CS4551 Multimedia Software Systems

PPM Image Format

You will be using the PPM (**P**ortable **P**ixel **M**ap) file format to store raw (uncompressed) COLOR image data. The PPM format is one of the simplest color image file formats.

The PPM format consists of two parts: a `header` and a sequence of *`color pixel data`*. Here is a PPM example.

```
P6
# comments
160 120
255
RGBRGBRGBRGBRGB (Raw image data in RGB order) . . . .
```

Header

- The first line contains the PPM identifier `P6`. It means that the file is a PPM file containing color pixels.
- The comment lines preceded by `#` symbol are *optional*. For our homework, assume that there is always one comment line.
- The next line following comment lines contains the input image size, width and height in pixel. In this example, the image size is 160 (width) x 120 (height) pixels.
- The last line of the header shows the maximum intensity. The maximum intensity gives the value that a pixel should be set to in order to have full intensity (always 255 in our case).

Color Pixel Data

After the header, PPM format stores a sequence of pixels. Each pixel consists of three bytes (***one byte character triples***) representing red(R), green(G) and blue(B) components of the pixel in that order. The pixels are stored in order from LEFT-to-RIGHT and from TOP-to-BOTTOM starting from the UPPER LEFT corner pixel of the image.

Open a PPM file using a plain text editor (eg. Wordpad) and check out the header and raw image data.

In order to display a PPM file on your screen, use Irfanview (www.**irfanview**.com). Download Irfanview from the website and use it to check your program input and output. In addition, you can perform conversions to other image file formats (.gif, .jpg, .bmp, etc).

Program to read/write a PPM file

Refer to the provided Java programs that read a PPM file and write back into another PPM file.

How to save a 8-bit value (down to a bi-level value) in a PPM file format

The given `Image` class template is designed to store only a 24bit image that require values for R, G, and B per pixel.

8-bit image: Let us assume that you have 8bit pixel depth values ranging [0 -255]. To store/display 8bit information using a 24bit `Image` object, assign the same gray-scale/index value to R, G, and B. The displayed image will be gray-scale.

1-bit binary image: If you have bi-level values BW (0 for black and 255 for white), assign the bi-level values BW to R, G, and B. The output will be displayed as a bi-level image.