



CS – 4802
Digital Image Processing
Lab #2

Histogram Equalization

Submitted by: Jiri Sumbera
Submitted to: Dr. Taylor
Submitted on: 02-06-01

Introduction

The number of different light intensities in an image often does not use the whole available spectrum and mostly accentuate a narrow spectrum. Images with such poor intensity distributions can be helped with a process known as histogram equalization, which in essence redistributes intensity distributions. The main goal of the second lab was to implement the histogram equalization and test it on few sample images.

Problem Statement

Within application developed in the previous lab:

- add the ability to perform histogram equalization for both grayscale and color images
- add the ability to perform the above-mentioned task for selected regions within the image

Secondary Goals:

- add the ability to select region within an image
- add the ability to convert the RGB color image into the HSI color space and vice versa
- add the ability to convert color images to grayscale
- add the ability to crop an image using the current selection

Procedure

Due to the fact that my application uses external viewer for displaying images, I had to incorporate a method of selecting a region within the image using a dialog box.

Design Steps:

- 1, First, a simple CDialog-based class 'Region' was designed to allow for user input of the selection in the format: first column, last column, first row, last row.
- 2, Next, a powerful 'Copy' function was coded allowing for the copying of a rectangle from the source image into a rectangle of the same size within the target image.
- 3, In the following steps message handling functions of the 'CNothingDoc' class for the various tasks to be performed were created. The original intention was that these functions would only call similarly named functions of the 'My' image class, which would then perform all necessary actions. However due to programming difficulties this was not strictly held to.
For example:
The approach used within the application for localized histogram equalization checks whether the user has made a selection, then creates a temporary image with the size of the selection, performs the equalization (this finally happens within the object) and then copies the selected region back to the original image. The better approach would be simply to call the objects 'Equalize' function and let it check whether a selection is active and then do the localized equalization as this would save more memory, be faster and be more OOP-like.
- 4, Next, the Equalize function of the 'My' class was coded, the grayscale equalization being based on the source code found in the book 'A Simplified Approach to Image Processing (see Resources). For histogram equalization of color images, two color conversion functions were developed - Convert2RGB and Convert2HSI - and the Equalize function was enhanced to deal with the Intensity channel.
- 5, As a by-product of the above the implementations a 'CNothingDoc' function class 'Crop' was easy to implement as well as the color-stripping function 'Convert2I' (based on 'Convert2HSI')
- 6, Method of displaying the histogram of both the whole image and its selection was developed. The corresponding 'Histogram' function is largely based on the first part of the 'Equalize' function.

Discussion

Below are the screen captures of the application before and after **global** histogram equalization was performed on the two sample images - *haleBopp.jpg* and *room.png*, which were of course converted first to the supported PPM and PGM file formats. The appropriate histograms are included as well.

room before



histogram (please note that it is rotated clockwise by 90 – as the application creates it)



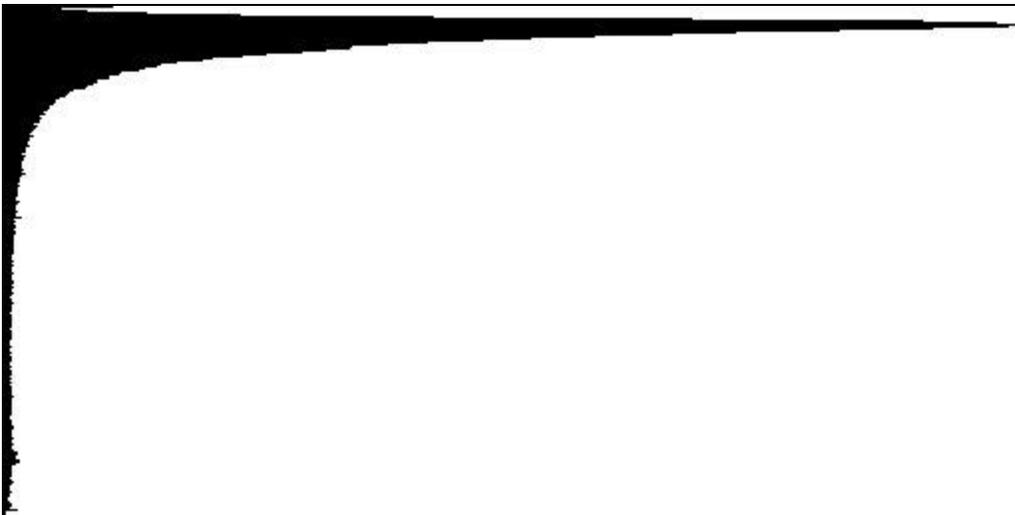
It can be easily seen that the image is predominantly black with some very white areas.

