**CAP 5400 - Digital Image Processing**

**Project 2: Histogram modification and Color processing**

Yu Liang

Department of Computer Sc. and Engineering.

University of South Florida,

Tampa, Florida

**1. Introduction**

This assignment focuses on efficient implementation of image manipulation techniques like histogram modification in gray scale and color-processing programs using color images wrote in C++. Image histogram modification is to conduct grey level linear histogram stretching with input intensity range [a,b] and output intensity range [c,d] on specific ROIs. We can also conduct bilinear histogram stretching with input [a,b,c] and output [d,e,f]. The histograms before and after doing histogram stretching will be generated. We will implement color level histogram stretching, both in RGB and HSI (only I channel) models. The histograms before and after doing RGB histogram stretching will be generated. The histograms before and after doing HIS histogram stretching will also be generated. Color level histogram stretching on H, S, or I channel will be experimented with different channels combinations. All implementation will be applied on Region Of Interest (ROI).

**2. Description of algorithms**

**2.1. Histogram modification**

Image Histogram modification is to conduct grey level linear histogram stretching with input intensity range [a,b] and output intensity range [c,d] on specific ROIs. For each pixel, the result value will be calculated using the following formula.

Result := ((Input - InputLow) / (InputHigh - InputLow)) \* (OutputHigh - OutputLow) + OutputLow;

After the stretching, the value will fully use the 0-255 value range instead of only a small part of the value range.

Grey level linear histogram will be generated based on the previous and after image. The histogram is the count of number of pixels for each value from 0-255 in grey level. The histogram is draw on a 256 by 256 pixels image, with black and white theme.

**2.2. Color Processing**

First, we conduct color level histogram stretching in RGB. We calculate the new R, G, B value based on the stretching formula:

Result (R, G, B) := ((Input - InputLow) / (InputHigh - InputLow)) \* (OutputHigh - OutputLow) + OutputLow;

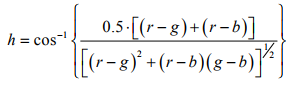
The new RGB will stretched only for the input range (a, b) to output range (c, d). Each of the R, G, B channel value will be calculated and replace the old values in the R, G, B channels.

Second, we conduct RGB to HIS conversion using the formula:

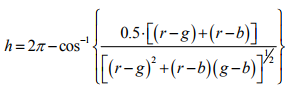
First, we convert RGB color space image to HSI space beginning with normalizing RGB values:



Each normalized H, S and I components are then obtained by,

















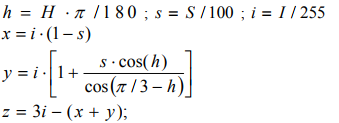
For convenience, h, s and i values are converted in the ranges of [0,360], [0,100], [0, 255], respectively

by: H h = ×180 /π ; S s = ×100 and I i = × 255 .

After that, the H, S, I channel will be stretched using the formula:

Result (H, S, I) := ((Input - InputLow) / (InputHigh - InputLow)) \* (OutputHigh - OutputLow) + OutputLow;

The new HIS value will be converted back to RGB using the following formula:



when h < 2π / 3 , b = x ; r = y and g = z.

when 2π / 3 ≤ h < 4π / 3 , h = h − 2π / 3, and r = x ; g = y and b = z.

when 4π / 3 ≤ h < 2π , h = h − 4π / 3, and g = x ; b = y and r = z.

The result r, g and b are normalized values which are in the ranges of [0,1], therefore, they should be multiplied by 255 for displaying.

**3. Description of implementation**

The entire code is developed in C++ language on an FSprime server. The code is based on the code for project 1. We will show the result of 1. Histogram Modification 2. Color Processing. Implement rectangular region of interest (ROI) selection by specifying pixel location (X,Y) of the left-top pixel of ROI and ROI size (Sx,Sy). Modify basic range function to operate only within specified ROI(s).

Add "hist" to options of gray level image processing. Let range be the a, b, c, d user defined input parameter.

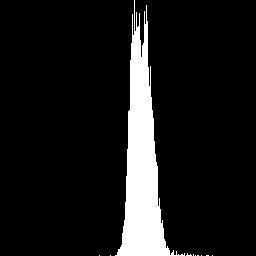
Add “conversion” color processing option to your image processing toolbox. Let range be the a, b, c, d user defined input parameter.

**4. Results**

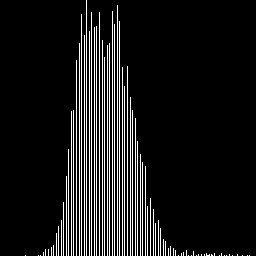
Figures 1 show the results of gray level histogram on grey scale images. We can see previous image the histogram concentrate on a small range. After histogram stretching to (0, 255), the pixel value uses more range space. The contract of picture become much clearer.



