Real Time Implementation of Bi-Histogram Equalization Method on Android Platform

Shiva Sumanth Reddy[1], Anil kumar B[2], Manjunath D R[3]

[1,2,3]Computer Science and Engineering, DSATM, Bangalore,Karnataka

---***---

Abstract - ***Android is an open-source platform developed by Google and Open handset alliance. Image Processing on Android based mobile devices is an emerging field in today's World. Android also comes with a vast library of useful functions, including functions for user interfaces, image/bitmap manipulation, and camera controls. Histogram equalization is widely used for contrast enhancement in a variety of applications due to its simplicity and effectiveness. Examples include medical image processing and radar signal processing. One drawback of the histogram equalization can be found on the fact that the brightness of an image can be changed after the histogram equalization, which is mainly due to the flattening property of the histogram equalization. This paper presents the implementation of Brightness Preserving Histogram Equalization in Android Platform. The drawback of histogram equalization can be overcome by using the above mentioned method. This paper also shows the output on Android platform.***

Key Words**:** Histogram Equalization, Image Processing, Image Acquisition, Android, BBHE.

1. INTRODUCTION

Histogram equalization is the one of the well-known methods for enhancing the contrast of given images in accordance with the sample distribution of an image. Useful applications of the histogram equalization scheme include medical image processing and radar image processing. In general, histogram equalization flats the density distribution of the resultant image and enhances the contrast of the image as a consequence, since histogram equalization has an effect of stretching dynamic range.

The BBHE firstly decomposes an input image into two sub-images based on the mean of the input image. One of the sub-images is the set of samples less than or equal to the mean whereas the other one is the set of samples greater than the mean.

The BBHE firstly decomposes an input image into two sub-images based on the mean of the input image. One of the sub-images is the set of samples less than or equal to the mean whereas the other one is the set of samples greater than the mean. Then the BBHE equalizes the sub-images independently based on their respective histograms with the constraint that the samples.

Then the BBHE equalizes the sub-images independently based on their respective histograms. In other words, one of the sub-images is equalized over the range up to the mean and the other sub-image is equalized over the range from the mean based on the respective histograms. Thus, the resulting equalized sub-images are bounded by each other around the input mean, which has an effect of preserving mean brightness.

The architecture of the image processing in the Android mobile is shown in below.

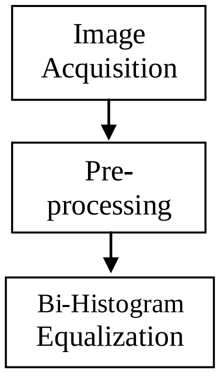

Fig -1: Enhancement Methodology

Image Acquisition refers to the capturing of image data by a particular sensor or data repository. Once the image data is acquired, Pre-Processing often includes removing of noise from the acquired image data. Histogram Equalization is the process of enhancing the image.

1.1 Motivation

Image processing on mobile phones is a new and exciting field with many challenges due to limited hardware and connectivity. Phones with cameras, powerful CPUs, and memory storage devices are becoming increasingly common. The need for benchmarking basic image processing routines such as: addition, convolution, thresholding and edge detection is important for comparison of systems. With this information developers and researchers can design complex computer vision and image processing applications while being aware of the current state of the art limitations and bottlenecks on mobile phones.

1.2 Goals

The goal of this paper is to focus on Image Acquisition, Pre-Processing through implementing image noise removal algorithms; implementing Histogram Equalization and Histogram Equalization based Enhancement method on Android based mobiles using the available Android Software Development Kit (SDK).

2. Approach and Challenges

The Android operating system is preferable for benchmarking due to its recent growth and popularity. Few of the hardware manufacturers are e.g. HTC, Motorola, LG and Samsung. The Android operating system is supported and a part of the Open Handset Alliance. This alliance positions key manufacturers, cellular providers and the Android operating system in a collaborative environment which has caused large growth since October 2008 when the first Android mobile phone was released.

