

Sujet de TP 3 : Image numérique et Python

Première Partie :

Manipulation de fichier d'image simple : le format BPM

Une image numérique est une suite d'informations stockées dans un fichier. Ce qui indique que le fichier contient une image (et pas autre chose) ce sont les extensions, comme par exemple : jpg, png, gif, bmp etc... On parle de format jpg, format gif...

Un format simple pour s'initier au codage d'une image est le format Portable PixMap.

Celui-ci se décline en 3 modes selon le degré de coloration souhaité : noir et blanc, dégradés de gris, couleurs. Les images utilisées ou construites dans cette activité seront visionnées avec un logiciel comme *GIMP2*.

Le format pbm pour portable bit map

Un fichier au format *pbm* comporte des informations diverses pour aider le logiciel à afficher l'image. Voici ces informations :

- P1
- Un caractère d'espacement (espace, nouvelle ligne)
- Largeur de l'image
- Un caractère d'espacement
- Hauteur de l'image
- Un caractère d'espacement
- Données de l'image :
 - ✓ L'image est codée ligne par ligne en partant du haut
 - ✓ Chaque ligne est codée de gauche à droite
 - ✓ Un pixel noir est codé par un 1, un pixel blanc est codé par un 0 (d'où le nom *bit map*)
 - ✓ Les caractères d'espacement à l'intérieur de cette section sont ignorés

Exemple :

```
P1
# Exemple de fichier PBM :
# ces lignes sont des
# commentaires
20 12
00000000000000000000
00001111111111110000
00001100000001100000
00001100000001100000
00001100000001100000
00001100000001100000
00001100000001100000
00000111111111000000
00001100000001100000
00001100000001100000
00001100000001100000
00001100000001100000
00001111111111000000
```

Travail à faire:

- Copier le contenu de ce cadre dans un éditeur de texte et l'enregistrer au format *.pbm*
- Ouvrir le fichier avec GIMP et le zoomer à 1600% (clic droit – affichage – zoom).
- Ecrire un programme python qui ouvre un fichier PBM et crée un second fichier (même nom fichier mais suivi de Reverse, par exemple im1.pbm -> im1Reverse.pbm). Le second fichier sera le « négatif » du premier (les pixels blancs deviendront noirs et réciproquement).

Le format *pgm* : des dégradés de gris

Difficile de rendre une image attrayante avec seulement le noir et le blanc. Le format *pgm* (Portable Grey Map) permet de rendre des dégradés de gris sur une échelle allant de 0 (noir) à 255 (blanc).

La nomenclature du fichier est sensiblement la même que pour le *pbm*, on remplace les 0 et les 1 par les nombres décimaux correspondant « au code de gris ». Plus ce nombre est élevé, plus le gris est proche du blanc.

- P2
- Un caractère d'espacement (espace, nouvelle ligne)
- Largeur de l'image
- Un caractère d'espacement
- Hauteur de l'image
- Un caractère d'espacement
- La valeur maximale utilisée pour coder les niveaux de gris, cette valeur doit être inférieure à 65536
- Un caractère d'espacement
- Données de l'image :
 - ✓ L'image est codée ligne par ligne en partant du haut
 - ✓ Chaque ligne est codée de gauche à droite
 - ✓ Chaque pixel est codé par un nombre, précédé et suivi par un caractère d'espacement. Un pixel noir est codé par la valeur 0, un pixel blanc est codé par la valeur maximale et chaque niveau de gris est codé par une valeur entre ces deux extrêmes, proportionnellement à son intensité.

Exemple :

```

P2
# Exemple de fichier PBM :
#taille 10 par 10
10 10
#50 = nuance max=blanc
50
15    50    50    50    50    50    50    50    50    50
15    20    20    20    20    20    20    20    20    15
15    15    15    20    20    20    20    20    15    15
15    15    15    0    0    0    0    0    15    15
15    15    15    0    0    0    0    0    15    15
15    50    50    0    0    0    0    0    50    50
15    50    50    0    0    0    0    0    50    50
15    50    50    20    20    20    20    20    50    50
15    50    50    20    20    20    20    20    50    50
15    50    50    50    50    50    50    50    50    50

```

Travail à faire :

- Copier le contenu de ce cadre dans un éditeur de texte et l'enregistrer au format *.pgm*.
- Ouvrir le fichier avec GIMP et le zoomer à 1600% (clic droit – affichage – zoom).
- Ecrire un programme python qui ouvre un fichier PGM et crée un second fichier (même nom fichier mais suivi de Reverse, par exemple im1.pgm -> im1Reverse.pgm). Le second fichier sera le « négatif » du premier (les pixels clairs deviendront foncés et réciproquement).

L'image en Rouge Vert Bleu (Portable PixMap)

Le format correspondant aux deux formats simples précédents, adapté à la couleur est le format *ppm* (Portable PixMap).

