garb34.pgm - <http://www.cs.ucf.edu/courses/cap4453/inputpics>

Sobel - <http://www.cs.ucf.edu/courses/cap4453/progs>

**PGM format -** <http://netpbm.sourceforge.net/doc/pgm.html>

The image file starts with header information. Header information contains the image file type, in our case, “P5”, image size (width and height), and the maximum intensity value, in our case, 255. For us, the width and height for the chess and lady’s face pictures are 256 and 256.

Following the header are the details of the image pixels. Essentially, an image is nothing but a two dimensional array where each cell holds an intensity (sometimes called color or gray) value of that particular pixel in the image. For this format, the intensity value is an integer between 0 and 255, inclusive. A value of 0 indicates black, a value of 255 indicates white, and every number in between is a tone of gray.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8 | 9 | 9 | 9 | 9 | . | . |
| 8 | 9 | 9 | 9 | 9 | . | . |
| 9 | 8 | 8 | 8 | 9 | . | . |
| 9 | 9 | 9 | 9 | 9 | . | . |
| 8 | 9 | 8 | 9 | 9 | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |

Intensity values of the chess1.pgm image

So, a .pgm file is simply a header followed by a list of ints (stored as chars), such as,

89999….89999….98889….99999….89899…., etc. Note that each pixel is stored as one byte (stored in the file as a single character, i.e., char).

**Running sobel.c**

Run the program sobel.c in your C environment (such as devC++, or codeblocks).

An output file will be produced, but it will not be viewable yet by a viewer, because this output file lacks the header lines at the top. (Before you proceed, check that the output file size is 65536 (why is it this size??))

So, you need to put these three lines (below) in the code before any pixels are written to the file (but make sure you modify the printf’s to fprintf’s, and put in the file pointer). (and set rows and cols to be 256 or whatever is appropriate for your picture).

printf("P5\n");

printf("%d %d\n", rows, cols);

printf("255\n");

Now, re-run the sobel.c program, the output file should be about 65551 in size (why is it this size??).

This output file is now ready to be viewed in your viewer (to view any picture .pgm file, simply drag and drop it into the viewer.)

**Having problems??**

If you are having difficulty understanding how the input/output works with sobel.c, maybe take a step back and read diffpic.c and then read diffpic3.c and understand why changes had to be made (both programs are at http://www.cs.ucf.edu/courses/cap4453/progs).

**Instructions**

Before you do this assignment, make sure that you have read and thoroughly understood the material in the file irfanviewAndPGM.docx. It will be very difficult to make sense of what to do, if you have missed this step.

In this assignment, you will download the Sobel.c code from the website (<http://www.cs.ucf.edu/courses/cap4453/progs>), and you will first get it to run on your computer. You must run it using the face05.pgm image that is at <http://www.cs.ucf.edu/courses/cap4453/inputpics>. Then you must add code to make it use thresholds to decide which locations are edges. Use a Hi Threshold and a Lo Threshold. Put both thresholds in one program and have that program produce the two new outputs. Make sure your outputs are ready for display with a .pgm viewer. So, totally you will have 3 outputs: 1) the magnitude image 2) the output due to Hi Threshold 3) the output due to Lo Threshold.

(A .pgm viewer lets you view files with extension .pgm. While one can get a viewer for a cost, eg., Photoshop, one can instead get freeware, such as Irfanview for Windows -- Google Irfanview and download and install it on Windows platforms. If your viewer is working you should be able to view the chess image that is stored at <http://www.cs.ucf.edu/courses/cap4453/inputpics/>, then click on garb34.pgm).

To get your pictures to be viewable by a .pgm viewer, you need to add the .pgm header at the top of each output file. This is done by Adding the three lines (lines 26, 27, and 28) from the program that is stored at <http://www.cs.ucf.edu/courses/cap4453/progs/rawtopgm.c>

You must add these three lines to your sobel code, and make sure that when you add these lines, make sure that you convert these lines to make them write to files (and not to the output screen, as they are doing in the current rawtopgm.c code). Submit the finished, compile-able, and execute-able code as a zipped file to the web course.

Grading will be done by grader using Conferences tool. We will help you to schedule this when you have finished the Assignment for Module 1. Finished means you have written the program, and you have tested it, and are ready to discuss it and show that it runs correctly.

Make sure that finally, you run your program on the chess image which is titled garb34.pgm.

Your three output pictures should look like the three pictures that are stored at 1) sobelmag.pgm 2) sobelout1.pgm and 3) sobelout2.pgm. All three are stored at <http://www.cs.ucf.edu/courses/cap4453/outputpics>. Please examine them, and ensure that your three outputs look like them.

Hi All,

Since there have been several questions regarding the submission and grading of Assignment 1, I have decided to make an announcement to make things clearer.

When submitting the sobel.c code include a README.txt file which shows how to run the code in the command line. The low and high thresholds should be command line arguments. Ensure your code is readable and commented (to ease in grading).

When grading the assignment, I will run the sobel program on the face05.pgm image as well as other images on several thresholds. Your output images do not need to exactly match the output given at http://www.cs.ucf.edu/courses/cap4453/outputpics (just try different thresholds to get similar enough outputs).

There has also been a common issue where output images are cut off on the right (and wrap around on the left). This is caused by incorrectly reading and parsing the .pgm image (look at pgmtoraw.c to fix this issue). For this assignment, there will not be points marked off for this issue.

Best,

Kevin Duarte

Hi All,

To help you have a warm start with Assignment 1, we have provided

- <http://www.cs.ucf.edu/courses/cap4453/progs/rawtopgm.c>, from which you can learn about the basic operations about images and how to save the image in the .pgm format. Besides, you should also infer how to parse the .pgm format and read the images of .pgm format.

- <http://www.cs.ucf.edu/courses/cap4453/progs/sobel.c>, from which you can learn how to program the convolution.

Given the feedback thus far, it looks like some of you relied on sobel.c too much and ignored the first piece of code. In particular, some of you encoutered problems because you failed to parse the .pgm format correctly in your code. (Note that the parsing part is not in sobel.c).

After discussing with Kevin Duarte, our TA, we decide to provide another tip so that you know how to correctly parse the .pgm images in your sobel.c. Here is the code: http://www.cs.ucf.edu/courses/cap4453/progs/pgmtoraw.c

After all, this is a class on Robot Vision instead of programming. However, you are encouraged to understand and use the code as opposed to copying & pasting the code.

Best,

Boqing