

TP1

Image manipulation

In this TP, we will study the image file formats PBM, mostly used for their simplicity. We will implement some simple operation on these formats: reading, writing, conversion and displaying.

Introduction. "Portable BitMap" formats

The image file format PBM (portable bitmap), PGM (portable grayscalemap) and PPM (portable pixmap) propose a simple solution to manipulate image files. In these three formats, an image is considered as a matrix where the values represent the light intensity in each pixel of the image: black or white (PBM), level of gray (PGM) or three level of colors RGB (Red, Green, Blue) (PPM).

Definition: The corresponding files are composed of the following elements:

1. A "magical number" to identify the type of file: P1 or P4 for PBM, P2/P5 for PGM and P3/P6 for PPM.
2. A whitespace character (white, TABs, CRs, LFs).
3. The image **length** (decimal value, encoded in ASCII) followed by a whitespace character. Then the image **height** (decimal value, ASCII) followed by a whitespace character.
4. Only for PGM and PPM : the maximum intensity (decimal value between 0 and 255, encoded in ASCII) followed by a whitespace character.
5. A matrix of number of size $length \times height$. These number are either:
 - (For P1/P2/P3) Decimal values (encoded in ASCII) separated by whitespace
 - (For P4/P5/P6) Binary values directly encoded on 1 or 2 octets. In this case, there is no whitespace between values.

Remarks:

- The lines starting by the character "#" are ignored.
- The lines have to contain less than 70 characters.

Examples

P1

feep.pbm

24 7

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0
0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0
0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 1 0
0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

File PBM of an image 24×7 whose values are coded in ASCII

P2

feep.pgm

24 7

15

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

File PGM of an image 24×7 . The intensity values coded in ASCII are at most 15

P3

pasbeau.ppm

4 4

15

```
0 0 0 0 0 0 0 0 0 15 0 15
0 0 0 0 15 7 0 0 0 0 0 0
0 0 0 0 0 0 0 15 7 0 0 0
15 0 15 0 0 0 0 0 0 0 0 0
```

File PPM of an image 4×4 . The intensity values coded in ASCII are at most 15

Exercise 1. Warming-up

Get the TP1 sources. This folder contains:

- A “library” `Util.h/c` including some useful functions to read files.
- A conversion program `pbmtopxm.c`
- Some images (sub-folder `images/`).

Questions:

- Try the program on the file `feep_P1.pbm`. What type of conversion the program does ?
- How is stored the image on the program ?
- How are the functions `pm_getc` and `pm_getint` of the file `Util.c` used ?
- What are the implied types to manage intensity ? In the case of decimal values in ASCII (P1/2/3) ? In the case of binary values (P4/5/6) ?
- What color is associated to the value of maximum intensity?

Exercise 2. Format conversion intra-PGM

By inspiring oneself of the function `pxmtopbm`, implement the conversion function between the format PGM (i.e. from P2 to P5, or from P5 to P2): `pgmtopgm`

Remark : Before generating a new executable, rename the executable of the previous exercise. To print-out an octet, use `printf("%c",...)`.

Exercise 3. Conversion from PGM to PBM

The PGM format allows us to store images through their gray level. Propose and implement a conversion algorithm from the PGM format to the PBM format.

Exercise 4. Conversion PPM to PGM

Likewise, the PPM format allows us to store images through their RGB color levels. Propose, then implement a conversion algorithm from the PPM format to the PGM format, i.e. store the 3 color levels on one level.

Bonus exercise. Conversion PGM to PPM

Let us consider a PGM image (gray level). We want to convert it to the format PPM, i.e. we want to color it.

Question:

- Think (deeply) about it, and propose some ideas of image coloring algorithm...

A general algorithm consist on using a *colormap*, i.e. a function which associate a color (RGB) to each gray level ¹.

Questions:

- Let us assume that we encode a *colormap* as 3 arrays of type `int [256]`, named `colormapR`, `colormapG` and `colormapB`. Implement a possible conversion function, with the constraints to re-enforce visually the image perception (2 close levels of gray must be associated to 2 close colors, but we associate the level of gray into a wider RGB spectrum)
- For example, propose a *colormap* which transform white roses into red roses...`I know some cards who are interes- euh, I said nothing...`

¹Some examples of colormap: <http://liris.cnrs.fr/dgtal/doc/nightly/moduleIO.html>