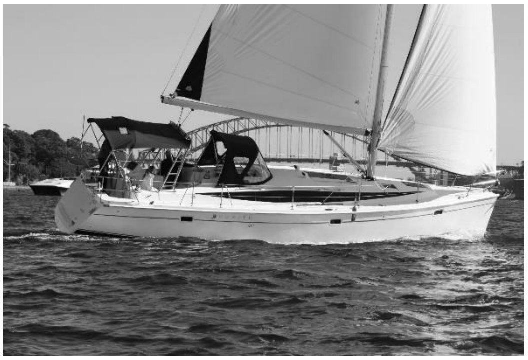
SJSU - CS 174 – Server-side Web Development – Project 1 – PHP GrayScale Conversion

Introduction



In this assignment you will create a PHP function to convert any jpg image to GrayScale (Black and White) using three (3) different color-to-grayscale conversion functions. You will test your function on images of different sizes and compare the performance of the function for each case. This exercise will use all the functionality you have learned from the first five class lectures including: functions, expressions, operators, precedence, type conversion, looping constructs, statements and the GD (PHP graphics) library.

Background

Color-to-Greyscale conversion of any image at the simplest level is straightforward. It involves iterating over each pixel value in an image, applying a conversion function to the (Red, Green and Blue) components of the pixel to compute a single grayscale value from the RGB values. The new grayscale value is put back in the image, replacing the original RGB components with the single greyscale value for each Red, Green and Blue.

There are several methods used for the function, each of which produce acceptable, but slightly different results. Three of them are:

1. *Averaging* – simply compute the average of the red, green and blue values (used to convert sailboat image above):

   Grey = (r + g + b) /3

2. *Lightness* - average just the “brightest” and “darkest” colors:

   Grey = (max(r, g, b) + min(r, g, b)) /2

3. *Luminous* – applies “weighted” values to each r, g and b component based on experiment results on human perception of colors.

   Grey = w1 * r + w2 * g + w3 * b; where w1 + w2 + w3 = 1;

   Humans perceive green more strongly than other colors, so this weight for the green value is higher than the other values. (For example: try w1 = .3 w2 = .6 w3 = .1). You can experiment with these weights to determine which value you prefer. It is really a creative choice.

Required Steps

- Create a new function (from scratch):

  int convertImageToGrayScale($fileNameIn , $fileNameOut, $method)

  The string $fileNameIn specifies the source image file.

  The string $fileNameOut specifies the destination image file created by the function. The source image file is not modified.

  The int $method specifies the greyscale conversion method. Your function should support all three (3) methods specified above.

  The function should return True if the conversion is successful otherwise false; You can optionally return the number of pixels converted (image width * height) if you want to return a more meaningful value instead of just True;

- Test your function for using various images and conversion methods. What are the differences in the quality of the greyscale conversion for each method ? Which one do you prefer ?

- Research the various time() functions supported in PHP. Log the time (in milliseconds) required to convert an image. Compare the time it takes for three different image sizes. How does the performance change with different size images ?

A few pointers you might find helpful:

- Look at the GD functions, *imagecolorat(), imagesetpixel() and imagecolorallocate();*
- The function *imagejpeg()* can write files.
- Develop your function is small parts, testing each part incrementally as you go. You might want to try just simply copying all the pixels in the source image to the destination image first just to get your work started.

Important - What to Submit

1. Submit your PHP source and zip it into a .zip file using the following naming convention:

   Project1-PHPImages-<your name>-<date>.zip.

   As an example:

   Project1-PHPImages-KevinSmith-09042018.zip

   Note: No other compressed format will be accepted (zip is available natively on both windows and Mac).

2. Submit a short document (PDF) with your sample images describing your results.

Due Date

This assignment due at the end of next week. Due date will be posted in Canvas system.

Grading Criteria

To receive the highest grade on the assignment, the code most be robust and well documented (commented) with all functionality specified included. Source code files must be signed by the author.   See Rubric for breakdown.