3-9: Recombine the separate new color channel

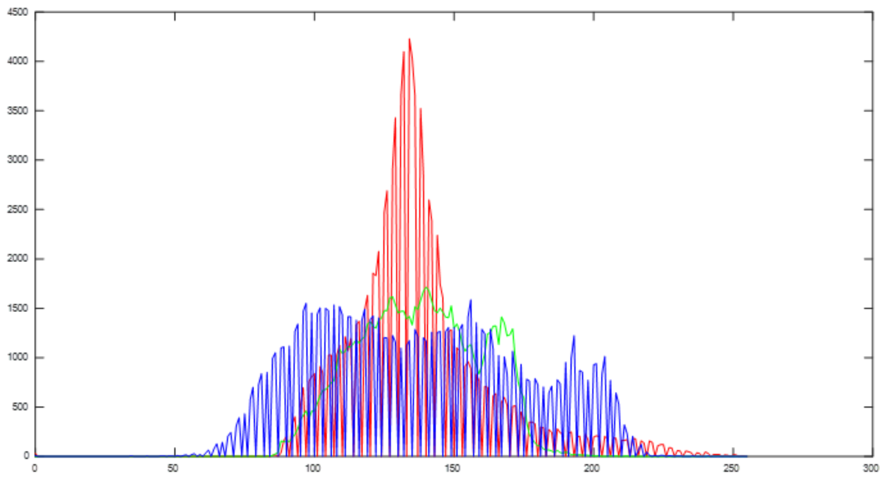


Figure 3-10: Histogram of each channel after Clipping

3.4.4 Normal images

From Figure 3-4, it is normal image because all the RGB component have equal maximum ratio that is 33:33:33. So, it has no dominant color cast because the RGB component is already equally balance as shown in Figure 3-11. After the equalization of RGB color stage, when the values from the target mean that is blue color, multiply with red and green component to obtain the new values for the lower color components that is assumed to be red and green and this step resulting a histogram as shown in Figure 3-12. The green component starts to spike up and it means that the number of pixels is increase. Figure 3-13 shows the histogram after clipping and for the contrast correction it's same as explained before.

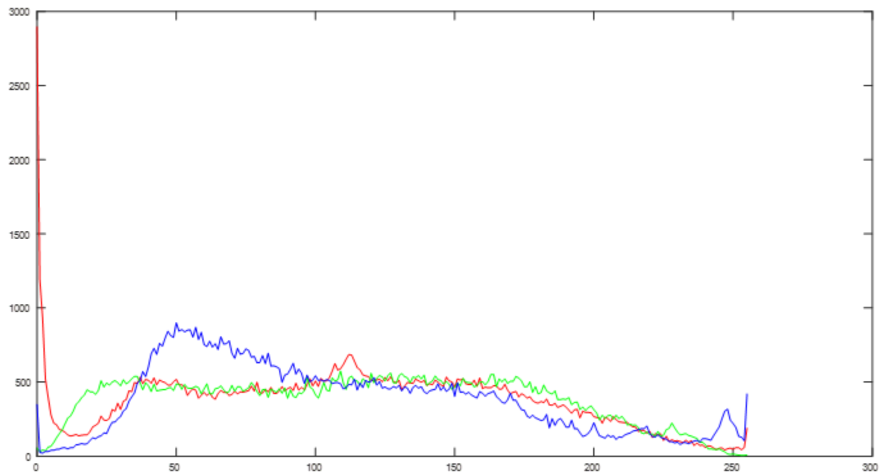


Figure 3-11: The separation of RGB histogram

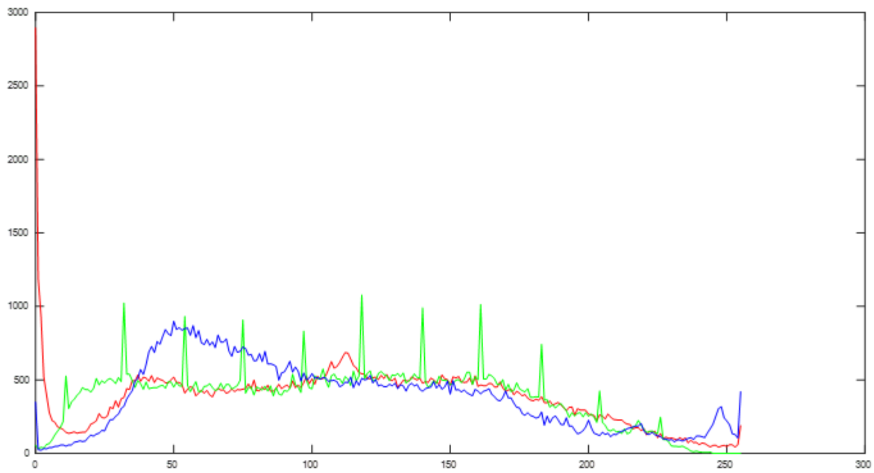


Figure 3-12: The histogram after recombined the new color channel

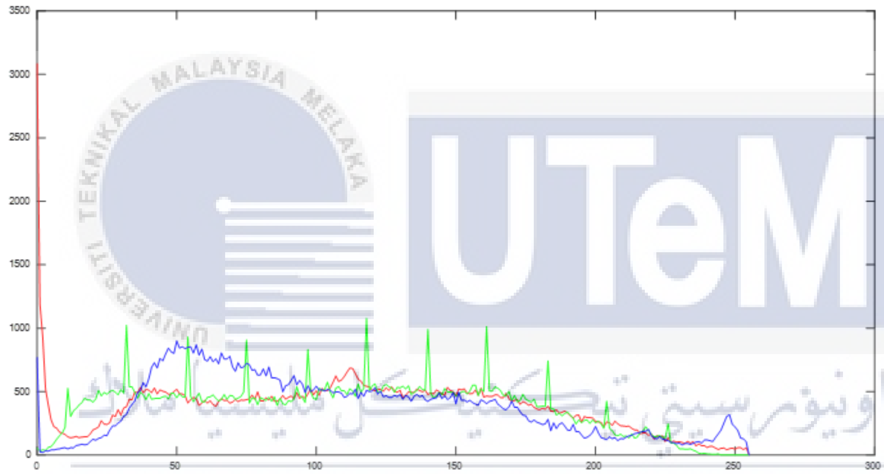


Figure 3-13: Histogram of each channel after Clipping

3.5 Program Flowchart

For the parameter setting, all the input image must have a same size. So, the input image must be resized to 360 x 270 which is the suitable size to run on the MATLAB. For the algorithms flow in this project, it is shown in the Figure 3-14.

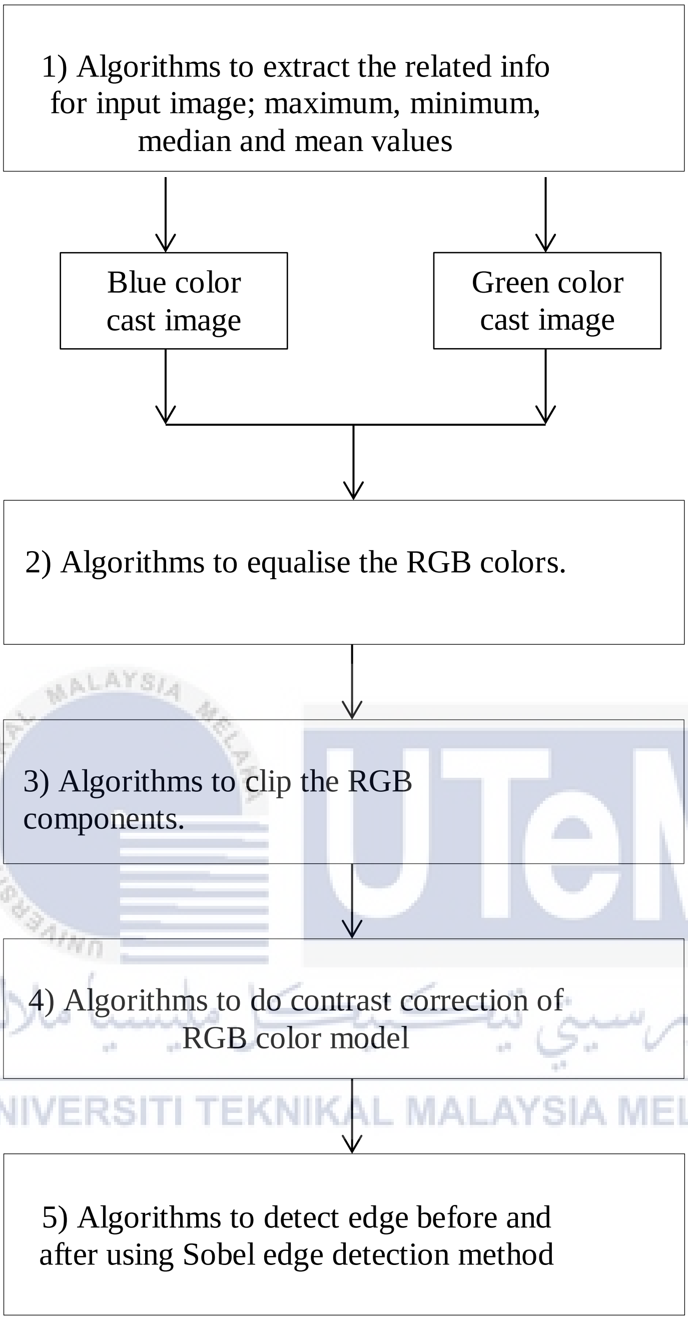


Figure 3-14: Algorithms flow

One example from the dataset images would show what happen to each step in Unsupervised Color Correction method. In step 1, the algorithms are developed to extract the related information such as maximum, minimum, median and mean values and the input image and its histogram is shown in Figure 3-15 and Figure 3-16. Then the images are categorized as bluish and greenish image to undergo the second step that is to equalise the RGB component. At this step, the dominant color cast is found, and it keeps the dominant color cast channel constant since in this case, it has high green color than red and blue. Therefore, the dominant color value is used to increase the other colors to make the image balanced in color. From the dominant color cast, two gain factors can be found out (Equation 2-7 and 2-8). The green color is set as the target mean and the red and blue component acts as multiplier to the target mean in order to achieve the balanced color image. UCM used these two colors channel to reduce the green cast on the affected image. Then the original green value, new red and green values is recombined and resulting image as in Figure 3-17 and histogram in Figure 3-18. The histogram starts to equalise at this time.

