Report on
Image Processing and Interpretation
Project 1

Submitted to
Dr. Bebis
for
CS474

September 21, 2015

By:
Aaron Whitehead
and
Jeremiah Berns

This report examines several introductory computer processing practice programs. The programs cover such things as spacial reductions, color reductions, and histogram equalizations. The project was a great success, and each part of the project runs smoothly and correctly.

Technical Discussion

As the overall structure of the project is broken up into 3 sections, the technical writeup will also be broken into three subsections.

Program one:

The first program is a image sampling practice set. The idea behind image sampling is to only take every x pixels or so, so that the overall image is similar, but has been scaled down to a lower resolution. The basic algorithm goes as thus:

1. Create a new image with the correct end dimensions.
2. Iterate across the new image one pixel at a time, while iterating across the original picture by only grabbing every x pixels, starting with the first pixel.
3. As you iterate through the original image and get to the end of a row, skip x number of rows also, eliminating both rows and columns to evenly scale the picture down.
4. At each pixel from the loop, copy the current value from the original picture and put into the new picture.

After you have followed all of these basic steps, you will see that you now have a picture that is scaled down by a certain factor, we use 2, 4, and 8 in our trials later on in the report. However, once the images were scaled down, the problem also asked to have the project scale back up to the original size so one can see the data that was lost in the transition. The algorithm for growth is as follows:

1. Create a new image with 256x256 resolution.
2. Iterate over every pixel of the shrunken image, and at each pixel copy the value to the new picture, and simply repeat each pixel by the amount of the scaling factor. IE, for a picture scaled down by 4, repeat each value four times.
3. Once your iteration has reached the end of a row, copy that row another n-1 times, where n represents the scaling factor again.

Program two:

The second program is a image Quantization problem. Image quantization means to simplify the data in the image by reducing the total number of colors allowed. This can be useful for making or using images in a system with little memory available. The algorithm that was used for this system goes as thus:

1. Generate a value called divisor by dividing the total number of

colors (256) by the amount of colors desired in the end result (any value from 0-256).
2. Iterate over each single cell of the picture.
3. At each cell, divide the value of the cell by the divisor from step 1 This will, by utilizing how integer math works, reduce each color from a 0 to 255 range down to a 0 to the desired number of colors.
4. Finally, multiply each cell now by the value of the divisor. This will simply make sure each color numbers now in the image are evenly spaced from 0-255. This final step will allow the image to utilize a wider range of colors instead of only using 0-desired number of colors, which would be difficult to see from the naked eye, as each color difference would only be 1 single value.

Program three:

The third and final program was a simple Histogram Equalization, which essentially counts the use of each color, and then tries and evenly distribute the colors slightly in order to get a wider range of color usage. The algorithm goes thusly:

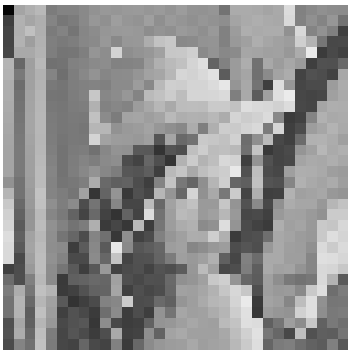
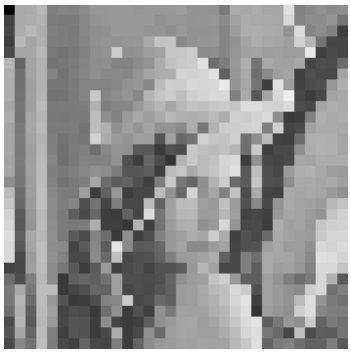
1. Iterate through the image and count how many times each color value appears, store these numbers into an array.
2. Add up the cumulative probability of each color in order to discern how much of the picture each color takes up.
3. Multiply each step of the cumulative probability array by 255.
4. Store the cumulative probability which was multiplied into an adjusted histogram array.
5. Print the adjusted histogram to file with the write image function.

Results and Discussion

Once again, the results section, like the technical discussion section, will be broken into three core parts.

Problem one:

The results for problem one were very promising, and the visual comparison of the pictures seems to be exactly as desired. Below, you will see three figures. The first one is the original image. The second image is the first picture scaled down by a factor of 8, and then finally the third image is the scaled down image scaled back up, showing the information loss of such an operation.

The first image. The second image. And the third image.

The images have been resized to fit correctly. As you may see by reviewing the second and third image, the growth function seems to work perfectly.

Problem two:

Reducing the number of colors in a picture had an unexpected visual result, but a correct one still. Below you will see three images. The first image is the given picture. The second image is the same picture reduced from 256 colors to 8 colors. The third and final picture is the same picture with only two colors present.



First image. Second image, 8 colors Third image, 2 colors.

As you can see by comparing the pictures, the biggest affect on a gray scale image of removing some colors is simply a less smooth gradient when going from light to dark. This drastically enhances the visibility of shadows, and slight gradations of the peppers in the picture. The final picture, which only has two colors, seems like it would be very useful in finding edges of objects, as the slight difference from the colors of the peppers in the original images are now stark and boolean. We as a team were not sure whether or not to leave the colors as 0 and 1 or 0 and 128, so we decided to use 0 and 128 in order to have the picture be visible to the human eye.

Problem three:

Finally, we come to the third problem, which while having a more subtle effect, has a very pleasing result. Below you will see two pictures, the first being the original, and the second being the equalized version of said picture.

Original Equalized

At first glance the equalized image seemed much better overall then the original. The details pop vs the white fogginess of the original. However, one downside is that the details in the forefront, such as the symbols on the plane have become slightly discolored and the edges are not as sharp. So while the equalized image may be slightly more pleasing to look at since your eye must strain less to see the subtle details, it is a downgrade overall due to the fact that we seem to be losing some data.

Division of Labor

It is slightly difficult to say exactly who did what, it can be said that both Aaron and Jeremiah worked equally on the write up, and also on problem three, while Aaron took a lead on problem two while Jeremiah took the lead on problem one. Using GitHub and voice chat while working means that the labor was very evenly split.