

Explanation of how to download Irfanview, how to run it, and what the .pgm image format is

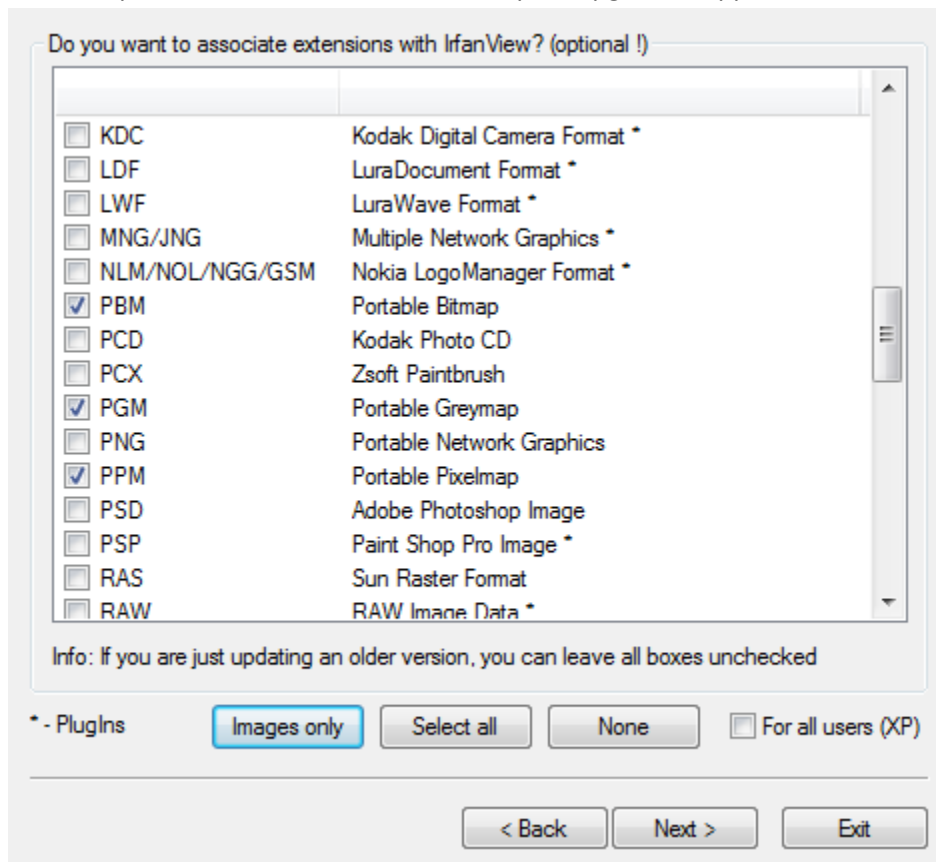
The files you will be working with represent pictures. In order to view the pictures you will work with, you need to install a free program called Irfanview.

- a) Using your web browser of choice, navigate to <http://www.irfanview.com> or to <http://www.tucows.com/>
- b) If using tucows.com, in the search field, enter Irfanview; click the Go button.

A search bar with the text "Irfanview" entered. To the right of the search bar is a dropdown menu labeled "Windows" and a blue button labeled "Go!!".

Then click on download. If on www.irfanview.com, just click on download, and select a download site, such as PC-intern.com or tucows.com

- c) Run the downloaded file (iview410_setup.exe or iview438_setup.exe); follow all instructions.
 - When the installation wizard asks what file extensions you would like Irfanview to process, make sure to select the .pbm, .pgm, and .ppm extensions.



d) In order to test your installation, download the chess picture stored in garb34.pgm from the website <http://www.cs.ucf.edu/courses/cap4453/inputpics>, and open it. A chess picture should appear.

Then download the program sobel.c from the webpage at <http://www.cs.ucf.edu/courses/cap4453/progs>

Be sure to save the program sobel.c and the garb34.pgm files all in the same directory.

Ensure that you understand the command-line input approach being used in sobel.c, or if you wish, re-write the way that the input file names are obtained by the program (eg., by simply scanf on a string variable).

PGM format

To start out, we first need to understand how the .pgm image format works and how it is stored in a file. (For a more detailed description of the .pgm file format, you can look at <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).

```
fprintf("P5\n");  
fprintf("%d %d\n", rows, cols);  
fprintf("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>).