Next step is to do clipping at RGB components to remove few pixel values by selecting 0.2% and 99.8% as determinants to obtain a clearer image. The histogram is shown in Figure 3-22. The fourth step would only be applied to the inside pixel value after the clipping process. To get the better result, the contrast correction is applied to the upper, lower and both sides as explained in the literature review section for UCM. The image and histogram after all the step is shown in Figure 3-24 and Figure 3-25. From the visual aspect, the resulting image is improved in color and the histogram for RGB components is equally balanced. The last step for the algorithms flow is about the Sobel edge detection that can detect the edges before and after the implementation of UCM method as shown in Figure 3-26 and Figure 3-27.

Figure 3-15: Greenish input image

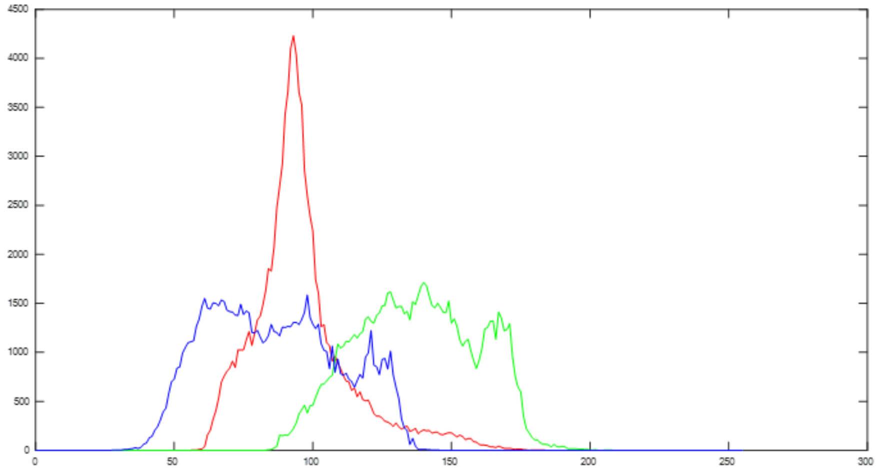


Figure 3-16: The separation of RGB histogram

Figure 3-17: Recombine the separate new color channel

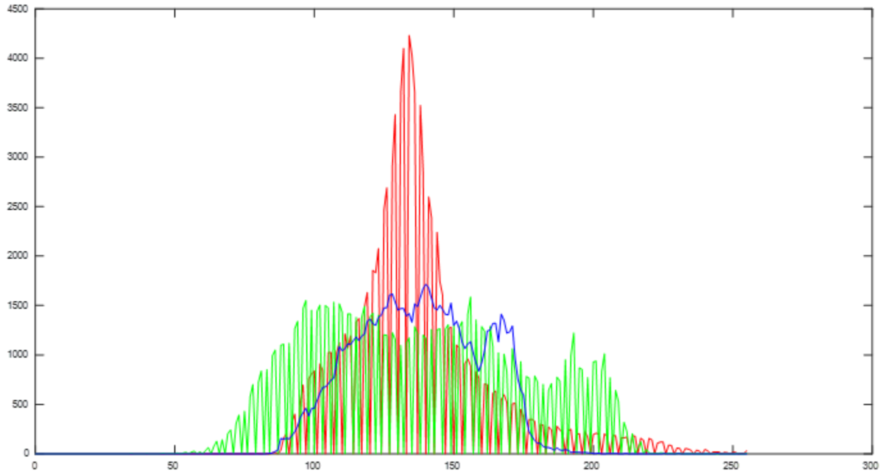


Figure 3-18: The histogram after recombined the new color channel

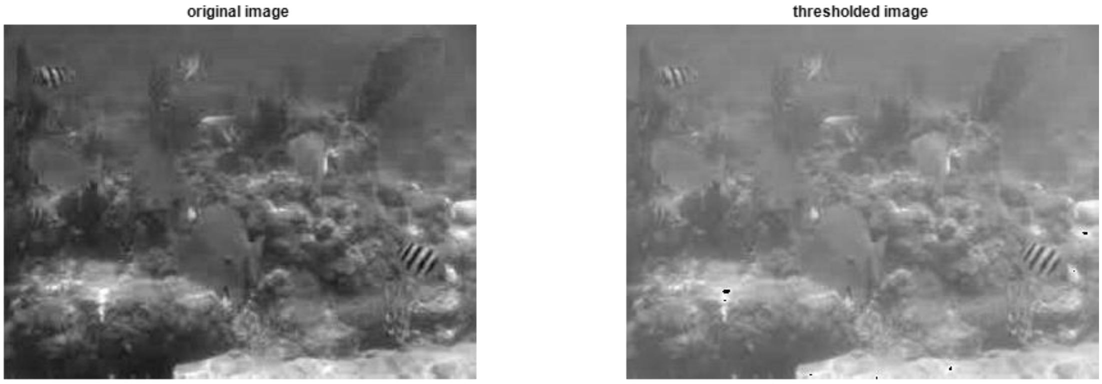


Figure 3-19: Clipping for Red histogram

Figure 3-20: Clipping for Green histogram

Figure 3-21: Clipping for Blue histogram

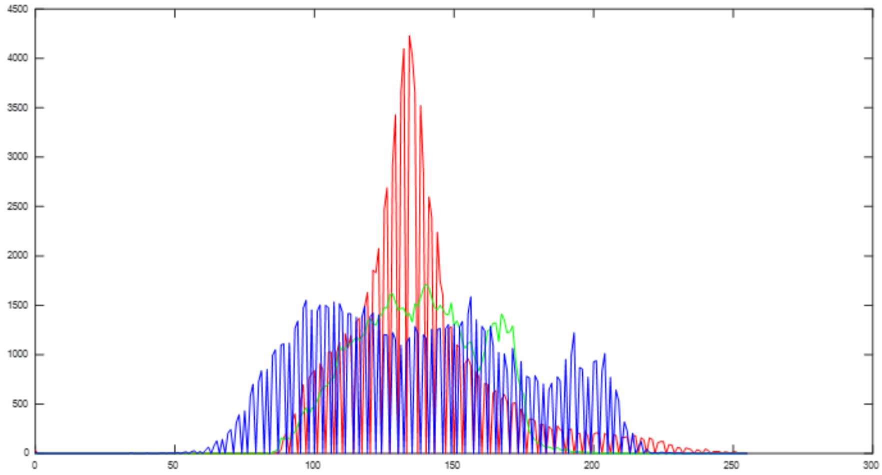


Figure 3-22: Histogram of each channel after Clipping

Figure 3-23: Image after the Clipping

Figure 3-24: Image after Contrast Correction

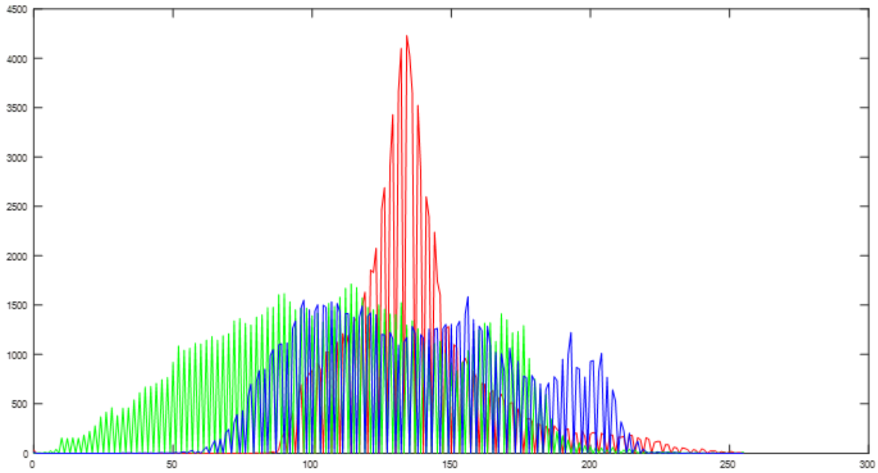


Figure 3-25: Histogram of each channel after Contrast Correction

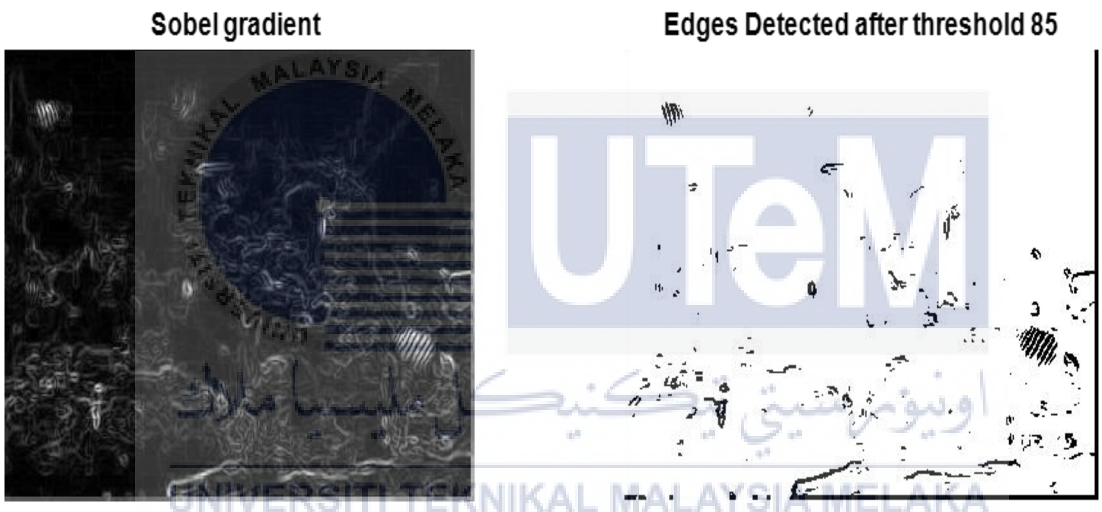


Figure 3-26: Edge detection for the original image

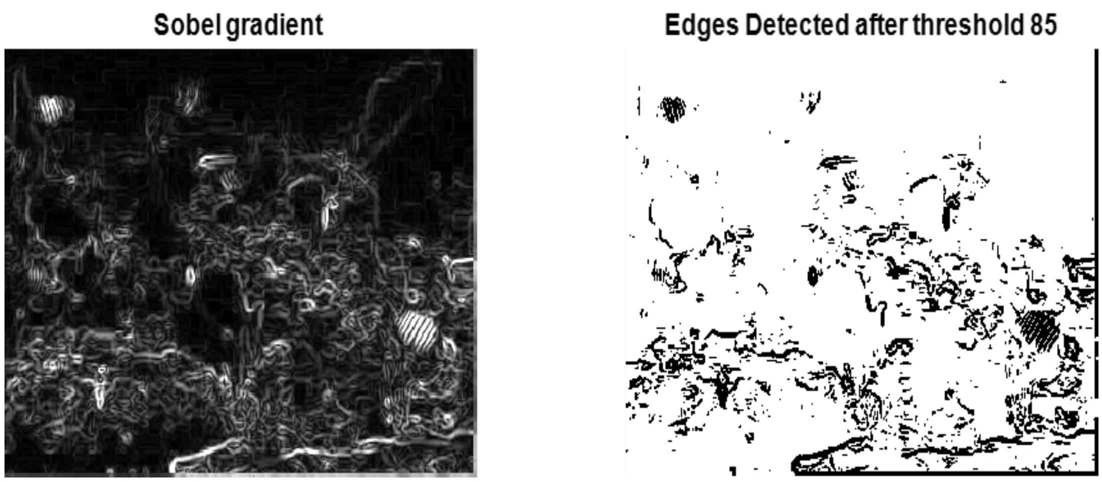


Figure 3-27: Edge detection of the image after color improvement

3.6 Conclusion

Unsupervised Color Correction method has been developed using MATLAB and is tested using 3 different type of images (bluish, greenish and normal). Other than visual inspection, the number of edges of the original images and the number of edges after using the UCM method will be used to validate the performance of the UCM method.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter shows and analysed the results of underwater images after it have been improved by Unsupervised Color Correction technique. The result was evaluated by using edge detection to determine whether the method is suitable or not and to be used to improve the color quality of underwater images.