Few challenges when implementing Android mobile devices with Android OS includes architecting software and optimizing code for

a. Memory limitations

b. CPU limitations

c. Image Quality limitations

3. Histogram Equalization Algorithm

Input: An Image file, Row M, Column N

Output: Enhanced Image after histogram Equalization

Steps:

1. Let X = { X (i ,j) } denote a given image composed of L discrete gray levels denoted as {XO,X1,.,. . ,X L - 1 } , where X (i , j) represents an intensity of the image at the spatial location (i , j) and X (i , j) ∈ {XO,X1,.,. . ,X L - 1 } . For a given image X, the probability density function p (X k) is defined as

$$p(x_k) = \frac{n^k}{n}$$

for k = 0 , 1 , . . . , L - 1, where nk represents the number of times that the level Xk appears in the input image X and n is the total number of samples in the input image.

2. Based on the probability density function, we define the cumulative density function as

$$C(x) = \sum_{j=0}^{k} p(x_j)$$

Where, Xk = x, for k = 0, l , . . . , L - 1. Note that C(XL- l) = 1 by definition. Histogram equalization is a scheme that maps the input image into the entire dynamic range, (Xo, XL-1), by using the cumulative density function as a transform function.

3. Define a transform function f(x) based on the cumulative density function as

$$f(x) = X_0 + (X_{L-1} - X_0)c(x)$$

4. Then the output image of the histogram equalization, Y = {Y (i,j) } , can be expressed as

$$Y = f(X)$$

$$= \{f(X(i,j)) | \in X\}$$

4. BBHE ALGORITHM

Input: An Image file, Row M, Column N

Output: Enhanced Image after histogram Equalization

Steps:

Denote by Xm the mean of the image X and assume that

$X_m \quad \{X_0, X_1, X_2, \ldots X_{L-1}\}$, ,Based on the mean, the input image is decomposed into two sub-images XL and XU as

$$X = X_L \cup X_U$$

2. The probability density functions of the sub-image XL and XU as

$$p_L(X_k) = \frac{n_L^k}{n_L}, \quad \text{where } k = 0, 1, \cdots, m,$$

and $$p_U(X_k) = \frac{n_U^k}{n_U}, \quad \text{where } k = m+1, m+2, \cdots, L-1,$$

