

# UJF - M1 Informatique 2005-2006 : Infographie

## TP1 : Images et espaces de couleur

30 septembre 2005

Dans ce premier TP nous verrons comment manipuler (lire et écrire) des fichiers images simples et ensuite nous utiliserons divers espaces dans le but d'appliquer différents traitements aux images.

## 1 Images au format PNM : Portable aNy Map

### 1.1 Description du format de fichier PNM

Le format PNM est un format d'images, sûrement le plus simple qu'il existe. Il permet très simplement de décrire des images en *bitmap* (2 couleurs : "image.pbm"), *graymap* (niveaux de gris : "image.pgm") et *pixmap* (couleurs RGB : "image.ppm"). Nous nous intéresserons uniquement à la version "ASCII" du format PNM.

- Reportez vous aux *manpages* associées pour plus de détails (`man pnm`).
- Pour visualiser un fichier image, utilisez la commande `xv nom_du_fichier` ou bien `display nom_du_fichier`.
- Dans un fichier de type pnm, toute ligne commençant par un `#` est considérée comme un commentaire (et ne contient donc pas de données utiles à l'image).

#### 1.1.1 Fichiers PBM (`man pbm`)

```
P1
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
```

#### En-tête

- P1 indique que les données de l'image vont être au format "PBM", c'est-à-dire un bit pour chaque pixel.

- 24 7 est la dimension de l'image : 24 pixels de large pour 7 pixels de haut.

### Données

Vient après la suite de 24x7=168 bits notés par 0 ou 1 (0 correspondant au blanc, 1 au noir).

1. Ecrivez un programme qui puisse lire et écrire un fichier au format PBM.
2. Dans ce programme, écrivez une fonction qui permette d'inverser les couleurs (ici le blanc et le noir).

Conseils :

- Pensez à séparer la lecture (et l'écriture) de l'en-tête et des données dans deux fonctions séparées
- Stockez les données (l'image) dans une matrice (tableau à deux dimensions, un `int **image` par exemple). Pour cela, écrivez une fonction qui alloue une matrice de taille LxH.

#### 1.1.2 Fichiers PGM (man pgm)

Le format PGM fonctionne de manière similaire. Dans l'en-tête, P1 est remplacé par P2 et une ligne est ajoutée après la taille de l'image, indiquant la valeur maximum du niveau de gris (dans l'exemple 15) :

```
P2
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 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 0 0 0 0 0 0 0 0
```

**Modifiez votre programme pour qu'il puisse lire aussi ce format d'images (le format est déterminé par la première ligne : P2).**

#### 1.1.3 Fichiers PPM

1. Lire la *manpage* : `man ppm`. Attention : maintenant il y a 3 couleurs (3 int) correspondant au canaux Rouge, Vert et Bleu par pixels!
2. Ecrire une fonction qui fasse le négatif de l'image (prendre l'opposé des couleurs).
3. Ecrire une fonction qui transforme naïvement une image couleur en niveaux de gris.

## 1.2 Manipulation des images

Utilisez l'image `lena.ppm`, il faut appliquer les traitements sur chaque canal de couleur.

### 1.2.1 Lissage

1. Ecrivez une fonction qui lisse l'image : pour chaque pixel, la valeur du nouveau pixel est la moyenne des pixels autours.
2. Ecrivez une fonction qui lisse l'image avec un noyau gaussien sur un carré de 3x3 pixels.
3. Lissez plusieurs fois l'image `lena_bruitee.pgm`. Qu'observez vous ?

### 1.2.2 Redimensionnement (naïf)

1. Ecrivez une fonction qui permette de diviser par deux la taille d'une image par différentes méthodes : prendre un pixel sur deux, prendre la moyenne des pixels autours, prendre un pixel sur deux puis lisser.
2. Ecrivez une fonction qui permette de multiplier par deux la taille d'une image par différentes méthodes : prendre le pixel le plus proche, prendre la moyenne des pixels autours, prendre le pixel le plus proche puis lisser.

## 2 Espaces de couleur

Les fichiers au format PPM sont écrits uniquement en RGB.

### 2.1 Manipuler l'espace RGB

Jouez avec les différents canaux pour obtenir différents résultats :

1. Visualisez les 3 canaux dans 3 fichiers différents en niveaux de gris (séparation des canaux).
2. Masquez un canal (c'est-à-dire mettez toutes ses valeurs à 0) et visualisez l'image couleur correspondante.
3. Inversez deux canaux
4. Appliquez plusieurs fois le lissage sur un seul des canaux séparément (essayez cela sur les 3 canaux). Quelle observation pouvez vous faire ?
5. Atténuez, augmentez (par addition ou multiplication), les valeurs dans un des canaux.

### 2.2 L'espace HSV (Teinte, Saturation, Valeur) (équivalent HSB)

1. Ecrivez une fonction qui passe d'une image RGB à une image en HSV et inversement.

2. Visualisez les 3 canaux H,S et V dans 3 fichiers différents en niveaux de gris.
3. Jouez avec les différents canaux avant de repasser l'image en RGB : augmentation/réduction de la saturation, valeur

### **2.3 L'espace XYZ**

1. Ecrivez une fonction qui passe d'une image RGB à une image en XYZ et inversement.
2. Créez une version en nuances de gris à partir de la luminosité. Que pouvez vous dire par rapport à la version naïve (directement à partir du RGB) ?