4.2 Result and Discussion

4.2.1 Result and discussion for the bluish input image

From the steps explained in the methodology part, the result has been improved visually such in Figure 4-2. The colors of the RGB looks balanced compared to the original image in Figure 4-1 and the red color has been increased thus reduced the color cast that is blue color. The shape of corals looks to be clearer after UCM is implemented. The output histogram for the RGB components is equally distributed such in Figure 4-4 compared to the original histogram in Figure 4-3. For this input image, the measured valued of edge detection for the original image is 557 010 and it has increased to 5 896 952. There is about 90.55% improvement from the both values. The visual of edge detection at Figure 4-6 is better compared to Figure 4-5. It is proved that UCM is suitable for this type of dataset image. For the overall dataset images, the average of the result edge detection for the original image is approximately 1 684 733 and for the image after the color improvement by UCM is approximately 5 922 525. From the averages, all the dataset bluish images that have been tested is improved significantly with the percentage of 71.55%.

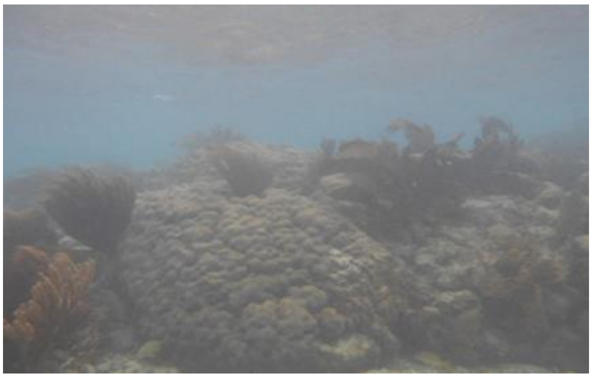


Figure 4-1: The bluish input image

Figure 4-2: Image after Contrast Correction

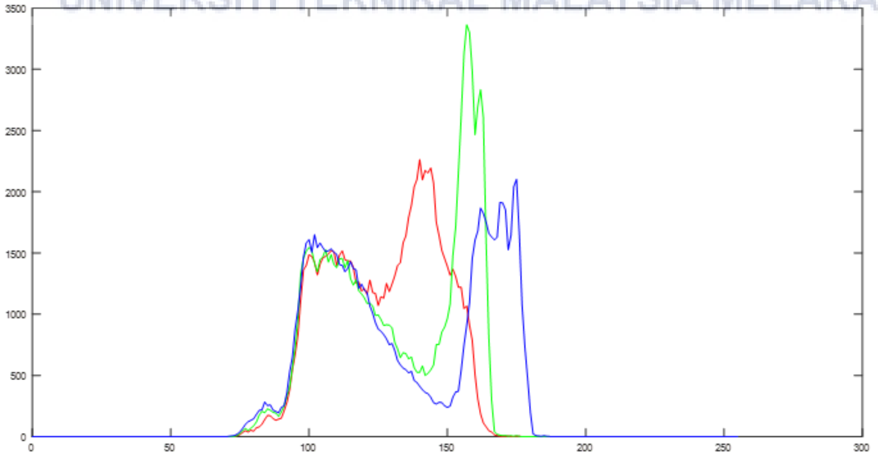


Figure 4-3: The histogram for the input image

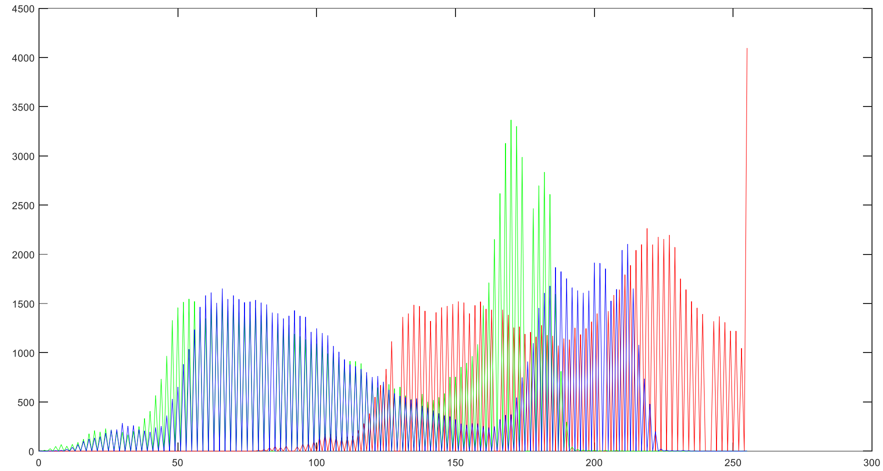


Figure 4-4: Histogram of each channel after Contrast Correction

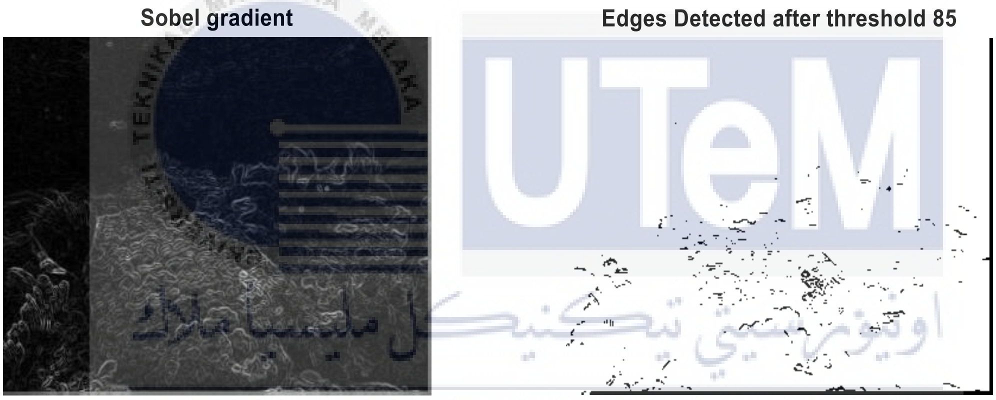


Figure 4-5: Edge detection for the original image

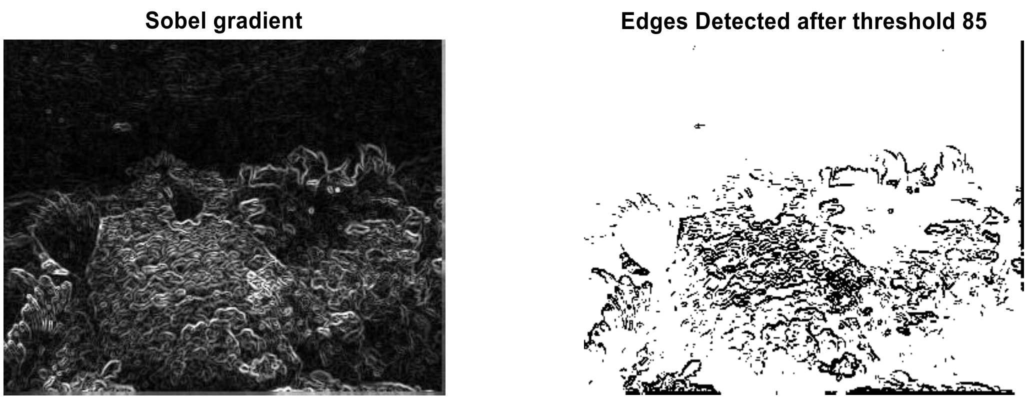


Figure 4-6: Edge detection of the image after color improvement

Table 4-1: Edge detection before and after using UCM method

Input image	Edge detected before color improvement	Edge detected after color improvement
1	557 010	5 896 952
2	619 436	7 973 516
3	4 028 666	7 160 789
4	282 586	1 544 132
5	453 116	4 666 638
6	1 351 408	9 104 991
7	965 703	6 173 530
8	1 265 244	4 800 493
9	2 222 642	6 153 645
10	2 692 439	6 091 839
11	2 093 436	4 566 194
12	3 685 108	6 937 577
TOTAL	20,216,794	71,070,296

Average edge detection:

Before: 20,216,794 / 12 = 1 684 733

After: 71,070,296 / 12 = 5 922 525

4.2.2 Result and discussion for the greenish input image

Compared to the dataset image in Figure 4-7, the result at Figure 4-8 has improved visually due to the RGB colors balancing and the contrast correction to RGB color model. The value of green component act as the color cast has been reduced and the red component is increased in the contrast correction. The output histogram is equally distributed as shown in Figure 4-10 compared to the histogram of the input image at Figure 4-9. Figure 4-11 shows the edge detection for the original image and Figure 4-12 shows the edge detection for the images that its color has been improved. The edge is improved from 424 629 to 1 507 934 and more edges can be seen in Figure 4-12. The percentage of improvement is 71.84%. For the other greenish input images, all of it also shows that it is improved a lot in color in Table 4-2 and after the averaged edge is calculated, the average number of edges increase from 1 785 437 to 3 624 267. There is significantly improvement measured that is 50.74% and it shows that UCM is suitable to use for this type of images.