room after



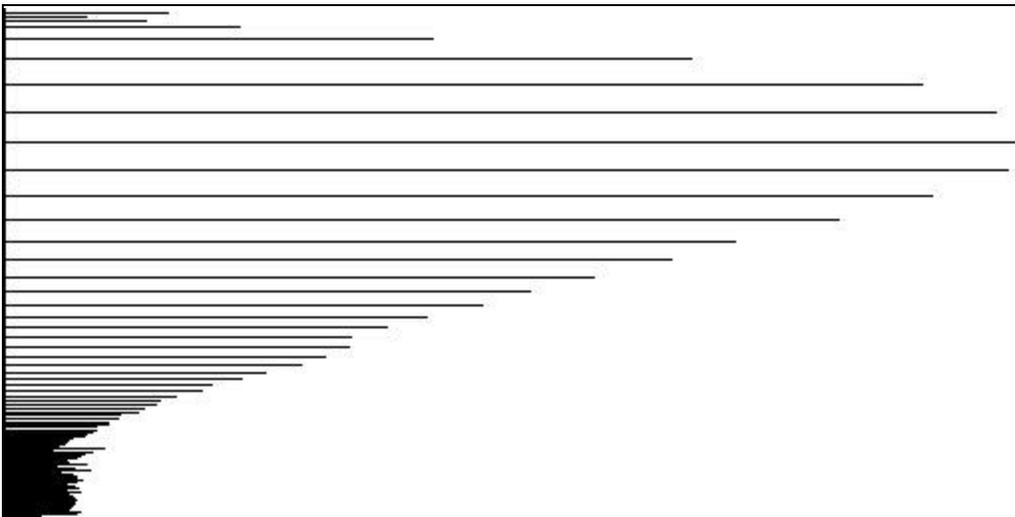
It can be seen that all the content from the black part of the spectrum was shifted towards the center, thus brightening the image and revealing all the detail in originally black parts. It should be noted that in this case pure histogram stretching would surely lead to great information loss at the originally white areas.

haleBopp before



The image of the comet clearly contains mostly the dark parts of the spectrum, with relatively smooth transition (on the histogram) to lighter areas.

haleBopp after



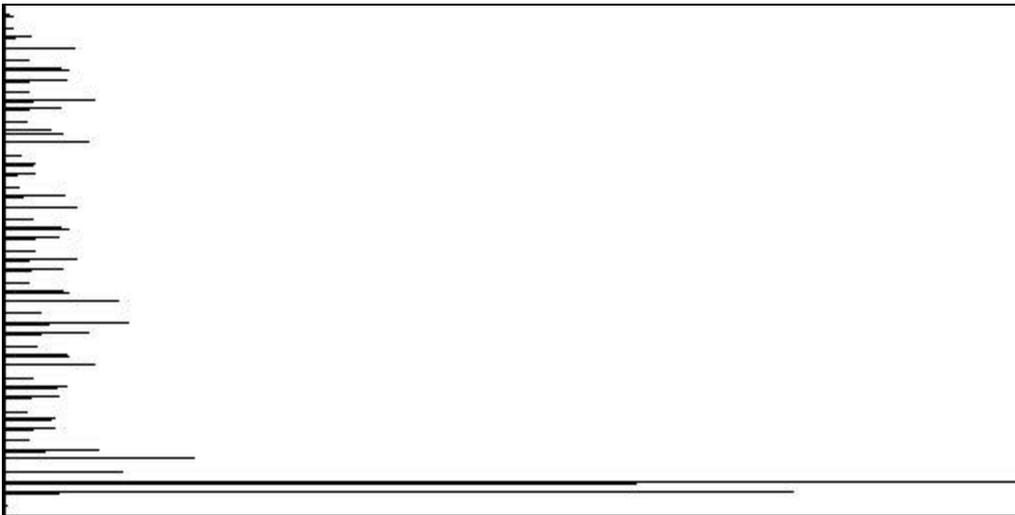
Histogram equalization spreads out the dark parts of the spectrum. This however reveals a lot of noise previously hidden in the image (only partly visible in the lighter areas of the original image). This noise highly resembles vector quantization performed by e.g. JPEG compression.

Below are the screen captures of the application before and after **local** histogram equalization was performed on the two sample images - *haleBopp.jpg* and *room.png*, which were of course converted first to the supported PPM and PGM file formats. The appropriate histograms are included as well.

