

# Image Processing: how to make a RAW image

(thing) by **Footprints**

Tue Aug 28 2001 at 18:22:15

To mess around with an image, you need to have it in RAW format. Of course, I am assuming you want to process an image yourself, not using Photoshop or some other tool that is wonderful and yet limited.

To briefly recap, the .raw extension means that there is nothing but image information in the file (and not even that). It is the succession of bytes that defines a picture. In b/w greyscale, it means that each consecutive byte represents a grey scale of a consecutive pixel. This is the most comfortable format for working with images. No need for reading headers and decoding, which makes modifying jpegs a bitch.

There are 3 simple steps to making a raw file:

1. Download Irfanview. (for example from Tucows). It's freeware, and can read most formats. You'll need it.
2. Open any picture you want in Irfanview (be it jpg, gif, bmp, whatever). Now convert it to pgm, if it's black and white, or ppm if it's in colour. I'm going to discuss pgm pics, because they're simpler. ppm is the same basically, only 3 bytes for every pixel.
3. Run the following C/C++ program on the pgm file, and you'll have a .raw file ready for manipulation.

Notes:

- I'm no great C/C++ wizard - so if you ARE one, this code is going to suck. But it works. If someone should wish to write a more elegant code, please /msg me and I'll change it.
- You have to open the files in binary (rb & wb);
- Due to my terrrrific inefficiency, you'll simply have to change the code to fit bigger or smaller pictures. I've linked the code where you have to do the changes. I've read 4 bytes, assuming that the pic is over a hundred pixels wide and high. If it's less than 100, you'll have to read 3 pixels.
- Naturally, you'll have to change the filenames too. I've linked them for ease of finding them.

```

/*****
 *
 * pgm2raw.cpp - removes the header from a pgm file
 *
 *****/

#include <stdio.h>
#include <iostream.h>
#include <stdlib.h>

bool removeHeader(FILE *fin, FILE *fraw){
    char s='a';
    char j1[3];
    int number_of_columns, number_of_lines;
    int pic_size=0;

    // read intro
    for (int i=0;i<3;i++){
        do{
            fread(&s, 1, 1, fin);
        }while (s!='\n');

        // read number of columns and rows
        if (i==1){
            fread(j1, 1, 4, fin);
            number_of_columns = (int)atof(j1);
            fread(j1, 1, 4, fin);
            number_of_lines = (int)atof(j1);
            pic_size=number_of_columns*number_of_lines;
            cout<<number_of_columns<<" " <<number_of_lines<<endl;
        }
    }
}

```

```
    }
    if (pic_size == 0)
        return false;
    unsigned char *buffer = new unsigned char[pic_size];

    // copy file
    fread(buffer, 1, pic_size*sizeof(unsigned char), fin);
    fwrite(buffer, 1, pic_size*sizeof(unsigned char), fraw);

    delete[] buffer;
    return true;
}

void main()
{
    FILE *fin, *fraw;

    // open input file
    if ((fin = fopen("FILE_NAME.pgm", "rb")) == NULL)
    {
        cout<<"No such file exists"<<endl;
        exit(0);
    }

    // make sure the output file can be opened
    if ((fraw = fopen("OTHER_FILE_NAME.raw", "wb+")) != NULL){
        if (!removeHeader(fin, fraw))
            cout<<"Error: problematic pgm file"<<endl;

        // close file
        fclose(fraw);
    }
    fclose(fin);
}
```