Figure 4-7: The greenish input image

Figure 4-8: Image after Contrast Correction

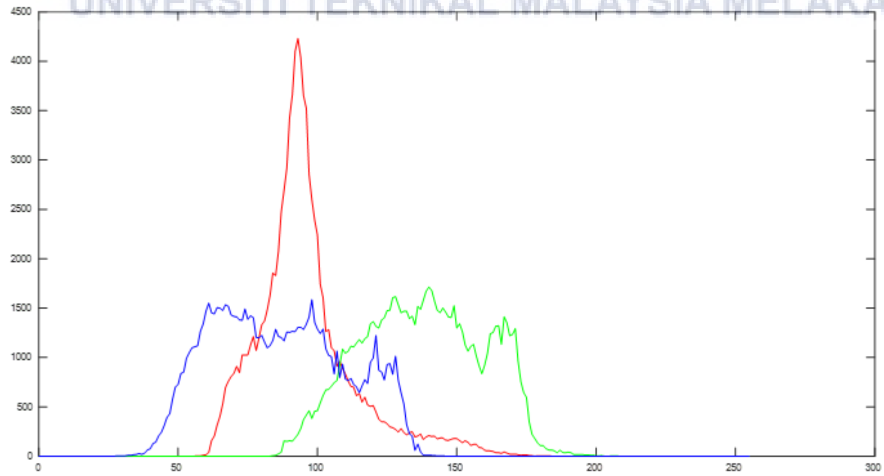


Figure 4-9: The histogram for the input image

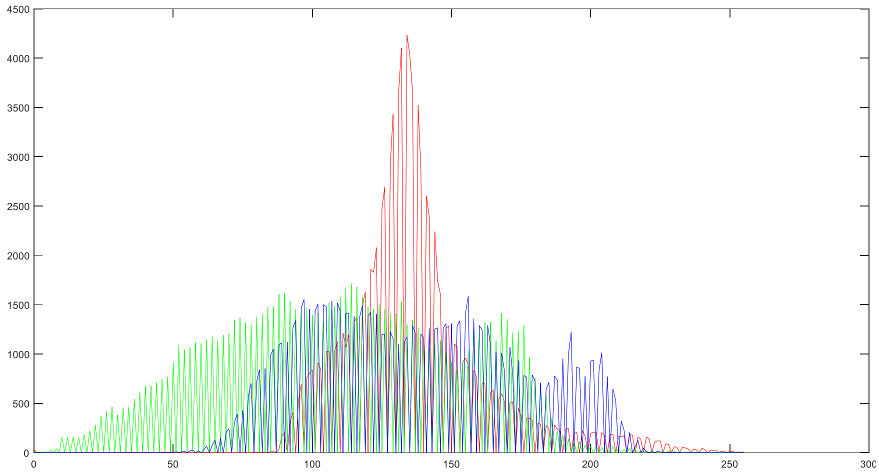


Figure 4-10: Histogram of each channel after Contrast Correction

Figure 4-11: Edge detection for the original image

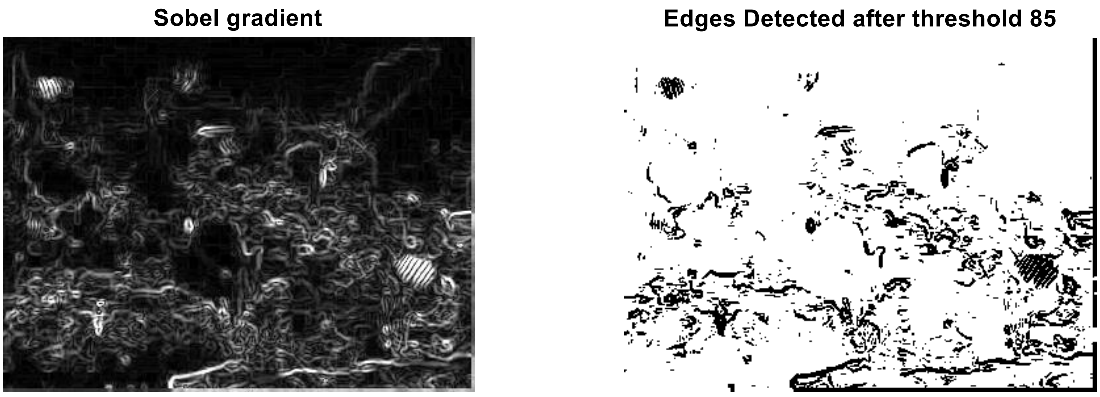


Figure 4-12: Edge detection of the image after color improvement

Table 4-2: Edge detection before and after using UCM method

Input image	Edge detected before color improvement	Edge detected after color improvement
1	2 565 869	8 902 766
2	424 629	1 507 934
3	6 020 016	1 077 294
4	3 279 306	5 268 388
5	179 490	1 326 467
6	1 023 501	3 103 231
7	297 116	1 154 694
8	1 576 896	2 677 683
9	4 861 065	7 345 137
10	581 199	802 573
11	438 820	9 610 967
12	177 337	714 074
TOTAL	21,425,244	43,491,208

Average:

Before: 21,425,244/ 12 = 1 785 437

After: 43,491,208/ 12 = 3 624 267

4.2.3 Result and discussion for the normal input image

For the normal image, even after the contrast correction stage, from the visual aspect, the original image in Figure 4-13 and the result in Figure 4-14 didn't have any difference. The green component in Figure 4-16 spikes up from the input image's histogram in Figure 4-15 and it means that the value of pixels in green component is increased but the intensity is still the same. The edge detection method was performed and for the original image, the edge detection is 9 029 141 but for the result image, it decreases to 8 824 577. The difference value of reduction is about 2.27%. All of the others input images also faced the same situation as shown in Table 4-3 and the average of the edge detection for the original images is reduced from 8 334 318 to 7 488 759. And the percentage loss is 10.15%. It can be interpreted as the original image had lost some of their edges due to UCM and it is not suitable for normal images to undergo this color improvement method.

