# Image Processing: how to turn a RAW image back to PGM

(thing) by **Footprints** Tue Aug 28 2001 at 18:52:28

If you've already made a .raw image with Image Processing: how to make a RAW image or by any other way, and played around with it, you may want to view it, which is impossible in any image viewer I've heard of. This is because the raw format doesn't specify how many pixels there are on each line, so the image viewers have no idea where lines start or end. Also, for example, the image viewers have no idea whether the picture is in colour or in black and white, whether it is encoded etc. So we have to add a header.

The following program is not state of the art, but it's simple, and easy to modify. For simplicity, it doesn't even check for errors, but if you really want, copy the relevant code segments here.

The pgm header looks like this:

**P5** - *this tells the image viewer it's a pgm file*
**#Comment** - *Which can come anywhere, but I put it here*
**columns rows** - *number of columns and rows*
**255** - *this can actually be any number. It tells the image viewer the max value of a pixel in the image. Leave it at 255, which is greyscale (or regular RGB). I can honestly not think of a situation where you'll need a pgm/ppm file with more. If the maximum pixel is less than 255, leave it at 255.*

For ppm, change the P5 to **P6**.

There are breaks at the end of each line. When converting to raw, my program (here) prints out the number of rows and columns. You'll have to enter their values in the following program.

Wherever **512** is written (linked and bold), change it to the number of columns, followed by the number of rows. (I've used 512*512 here as it is the standard size).

```
/****************************
 *
 * raw2pgm.cpp - makes a .pgm picture out of the .raw picture
 *
 ****************************/


#include <stdio.h>

void main()
{
        // open raw file and pgm file
        FILE *fin = fopen("FILE_NAME.raw", "rb");
        FILE *fout = fopen("FILE_NAME.pgm", "wb");

        unsigned char buffer[512*512];

        // write header
        fprintf(fout, "P5\n#Created by Footprints\n512 512\n 255\n");

        // copy file
        fread(buffer, 1, sizeof(buffer), fin);
        fwrite(buffer, 1, sizeof(buffer), fout);

        // close files
        fclose(fin);
        fclose(fout);
}
```