Programming Assignment #4
CS253, Fall 2016
Due: Thursday, Oct. 6th

"*Matching Images*"

Motivation

In PA3, you compared depth images using normalized histograms and direct image comparison. In PA4 you will extend PA3 to compare a whole set of images, reporting for every image what its closest match is. The goal is to get you to manipulate dynamic sets of images, and to get you to start thinking about efficiency (although you are not yet being graded on efficiency). Also, I want you to start asking yourself: which comparison method is better?

Task

Your program will take a single file name as an argument. Instead of an image, the input file is a text file containing the names of image files, one filename per line. Your program will read every image listed in the input file. If there are N input images, it will output N lines of text to std::cout. Each line of output corresponds to one of the input images. The line of output contains (1) the input image filename, followed by whitespace, followed by (2) the most similar image filename as determined by the normalized histogram measure, followed by whitespace, followed by (3) the normalized histogram error measure between the two images. The lines should be in order, meaning that the first line corresponds to the first input image, the second line corresponds to the second input image, etc.

Since we are using the normalized histogram comparison measure, the input images do not need to be the same size. However, every string in the input file should be the name of a legal image, otherwise there is an error. Also, there need to be at least two filenames in the input file. In the case of error, you should print a message to std::cerr and return -1, as usual.

Submitting Your Work

You must submit a tar file that includes not only your .cpp and .h file, but also a makefile. The makefile MUST CREATE AN EXECUTABLE CALLED PA4. Submit the tar file as PA4 on Canvas.

Grading Your Homework

To grade your assignment, the GTAs will unpack your archive in an empty directory and compile it using "make". If your program does not compile using make, you may be awarded no points for the assignment. Assuming your program compiles, the remaining points will be determined by how your program performs on test cases, including test cases with errors in the input files. When the input files do not contain errors, you will receive full credit for a test if the value printed to std::cout is correct. If an input file does contain errors, you will receive full credit if your program

produces a meaningful error message to std::cerr and returns -1. Note that unless there is an error, the ONLY thing you output to std::cout or std::cerr should be the computed value.

Hints

- People are still failing to test their tar files. After you make your tar file, create an empty directory, copy your tar file to it, untar it, type make, and then test your code.
- Be careful with your Makefile, too.

Policies

All work you submit must be your own. You may not submit code written by a peer, a former student, or anyone else. You may not copy or buy code from the web. The department academic integrity policies apply.

You may not submit your program late. To receive credit, it must be submitted on Tuesday, Sept. 20th. There is no late period. The exception is an unforeseeable emergency, for example a medical crisis or a death in the immediate family. If an unforeseeable emergency arises, talk to the instructor.