5. Results

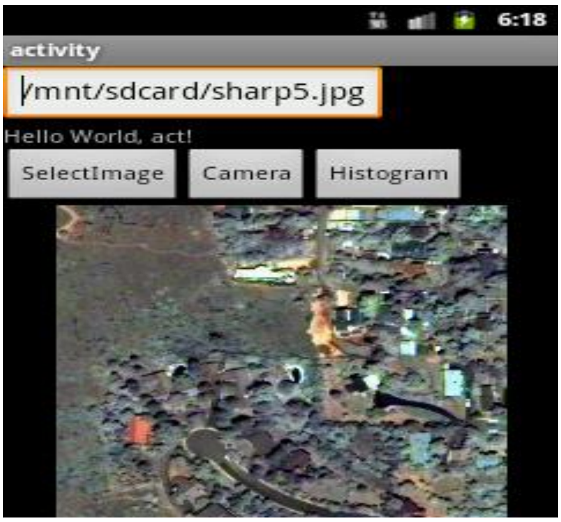

Fig -2: Input image of Forest

Fig -3: Output Image of Forest

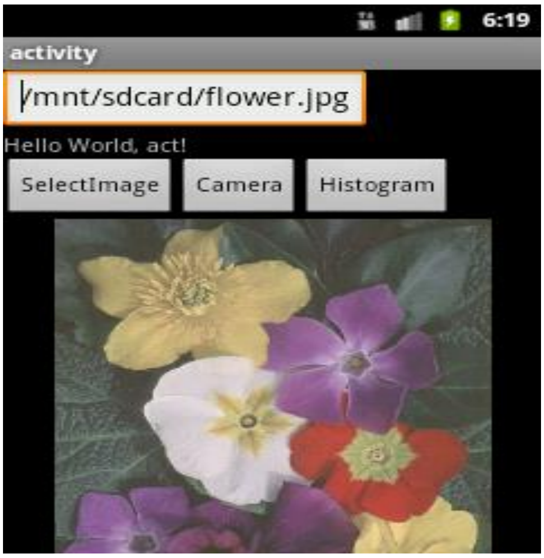

Fig -4: Input Image of Under water Fish

Fig -5: Output Image of Under water Fish

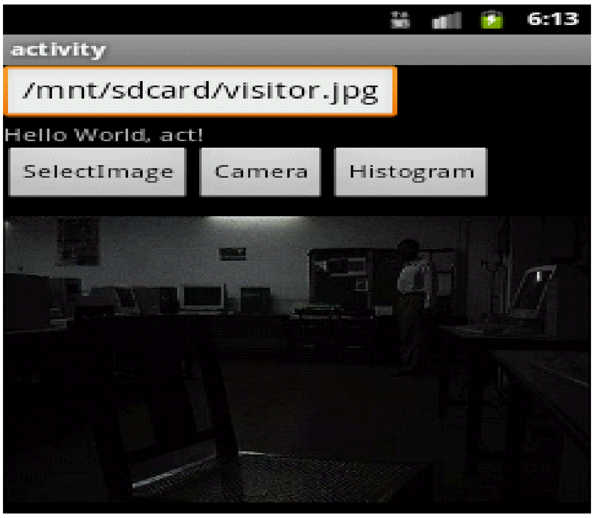

Fig -6: Input Image of Furniture

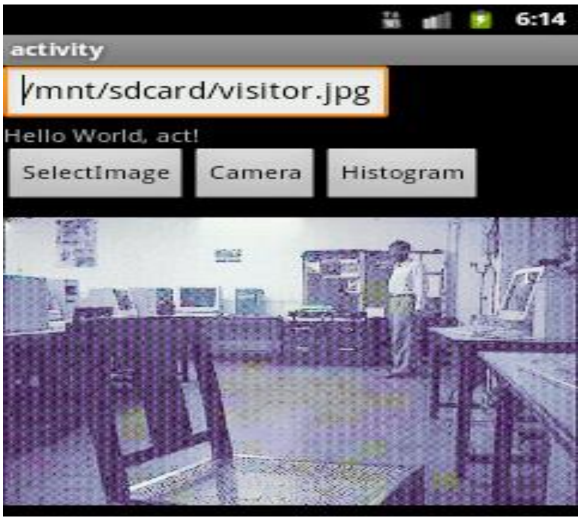

Fig -7: Output Image of Furniture

6. Conclusion

In this paper we have used two different image enhancement methods that are popularly used in image understanding studies. Our results show that the images after enhancement have better visibility than the original images. This paper shows that the processing can be applied even for the real time image that is the image captured from mobile. This paper also avoids the flattening property of Histogram Equalization by using the Bi-Histogram Equalization.

REFERENCES

[1]. J.S Lim, Two-Dimensional Signal and Image processing, Prentice Hall, Englewood Cliffs, New Jersey 1990.

[2]. R.C Gonzalez and P. Wints, Digital Image Processing, 3rd Edition, Addison-Wesley Publishing Co., Reading, Massachusetts.

[3].Y. Li, Wang, and D. Y. Yu, “Application of adaptive histogram equalization to x-ray chest image,” Proc. Of the SPIE, pp. 513-514, Vol. 2321 1994.

[4].Y.T. Kim, “Contrast Enhancement using Brightness Preserving Bi–Histogram Equalization”. IEEE Transactions on Consumer Electronics, Volume 43, No.1, 1997.

[5]. Tutorial on Using Android for Image Processing Projects, EE368 Digital Image Processing, Spring 2010, Linux Version