

# TD1

## Manipulations d'images

Dans ce TD, nous allons étudier les formats de fichiers d'images PBM, qui sont largement répandus en raison de leur simplicité. Nous effectuerons quelques opérations simples sur ces formats : lecture, écriture, conversion et affichage.

### Introduction. Les formats "portable map"

Les formats de fichiers d'images PBM (portable bitmap), PGM (portable grayscalemap) et PPM (portable pixmap) offrent une solution simple à tout programmeur confronté au problème de la manipulation de fichiers d'images. Dans ces formats, une image est considérée comme une matrice dont les valeurs représentent l'illumination en chaque pixel de l'image : noir ou blanc (PBM), un niveau de gris (PGM) ou trois niveaux de couleurs RGB (rouge, vert, bleu) (PPM).

**Définition:** Les fichiers correspondants sont constitués des éléments suivants :

1. Un "nombre magique" pour identifier les type du fichier : P1 ou P4 pour PBM, P2 ou P5 pour PGM et P3 ou P6 pour PPM.
2. Un caractère d'espacement (blanc, TABs, CRs, LFs).
3. La **largeur** de l'image (valeur décimale, codée en ASCII) suivie d'un caractère d'espacement, la **longueur** de l'image (valeur décimale, ASCII) suivie d'un caractère d'espacement.
4. Uniquement pour PGM et PPM : l'intensité maximum (valeur décimale comprise entre 0 et 255, codée en ASCII) suivie d'un caractère d'espacement.
5. Une matrice de nombres de taille largeur  $\times$  hauteur. Ces nombres sont soit des valeurs décimales codées en ASCII et séparées par des espacements dans le cas des formats P1, P2, P3, soit directement les valeurs binaires sur 1 ou 2 octets dans le cas des formats P4, P5, P6. Dans ce dernier cas, il n'y pas de caractères d'espacement entre les valeurs.

**Remarques:**

- Les lignes commençant par le caractère "#" sont ignorées.
- Les lignes contiennent moins de 70 caractères.

## Exemples

```
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
```

*Fichier PBM d'une image 24×7 dont les valeurs sont codées en 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 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 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
```

*Fichier PGM d'une image 24×7. Les valeurs d'intensité codées en ASCII sont au maximum de 15*

```
P3
# pasbeau.ppm
4 4
15
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
```

*Fichier PPM d'une image 4×4. Les valeurs d'intensité codées en ASCII sont au maximum de 15*

## Exercice 1. Mise en bouche

Récupérez les sources du TP1. Ce répertoire contient:

- Une “bibliothèque” `Util.h/c` comprenant quelques utilitaires pour la lecture de fichiers.
- Un programme de conversion `pbmtopxm.c`
- Quelques fichiers image (sous-dossier `images/`).

### Questions:

- Testez le programme sur le fichier `test.pbm`. Quel conversion ce programme effectue ?
- De quelle manière est stockée l’image dans le programme ?
- À quoi servent les fonctions `pm_getc` et `pm_getint` du fichier `Util.c` ?
- Quels sont les types impliqués pour manipuler les intensités ? dans le cas de valeurs décimales codées en ASCII (P1, P2, P3) ? dans le cas de valeurs binaires (P4, P5, P6) ?
- Quelle couleur est associée à la valeur d’intensité maximum ?

## Exercice 2. Conversion de format intra-PGM

En s’inspirant de la fonction `pxmtopbm`, implémentez la fonction de conversion entre les formats PGM (c’est à dire de P2 vers P5 ou de P5 vers P2) : `pgmtopgm`

**Notes :** Avant de générer le nouvel exécutable, renommer l’exécutable de l’exercice précédent. Pour affichez un octet, utiliser `printf("%c", ...)`.

## Exercice 3. Conversion PGM vers PBM

Le format de fichier d’images PGM permet de stocker des images en niveaux de gris. Proposez et implémentez un algorithme de conversion au format PBM.

## Exercice 4. Conversion PPM vers PGM

De la même manière, le format de fichier d’images PPM permet de stocker des images en niveaux de couleurs RGB. Proposez, puis implémentez un algorithme de conversion au format PGM, c’est à dire stocker les trois niveaux de couleurs sur un seul niveau.

## Exercice bonus. Conversion PGM vers PPM

Considérons une image PGM (image de niveaux de gris). On souhaite la convertir en format PPM, ie la colorier.

### Question:

- Proposez des idées d’algorithmes de coloriage d’image...

Un algorithme général consiste à utiliser une *colormap*, c'ad une fonction qui associe une couleur (RGV) à un niveau de gris <sup>1</sup>.

**Questions:**

- On suppose qu'on encode la *colormap* comme 3 tableaux de type `int[256]`, nommés `colormapR`, `colormapG` et `colormapB`. Implémentez une fonction de conversion possible avec comme contrainte de renforcer visuellement la perception de l'image (deux niveaux de gris proches doivent être associée à des couleurs proches mais on associe les niveaux de gris sur une plus large spectre RVB).
- Par exemple, proposez une *colormap* qui colorie des roses blanches en rouge...

---

<sup>1</sup>Quelques exemples de *colormap*: <http://liris.cnrs.fr/dgtal/doc/nightly/moduleI0.html>