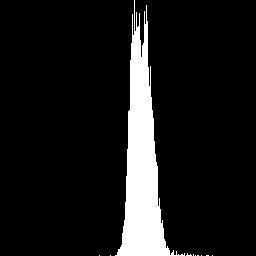
1. Original image



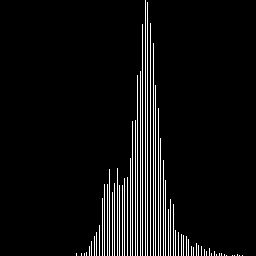
1. Histogram of original ROI 1



1. Histogram of stretched ROI 1



1. Histogram of orignial ROI 2



1. Histogram of stretched ROI 2



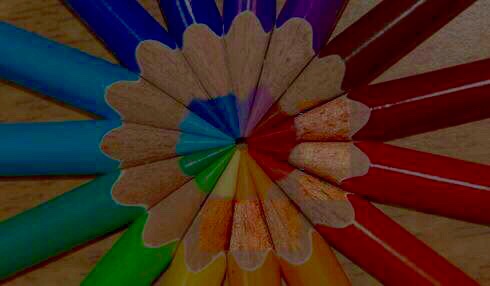
1. stretched image

**Figure 1. Original images of slope and two ROIs on Gray Scale Images. The input range is (100, 200); the output range is (0, 255).**

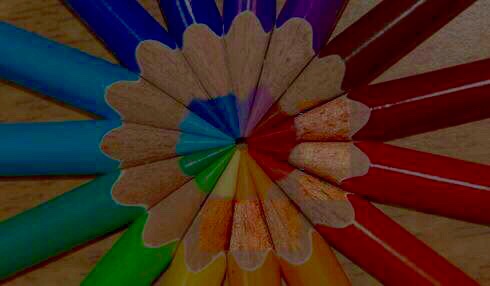
Figures 2 show the results of Color Processing on color scale images. We can see the color varied as we stretch the original RGB channels. We can also see the color varied as we stretch the converted I channels in the HSI format. Generally, the color become more vivid after stretching. The RGB and HSI stretching have similar effect.



1. original image



1. RGB stretched image



1. HSI I cannel stretched image

**Figure 2. Original images of a pens on color Images in two ROIs ( in the center right and center bottom) . The RGB stretched image is input range (20,100) to output range (0, 200). The stretched HIS ( I channel) is input range (20,100) and output (0, 200).**

**5. Discussion**

Discuss results and performance (including execution speed of two implementations) of Histogram modification: Histogram stretching can be used to increase the contract of image so that the items in the images is easier to recognize. With the help of histogram, user can manually set-up the input/output range to maximum the ability of stretching.

Bilinear histogram stretching can also be used, as to convert (a, b) to (d, e), convert (b, c) to (e, f). The bilinear histogram is useful when the image has two peaks in the histograms.

Discuss results of color processing: Histogram stretching can be used in RGB image to increase the contract of color so that the colors in the images is more vivid to viewer. With the help of histogram, user can manually set-up the input/output range to maximum the ability of stretching. After convert RGB to HSI, we found that the image improve similarly while stretching the I – intensity channel. This make the stretching easier because you don’t have to stretch all three channels. Stretching H, and S channel will make different effects but not as good as the I channel.

**6. Conclusions**

This assignment shows the basic implementation of the histogram modification and color processing. It highlighted the good performance of histogram stretching to help the image more clear and vivid. Also HSI conversion has the advantage to allow just one channel stretching.

Yu Liang

Project 2

Histogram Modification and Color Processing

\*\*\*\*\*

This software is architectured as follows. This software can work on grad server.

iptools -This folder hosts the files that are compiled into a static library.

image - This folder hosts the files that define an image.

utility- this folder hosts the files that students store their implemented algorithms.

lib- This folder hosts the static libraries associated with this software.

project- This folder hosts the files that will be compiled into executables.

bin- This folder hosts the binary executables created in the project directory.

\*\*\* INSTALATION \*\*\*

On Linux

Enter the project directory in terminal and run make

As a result you should get iptool in project/bin directory.

\*\*\* FUNCTIONS \*\*\*

1. Histogram Modification: hist

show the before and after histogram of ROIs

linear histogram stretching with input range (a,b) to (c,d)

2. Color Processing: conversion

histgram stretching in RGB.

Covert RGB to HSI and histgram stretching I channel and covert back to RGB

ROI\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Hist:

100 300 100 100 5 100 200 0 255 # Parameters for ROI1: x, y, sx, sy, T/WS, a, b, c,d

200 200 50 100 4 100 200 0 255 # Parameters for ROI1: x, y, sx, sy, T/WS, a, b, c,d

#need a space after 255#

Conversion:

50 50 50 50 5 20 100 0 200 # Parameters for ROI1: x, y, sx, sy, T/WS, a, b, c,d

100 100 50 50 4 20 100 0 200 # Parameters for ROI1: x, y, sx, sy, T/WS, a, b, c,d

#need a space after 200#

\*\*\* PARAMETERS FILE \*\*\*

There are for parameters:

1. the input file name;

2. the output file name;

3. the name of the filter. Use "hist" or "conversion" for your filters;

4. the value for adding intensity, threshold value for binarize filter, or the scaling factor for scale filter Or input, output range.

\*\*\* Run the program: ./iptool parameters.txt

parameters.txt Example:

slope.pgm slope\_histogram.pgm hist 50

pen.ppm pen\_conversion.ppm conversion 50