Figure 4-13: The normal input image

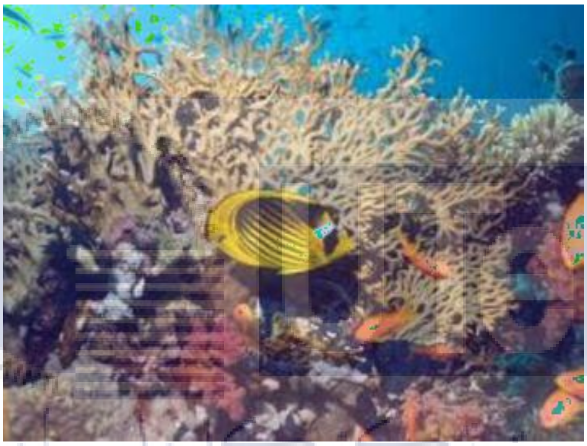


Figure 4-14: Image after Contrast Correction

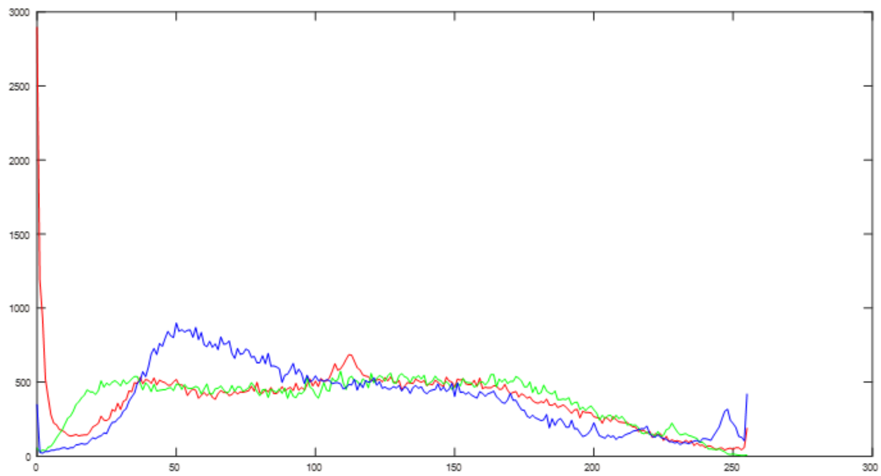


Figure 4-15: The histogram for the input image

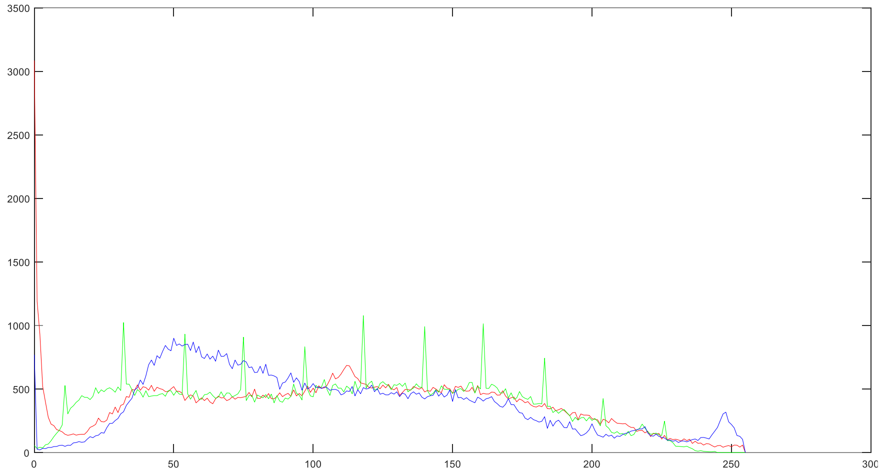


Figure 4-16: Histogram of each channel after Contrast Correction

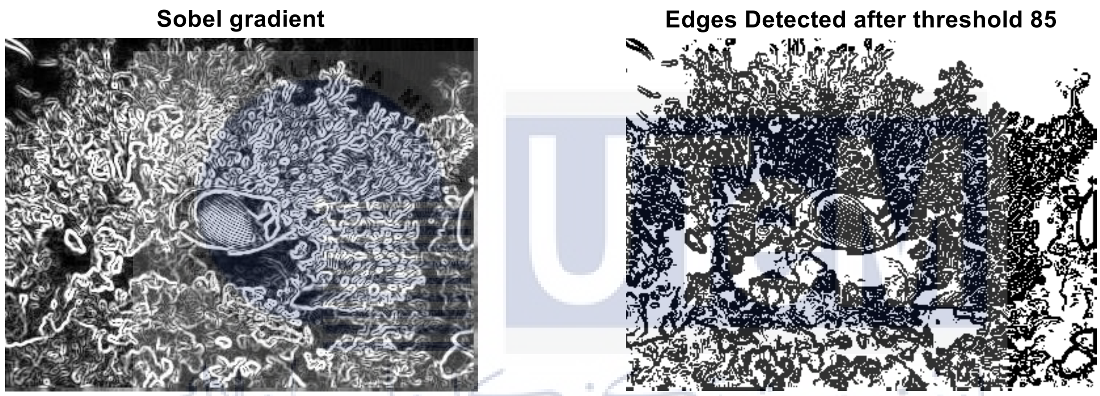


Figure 4-17: Edge detection for the original image

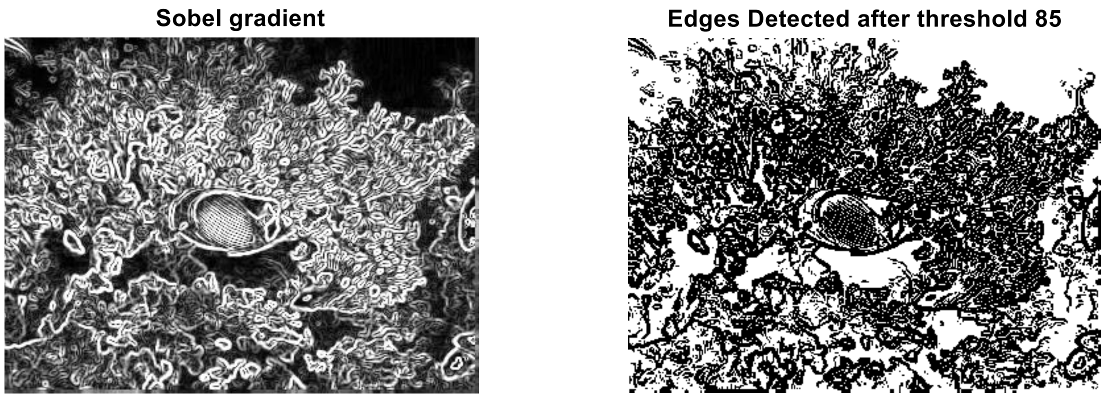


Figure 4-18: Edge detection of the image after color improvement

Table 4-3: Edge detection before and after using UCM method

Input image	Edge detected before color improvement	Edge detected after color improvement
1	6 113 388	5 240 249
2	11 356 245	11 221 364
3	8 657 134	8 427 896
4	9 029 141	8 824 577
5	5 006 615	4 797 743
6	6 976 885	6 237 286
7	9 748 530	7 872 307
8	8 464 050	7 392 020
9	8 783 715	8 057 558
10	6 782 131	5 611 892
11	10 000 906	7 180 042
12	9 093 076	9 002 173
TOTAL	100 011 816	89 865 107

Average:

Before: 100 011 816 / 12 = 8 334 318

After: 89 865 107 / 12 = 7 488 759

4.3 Chapter Conclusion

As conclusion for this chapter, it can be concluded that the UCM method is basically worked out for certain types of images that have these criteria; have maximum of green or blue color intensity. It cannot be used on the image that have equally balance of RGB values because it can reduce the color intensity and quality of the images.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter contains a brief summary of the entire work, including methods, results and major conclusions /recommendations arising from the work. Weaknesses, shortcomings and strengths of the project are presented. Recommendations for future work may also be included together with contributions of project.

5.2 Conclusion

The underwater images have poor color and contrast because of some factors such as the limited wavelength of the penetration light and density of the seawater. It needs the image enhancement process to make it look better. From the experimental results, it has been demonstrated that this UCM method is suitable to remove undesired color casts that is greenish-blue components in underwater imaging. The Unsupervised Color Correction method efficiently had removed the bluish and greenish color cast, improved the lowest color component that is red color and true color of the input images.

In order to evaluate whether the images are really improved on the color, the Sobel edge detection was used, and it can be seen visually and quantitatively at the output images. By using Unsupervised Color Correction method, the improvement of the input images' color also increased the visibility of the objects, thus improving object recognition in the image. This method can be implemented by the scientist to collect the data to know the coral's health and its current population. Even though the result in this project meet the expectation from the literature review, however, there are some weakness by using Unsupervised Color Correction method in this project. The weakness is even the color of the image is improved, but it cannot solve for the noise problems such as scattering problem due to the particles at the underwater and this method is limited only to the underwater images that is bluish or greenish.

5.3 Future Works

For future improvement of the project, a new algorithm can be proposed for reducing the noise and haze that can be appeared on certain images that captured at the underwater. By doing that, a clearer image can be obtain and it will ease the researcher's task to observe and tracking the health of the corals and to find a new or improved the way for navigation under the sea. Other suggestion is to develop a new algorithm for the area that is in reddish and input image that has higher values in red component because this project is only focused to the underwater area that is blue and green color is the color cast to the image. By implementing this suggestion, a clearer and more quality images can be obtained by using Unsupervised Color Correction method.

REFERENCES

[1] R. A. Pramunendar, G. F. Shidik, and C. Supriyanto, “Auto Level Color Correction for Underwater Image Matching Optimization,” no. June 2014, pp. 17–23, 2013.

[2] J Floor Anthoni, “Water and light,” 2000. [Online]. Available: http://www.seafriends.org.nz/phgraph/water.htm. [Accessed: 05-Dec-2018].

[3] C. Wilkinson, “Status of the Coral reefs in the world,” 2008.

[4] “Picture of Gebel El Rosas, Marsa Alam - TripAdvisor.” [Online]. Available: https://www.tripadvisor.co.za/LocationPhotoDirectLink-g311425-d3354585-i228244660-Gebel_El_Rosas-Marsa_Alam_Red_Sea_and_Sinai.html. [Accessed: 11-Dec-2018].

[5] “Removing blue-green cast from underwater photos: Retouching Forum: Digital Photography Review.” [Online]. Available: https://www.dpreview.com/forums/thread/2981054. [Accessed: 11-Dec-2018].

[6] “underwater photography | Adventures in Loom-Science,” 2013. [Online]. Available: https://loomsci.wordpress.com/tag/underwater-photography/. [Accessed: 05-Dec-2018].

[7] “underwater images - turtle.” [Online]. Available: https://www.worldatlas.com.

[8] K. K. Eric Madeja, “Coral/Biodiversity,” 2012. [Online]. Available: http://www.coraltriangle.org/biodiversity/files.

[9] Xueyang Fu 傅雪阳, “Two-step approach for single underwater image enhancement.”

[10] M. Manoach, H. Netz, M. M. Aygen, T. Pauker, and S. Gitter, “A Brief History of Pixel,” *Symp. Electron. Imaging*, vol. 13, no. 4, pp. 371–376, 1977.

[11] J. H. Lee, “Handling digital images for publication,” *Sci. Ed.*, vol. 1, no. 2, pp. 58–61, 2014.

[12] B. Sheets, K. and W.Jones, “citelr,” *Image Analysis - Learn ImageJ*. .

[13] “The range of intensity values from 0 (black) to 255 (white) | Download Scientific Diagram.” [Online]. Available: https://www.researchgate.net/figure/The-range-of-intensity-values-from-0-black-to-255-white_fig2_260571873. [Accessed: 05-Dec-2018].

[14] A. S. Perceived and B. Y. The, “Experiments on colour,” vol. XXI.

[15] A. N. V. Plataniotis, K.N., “Color Image Processing and Application.”

[16] E. W. R.FISHER, S.PERKINS, A.WALKER, “Image Histogram.” .

[17] “histogram-cheat-sheet.jpg (1920×1080).” [Online]. Available: http://community.clickersart.com/wp-content/uploads/2016/04/histogram-cheat-sheet.jpg. [Accessed: 05-Dec-2018].

[18] K. He, “Single Image Haze Removal Using Dark Channel Prior,” pp. 1956–1963, 2009.

[19] J. Y. Chiang and Y. C. Chen, “Underwater image enhancement by wavelength compensation and dehazing,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1756–1769, 2012.

[20] R. T. Tan, “Visibility in Bad Weather from a Single Image.”

[21] P. Philadelphia, “A Spatial Processor Model for Object Colour Perception.”

[22] M. Vision and M. Ebner, “Color Constancy Based on Local Space Average Color,” vol. 11, no. 5, pp. 283–301, 2009.

[23] A. Galdran, D. Pardo, A. Picón, and A. Alvarez-Gila, “Automatic Red-Channel underwater image restoration,” *J. Vis. Commun. Image Represent.*, vol. 26, no. April, pp. 132–145, 2015.

[24] R. Jaiswal, A. G. Rao, and H. P. Shukla, “Image Enhancement Techniques Based on Histogram Equalization,” pp. 69–78.

[25] B. Hu and B. Zheng, “Underwater image color correct in extremely poor visibility,” vol. 0, 2013.

[26] K. Iqbal, M. Odetayo, A. James, R. A. Salam, and A. Z. H. Talib, “Enhancing the low quality images using unsupervised colour correction method,” *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, pp. 1703–1709, 2010.

APPENDICES

APPENDIX A GANTT CHART FOR FINAL YEAR PROJECT 2

Tasks	September				October				November				December				January				February				March				April				May				June			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
PSM 1 Briefing																																								
PSM Registration																																								
Literature Review																																								
Developed methodology using MATLAB																																								
Obtained the dataset images																																								
PSM 1 Report Writing																																								
PSM 1 Draft Report Submission																																								
PSM 1 Draft Report Correction																																								
Presentation																																								
PSM 1 Report Resubmission																																								
Developed the algorithms in MATLAB																																								
Data Analysis 2																																								
PSM 2 Report Writing																																								
PSM 2 Draft Report Submission																																								
PSM 2 Draft Report Correction																																								
Presentation																																								
PSM 2 Report Resubmission																																								