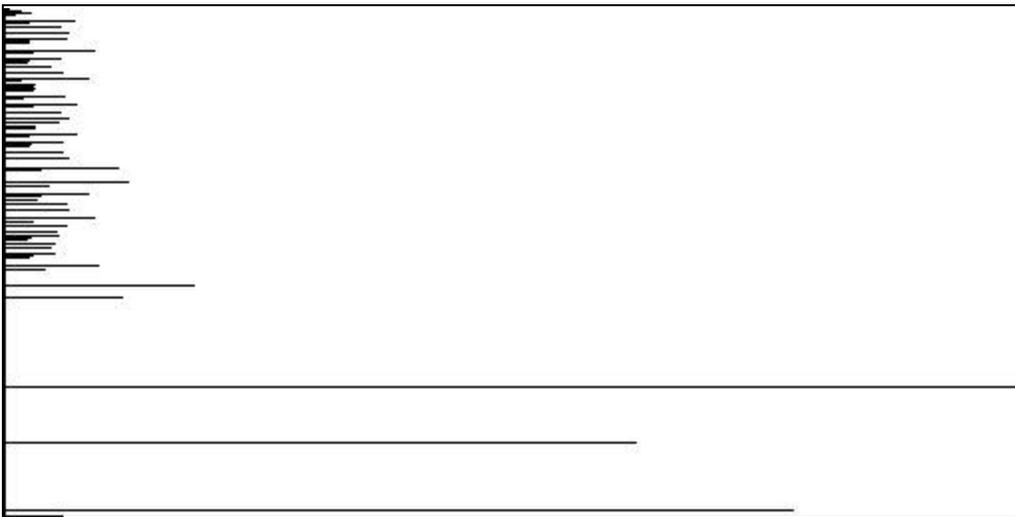
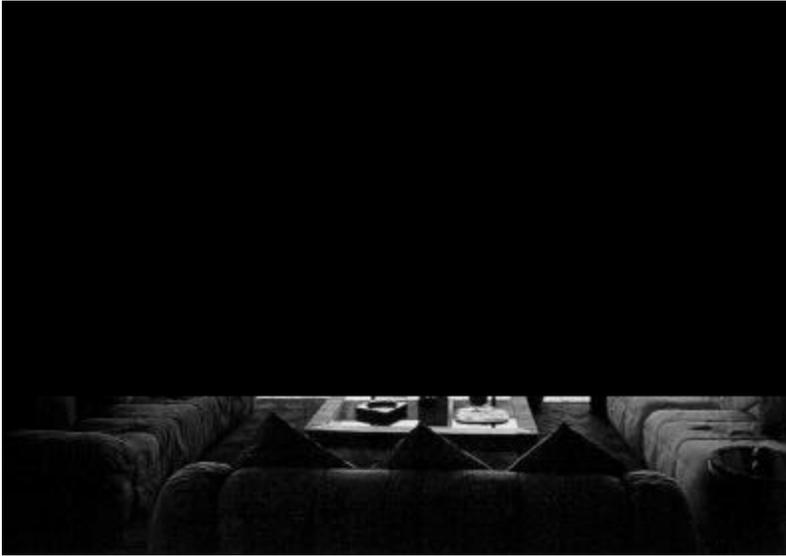
For the picture of the room, the too-white window was chose as the area to be equalized.

room before

area selected = rectangle ([80,80]; [360,160])



room after



As the histogram shows, the previously almost uniform (with the exception of very white parts) spectrum was shifted towards the darker part, revealing a lot of information previously invisible.

For the picture of the room, the left part of the picture was chose as the area to be equalized, to see if information about the stars with lesser brightness could be extracted.

haleBopp before

area selected = rectangle ([1,1]; [210,375])



The selected region is dominantly black, with a small cluster of gray levels – the stars.

haleBopp after



As with the global haleBopp equalization, the quality of the stars is not improved, only the quantization can be more easily seen. The only information revealed is about the position of the comet's tail.

Resources used:

Irfan Viewer: <http://stud4.tuwien.ac.at/~e9227474/>

Irfan Viewer is a free image viewer supporting various image formats

Copyright Irfan Skiljan 1996-2000

A Simplified Approach to Image Processing:

Classical and Modern Techniques in C

by Randy Crane (Hewlett-Packard Company)

Published by Prentice Hall PTR

Upper Saddle River, New Jersey 07458

Copyright: Hewlett-Packard Company 1997

Activity Log

The following are very gross estimates:

Design: 1hr

Coding: 3 hrs

Debug: 2 hrs

Test: 1hr

Documentation: 2 hrs

Other - research: 1hrs

Conclusions

This lab had shown the strength of histogram equalization especially in its ability to locally enhance images. In addition other useful functions such as conversion to grayscale or selection setting were accomplished.