APPENDIX B THE DATASET OF BLUISH, GREENISH AND NORMAL IMAGES







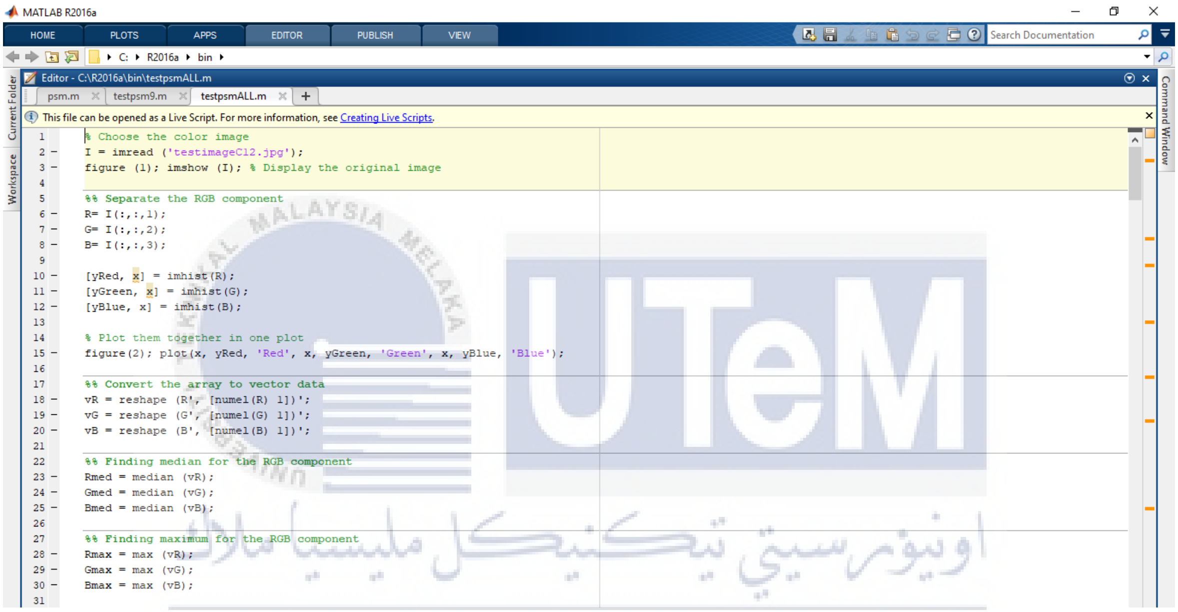




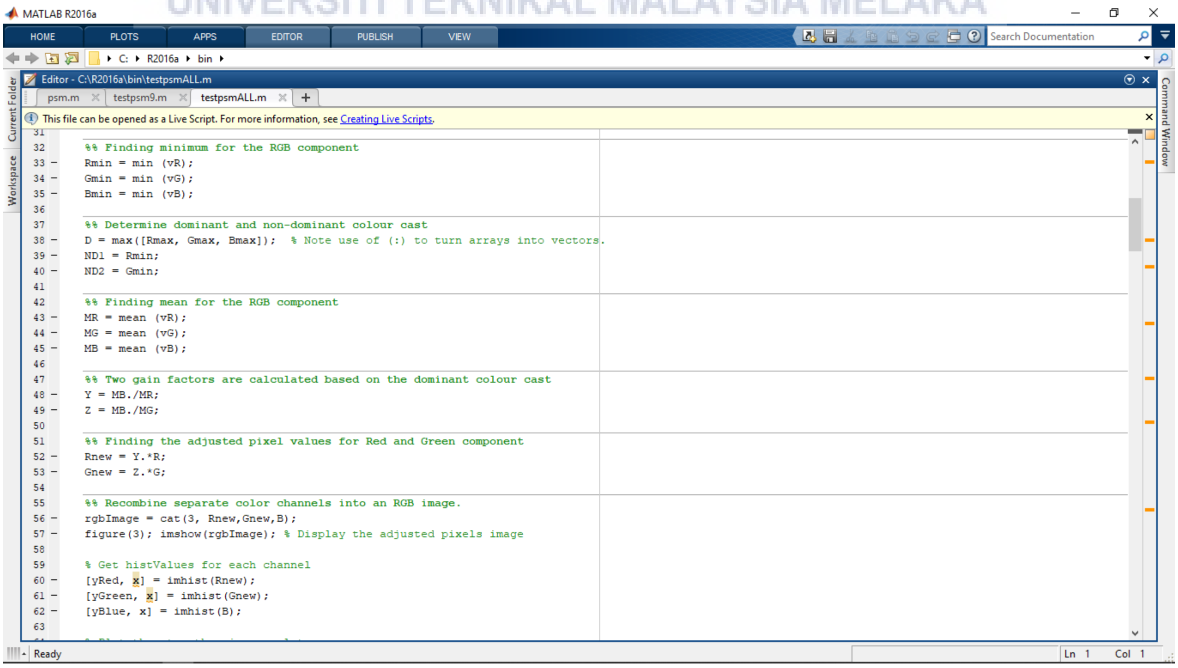

© Chris Harvey-Clark

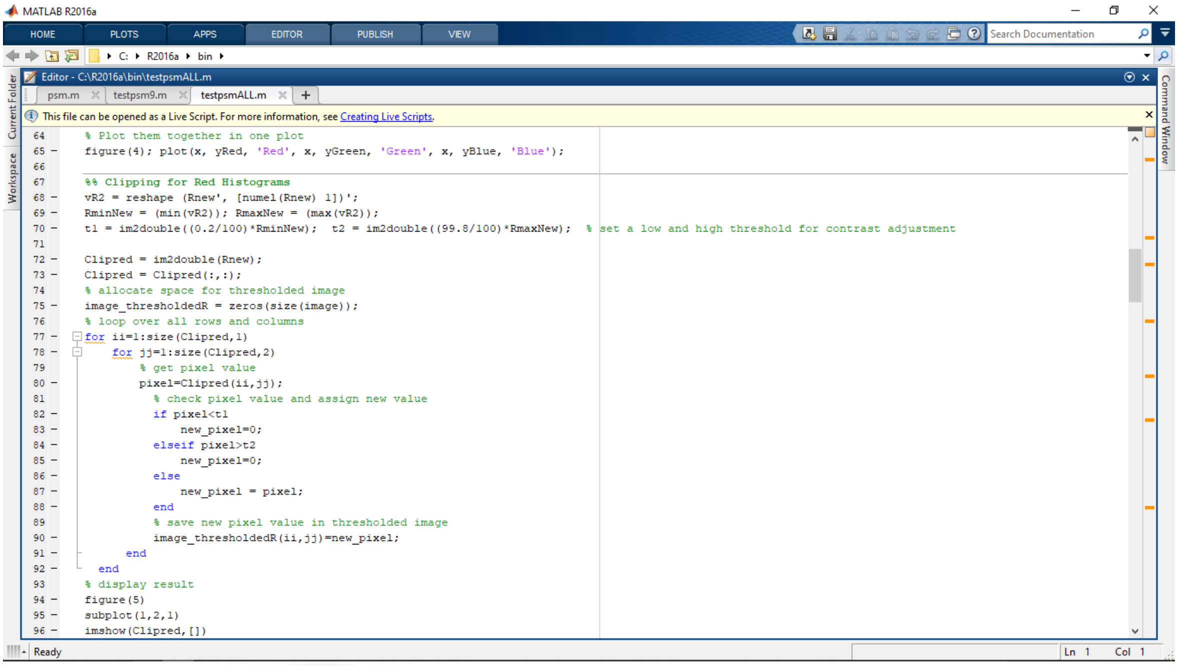
APPENDIX C PROJECT CODE IN MATLAB



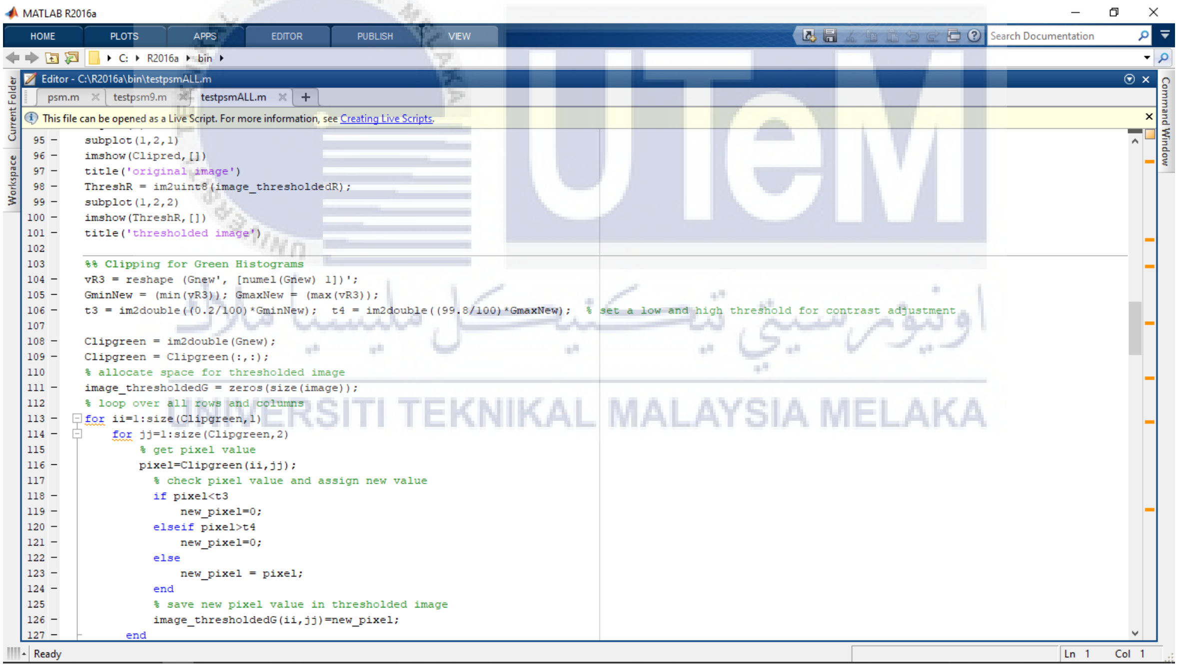
```matlab
% Choose the color image
I = imread ('testimageC12.jpg');
figure (1); imshow (I); % Display the original image

%% Separate the RGB component
R= I(:,:,1);
G= I(:,:,2);
B= I(:,:,3);

[yRed, x] = imhist(R);
[yGreen, x] = imhist(G);
[yBlue, x] = imhist(B);

% Plot them together in one plot
figure(2); plot(x, yRed, 'Red', x, yGreen, 'Green', x, yBlue, 'Blue');

%% Convert the array to vector data
vR = reshape (R', [numel(R) 1])';
vG = reshape (G', [numel(G) 1])';
vB = reshape (B', [numel(B) 1])';

%% Finding median for the RGB component
Rmed = median (vR);
Gmed = median (vG);
Bmed = median (vB);

%% Finding maximum for the RGB component
Rmax = max (vR);
Gmax = max (vG);
Bmax = max (vB);
```

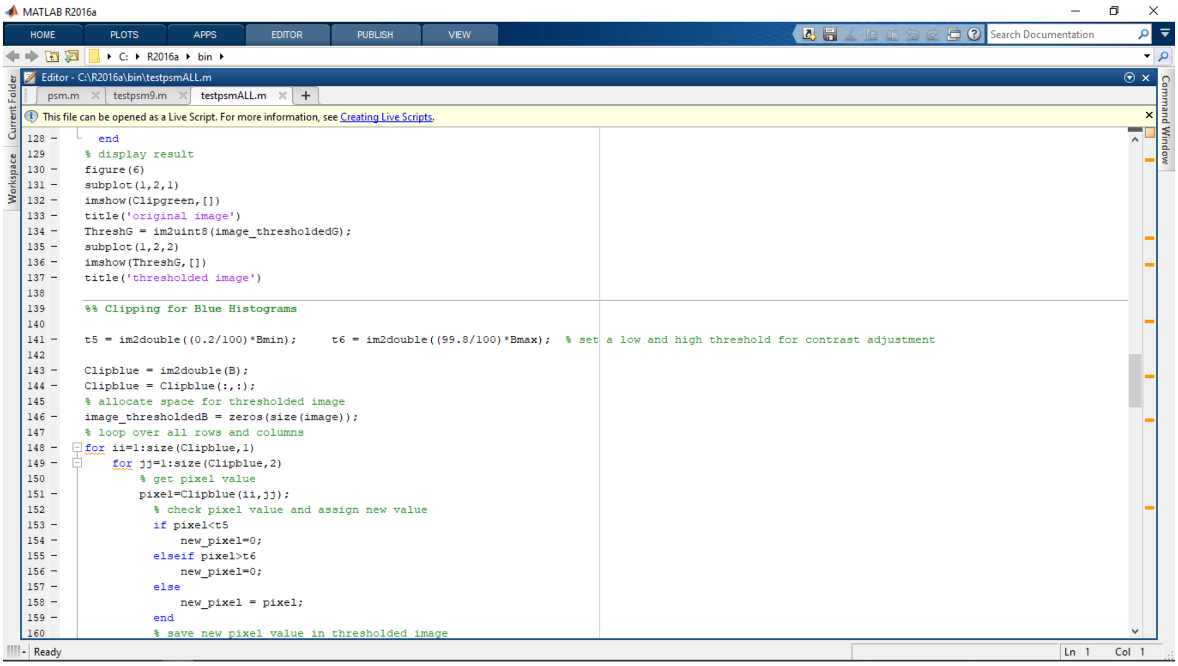
```matlab
%% Finding minimum for the RGB component
Rmin = min (vR);
Gmin = min (vG);
Bmin = min (vB);

%% Determine dominant and non-dominant colour cast
D = max([Rmax, Gmax, Bmax]);  % Note use of (:) to turn arrays into vectors.
ND1 = Rmin;
ND2 = Gmin;

%% Finding mean for the RGB component
MR = mean (vR);
MG = mean (vG);
MB = mean (vB);

%% Two gain factors are calculated based on the dominant colour cast
Y = MB./MR;
Z = MB./MG;

%% Finding the adjusted pixel values for Red and Green component
Rnew = Y.*R;
Gnew = Z.*G;

%% Recombine separate color channels into an RGB image.
rgbImage = cat(3, Rnew,Gnew,B);
figure(3); imshow(rgbImage); % Display the adjusted pixels image

% Get histValues for each channel
[yRed, x] = imhist(Rnew);
[yGreen, x] = imhist(Gnew);
[yBlue, x] = imhist(B);
```



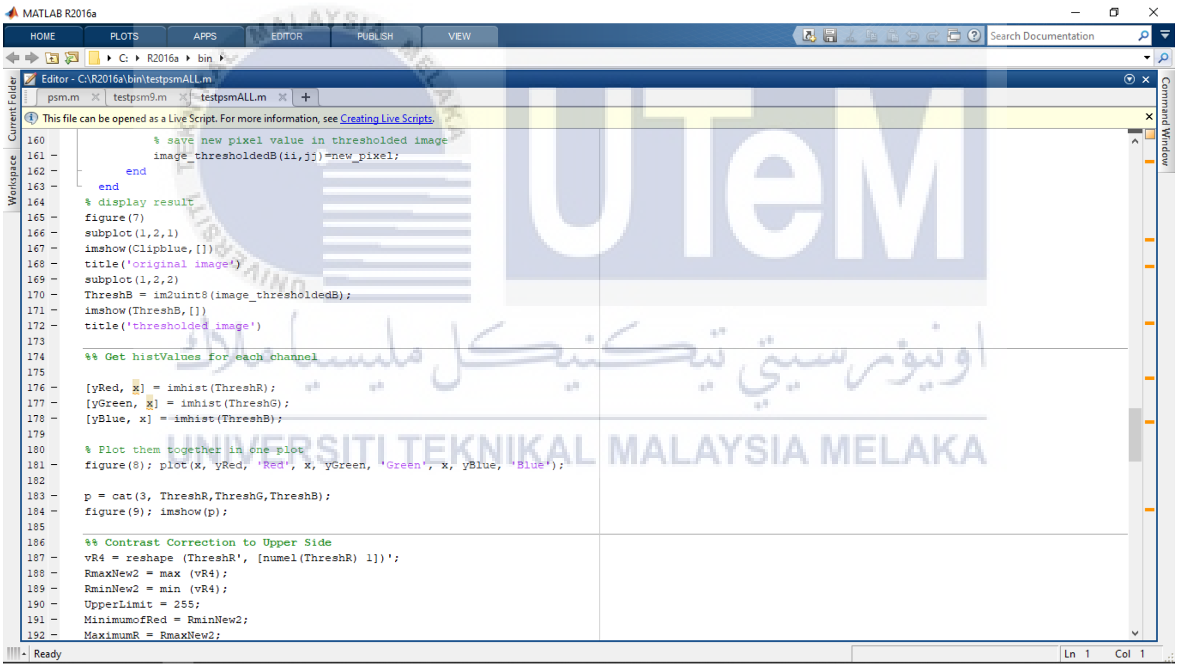
```
% Plot them together in one plot
figure(4); plot(x, yRed, 'Red', x, yGreen, 'Green', x, yBlue, 'Blue');

%% Clipping for Red Histograms
vR2 = reshape (Rnew', [numel(Rnew) 1])';
RminNew = (min(vR2)); RmaxNew = (max(vR2));
t1 = im2double((0.2/100)*RminNew);  t2 = im2double((99.8/100)*RmaxNew);  % set a low and high threshold for contrast adjustment

Clipred = im2double(Rnew);
Clipred = Clipred(:,:);
% allocate space for thresholded image
image_thresholdedR = zeros(size(image));
% loop over all rows and columns
for ii=1:size(Clipred,1)
    for jj=1:size(Clipred,2)
        % get pixel value
        pixel=Clipred(ii,jj);
          % check pixel value and assign new value
          if pixel<t1
              new_pixel=0;
          elseif pixel>t2
              new_pixel=0;
          else
              new_pixel = pixel;
          end
          % save new pixel value in thresholded image
          image_thresholdedR(ii,jj)=new_pixel;
    end
end
% display result
figure(5)
subplot(1,2,1)
imshow(Clipred,[])
```



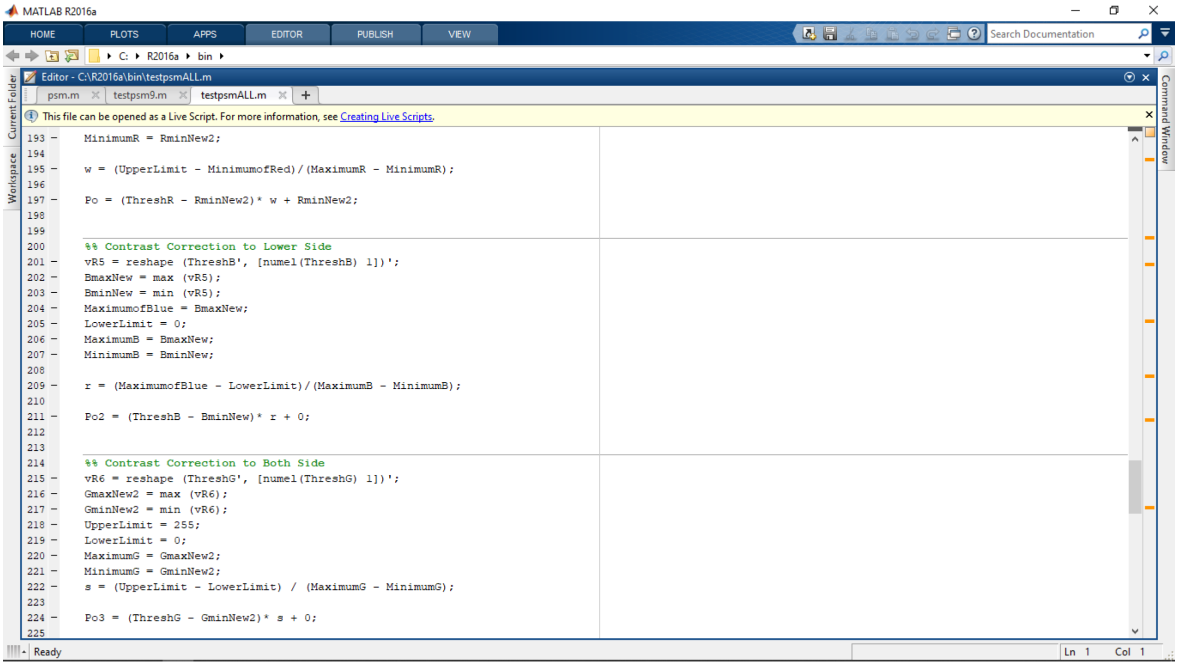
```
subplot(1,2,1)
imshow(Clipred,[])
title('original image')
ThreshR = im2uint8(image_thresholdedR);
subplot(1,2,2)
imshow(ThreshR,[])
title('thresholded image')

%% Clipping for Green Histograms
vR3 = reshape (Gnew', [numel(Gnew) 1])';
GminNew = (min(vR3)); GmaxNew = (max(vR3));
t3 = im2double((0.2/100)*GminNew);  t4 = im2double((99.8/100)*GmaxNew);  % set a low and high threshold for contrast adjustment

Clipgreen = im2double(Gnew);
Clipgreen = Clipgreen(:,:);
% allocate space for thresholded image
image_thresholdedG = zeros(size(image));
% loop over all rows and columns
for ii=1:size(Clipgreen,1)
    for jj=1:size(Clipgreen,2)
        % get pixel value
        pixel=Clipgreen(ii,jj);
          % check pixel value and assign new value
          if pixel<t3
              new_pixel=0;
          elseif pixel>t4
              new_pixel=0;
          else
              new_pixel = pixel;
          end
          % save new pixel value in thresholded image
          image_thresholdedG(ii,jj)=new_pixel;
    end
```



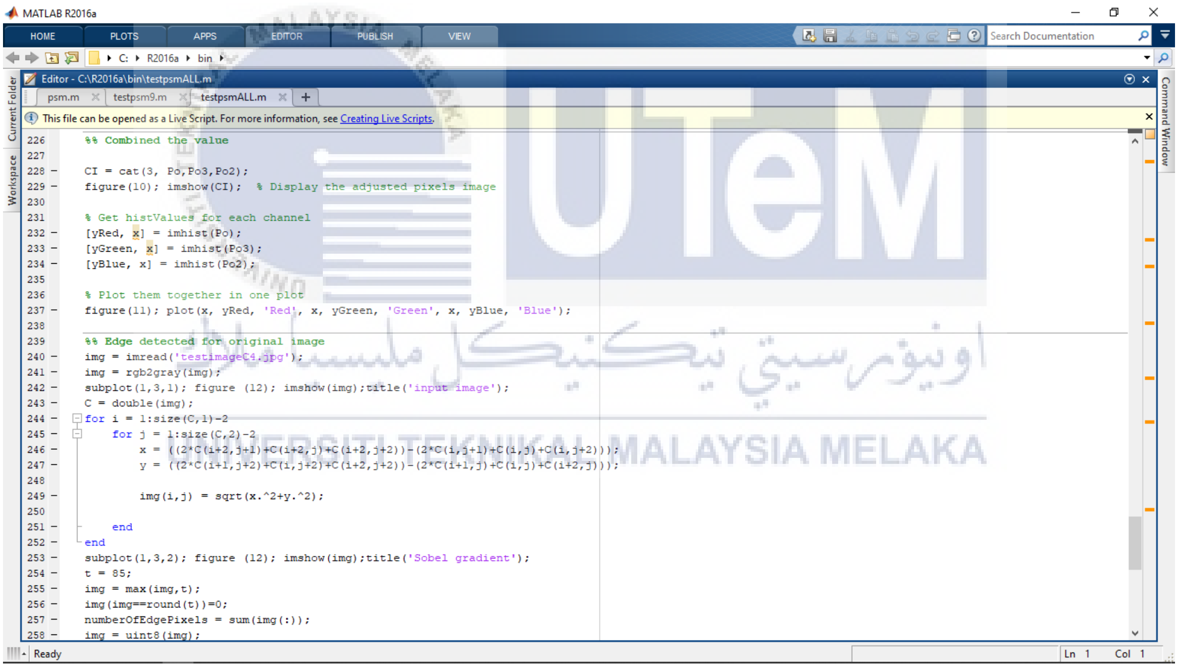
MATLAB R2016a
HOME PLOTS APPS EDITOR PUBLISH VIEW
Search Documentation
C: ▸ R2016a ▸ bin
Editor - C:\R2016a\bin\testpsmALL.m
psm.m testpsm9.m testpsmALL.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
end
% display result
figure(6)
subplot(1,2,1)
imshow(Clipgreen,[])
title('original image')
ThreshG = im2uint8(image_thresholdedG);
subplot(1,2,2)
imshow(ThreshG,[])
title('thresholded image')
%% Clipping for Blue Histograms
t5 = im2double((0.2/100)*Bmin); t6 = im2double((99.8/100)*Bmax); % set a low and high threshold for contrast adjustment
Clipblue = im2double(B);
Clipblue = Clipblue(:,:);
% allocate space for thresholded image
image_thresholdedB = zeros(size(image));
% loop over all rows and columns
for ii=1:size(Clipblue,1)
for jj=1:size(Clipblue,2)
% get pixel value
pixel=Clipblue(ii,jj);
% check pixel value and assign new value
if pixel<t5
new_pixel=0;
elseif pixel>t6
new_pixel=0;
else
new_pixel = pixel;
end
% save new pixel value in thresholded image
Ready
Ln 1 Col 1



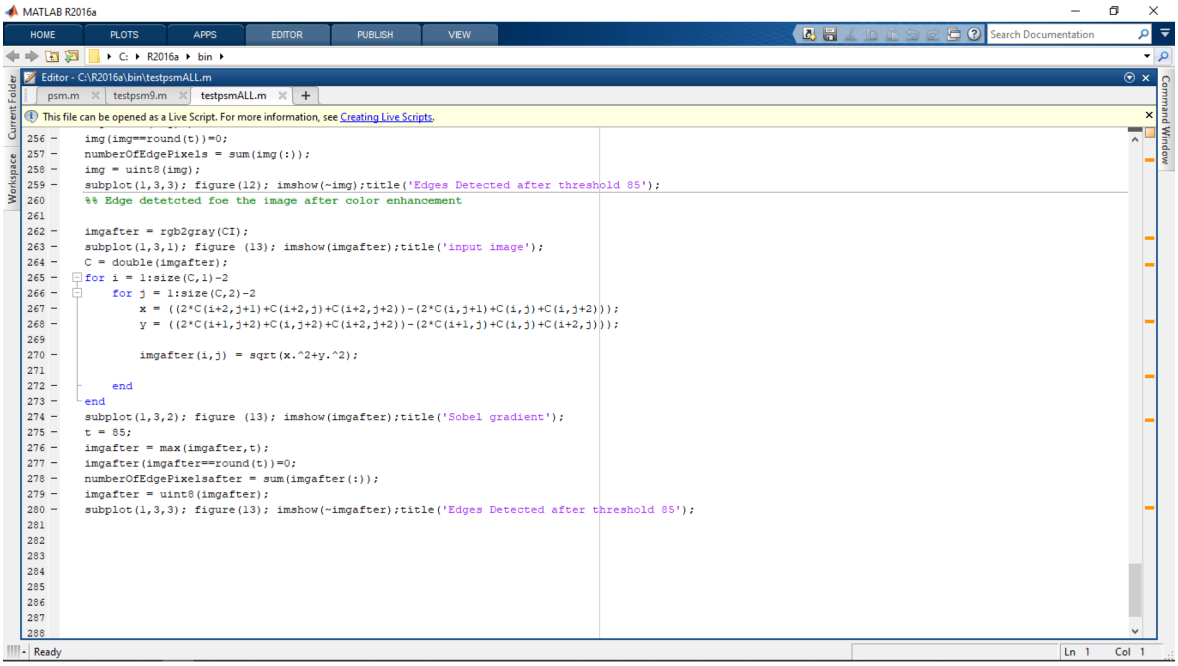
MATLAB R2016a
HOME PLOTS APPS EDITOR PUBLISH VIEW
Search Documentation
C: ▸ R2016a ▸ bin
Editor - C:\R2016a\bin\testpsmALL.m
psm.m testpsm9.m testpsmALL.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
% save new pixel value in thresholded image
image_thresholdedB(ii,jj)=new_pixel;
end
end
% display result
figure(7)
subplot(1,2,1)
imshow(Clipblue,[])
title('original image')
subplot(1,2,2)
ThreshB = im2uint8(image_thresholdedB);
imshow(ThreshB,[])
title('thresholded image')
%% Get histValues for each channel
[yRed, x] = imhist(ThreshR);
[yGreen, x] = imhist(ThreshG);
[yBlue, x] = imhist(ThreshB);
% Plot them together in one plot
figure(8); plot(x, yRed, 'Red', x, yGreen, 'Green', x, yBlue, 'Blue');
p = cat(3, ThreshR,ThreshG,ThreshB);
figure(9); imshow(p);
%% Contrast Correction to Upper Side
vR4 = reshape (ThreshR', [numel(ThreshR) 1])';
RmaxNew2 = max (vR4);
RminNew2 = min (vR4);
UpperLimit = 255;
MinimumofRed = RminNew2;
MaximumR = RmaxNew2;
Ready
Ln 1 Col 1



```
MinimumR = RminNew2;

w = (UpperLimit - MinimumofRed)/(MaximumR - MinimumR);

Po = (ThreshR - RminNew2)* w + RminNew2;


%% Contrast Correction to Lower Side
vR5 = reshape (ThreshB', [numel(ThreshB) 1])';
BmaxNew = max (vR5);
BminNew = min (vR5);
MaximumofBlue = BmaxNew;
LowerLimit = 0;
MaximumB = BmaxNew;
MinimumB = BminNew;

r = (MaximumofBlue - LowerLimit)/(MaximumB - MinimumB);

Po2 = (ThreshB - BminNew)* r + 0;


%% Contrast Correction to Both Side
vR6 = reshape (ThreshG', [numel(ThreshG) 1])';
GmaxNew2 = max (vR6);
GminNew2 = min (vR6);
UpperLimit = 255;
LowerLimit = 0;
MaximumG = GmaxNew2;
MinimumG = GminNew2;
s = (UpperLimit - LowerLimit) / (MaximumG - MinimumG);

Po3 = (ThreshG - GminNew2)* s + 0;
```



```
%% Combined the value

CI = cat(3, Po,Po3,Po2);
figure(10); imshow(CI);  % Display the adjusted pixels image

% Get histValues for each channel
[yRed, x] = imhist(Po);
[yGreen, x] = imhist(Po3);
[yBlue, x] = imhist(Po2);

% Plot them together in one plot
figure(11); plot(x, yRed, 'Red', x, yGreen, 'Green', x, yBlue, 'Blue');

%% Edge detected for original image
img = imread('testimageC4.jpg');
img = rgb2gray(img);
subplot(1,3,1); figure (12); imshow(img);title('input image');
C = double(img);
for i = 1:size(C,1)-2
    for j = 1:size(C,2)-2
        x = ((2*C(i+2,j+1)+C(i+2,j)+C(i+2,j+2))-(2*C(i,j+1)+C(i,j)+C(i,j+2)));
        y = ((2*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-(2*C(i+1,j)+C(i,j)+C(i+2,j)));

        img(i,j) = sqrt(x.^2+y.^2);

    end
end
subplot(1,3,2); figure (12); imshow(img);title('Sobel gradient');
t = 85;
img = max(img,t);
img(img==round(t))=0;
numberOfEdgePixels = sum(img(:));
img = uint8(img);
```



```
img(img==round(t))=0;
numberOfEdgePixels = sum(img(:));
img = uint8(img);
subplot(1,3,3); figure(12); imshow(~img);title('Edges Detected after threshold 85');
%% Edge detetcted foe the image after color enhancement

imgafter = rgb2gray(CI);
subplot(1,3,1); figure (13); imshow(imgafter);title('input image');
C = double(imgafter);
for i = 1:size(C,1)-2
    for j = 1:size(C,2)-2
        x = ((2*C(i+2,j+1)+C(i+2,j)+C(i+2,j+2))-(2*C(i,j+1)+C(i,j)+C(i,j+2)));
        y = ((2*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-(2*C(i+1,j)+C(i,j)+C(i+2,j)));

        imgafter(i,j) = sqrt(x.^2+y.^2);

    end
end
subplot(1,3,2); figure (13); imshow(imgafter);title('Sobel gradient');
t = 85;
imgafter = max(imgafter,t);
imgafter(imgafter==round(t))=0;
numberOfEdgePixelsafter = sum(imgafter(:));
imgafter = uint8(imgafter);
subplot(1,3,3); figure(13); imshow(~imgafter);title('Edges Detected after threshold 85');
```