Le début du fichier est très ressemblant aux deux précédents :

```

P3
# Exemple de fichier PBM :
10 10
255

```

On retrouve les mêmes paramètres, type (P3), ligne de commentaire, dimensions puis la valeur maximale pour l'intensité des couleurs. Celle-ci est couramment 255.

On code ensuite chaque ligne pixel par pixel avec les trois composantes RVB en partant de gauche à droite.

On peut donc trouver un début comme celui-ci :

```

P3
# exemple 3 colonnes and 2 lignes,
3 2
255
150 0 0 0 255 0 0 255 255
120 200 075 255 0 255 075 0 0

```

Nous remarquons que pour coder une minuscule image de $3 \times 2 = 6$ pixels, nous avons utilisé 18 codes couleurs.

Voici le rendu (très « zoomé ») de cet exemple....



Travail à faire :

- Ecrire un programme python qui demande à l'utilisateur la largeur et la hauteur (en pixels) désirées et qui crée un fichier *francais.ppm* et qui codera une image portable pixmap du drapeau français (bleu blanc rouge) selon les dimensions souhaitées par l'utilisateur.
- Ouvrir avec gimp l'image cassis.JPG et l'exporter au format ppm (si l'option est demandée répondre ASCII)
-



La photo cassis.JPG

- Ouvrir le fichier avec un éditeur de texte
- Pour convertir un pixel R, V, B en niveau de gris G, G, G il suffit de calculer $G = (R+V+B)/3$. On notera que G doit être entier. Ecrire un programme Python qui convertit le fichier cassis.ppm en cassisNG.ppm correspondant à l'image en niveau de gris.
- Ecrire un programme python qui convertit une image PPM (couleur) en image PGM (niveaux de gris)

Séance 2

1 Principe d'un filtre *simple*

On s'intéresse d'abord aux filtres qui fonctionnent de la façon suivante :

Pour chaque pixel d'une image : On lit les composantes RVB On effectue une opération sur les composantes RVB et on obtient ainsi de nouvelles composantes. On remplace les anciennes composantes par les nouvelles.

2. Filtre niveaux de gris

Pour passer d'une image couleur à une image en niveaux de gris, il faut remplacer chaque composante par une même valeur L, appelée luminance (à ne pas confondre avec la luminosité du mode TSL). La luminance est le résultat d'un calcul de moyenne sur les 3 composantes d'un pixel arrondie à l'entier le plus proche. L est donc un entier entre 0 (noir) et 255 (blanc).

On constate qu'avec la moyenne non pondérée, il est impossible de distinguer les trois dernières couleurs en niveaux de gris.

Pour éviter cela, on utilise une moyenne pondérée. Les pondérations dépendent de la sensibilité des couleurs primaires à l'œil humain et du support utilisé pour afficher l'image.

La formule couramment utilisée est: $L = 0,21R + 0,71V + 0,08B$.

Modifier le code de conversion d'une image PPM en PGM en tenant compte de cette formule

3. Filtre noir et blanc

Le principe du filtre noir et blanc reprend en partie celui du niveaux de gris en fixant un seuil.

On fixe un seuil entre 0 et 255. On calcule la luminance L du pixel. Si L est supérieure au seuil, alors on remplace les trois composantes RVB par 255. Sinon, on remplace les 3 composantes par 0

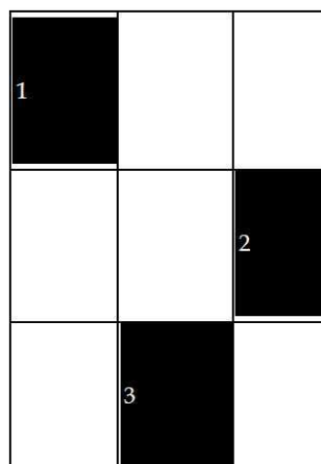
Ecrire un programme python qui convertit une image PPM (couleur) en image PBM (Noir ou blanc)

4. Détection du contour

L'objectif est de ne garder que les contours d'une image bicolore en noir et blanc.

Il n'y a pas qu'une seule technique pour réaliser un tel filtre. Certaines sont complexes. En voici une assez simple qui donne des résultats très corrects :

Pour chaque pixel, nous allons compter combien il y a de pixel(s) noir(s) autour. Dans l'exemple ci-dessous, le pixel central est entouré de 3 pixels noirs et de 5 blancs.



On ne peut pas modifier directement l'image de départ sans altérer le comptage. On a donc besoin d'ouvrir au départ deux images identiques. La première sera lue pour le comptage mentionné ci-dessus (l'originale). La seconde image servira de support pour "écrire" (l'image finale) en utilisant la règle suivante: Si le nombre de pixel(s) est compris entre 3 et 6 inclus, on considère que l'on est près d'un contour et on écrit un pixel noir dans l'image finale. Sinon, on considère que l'on est dans une zone unicolore blanche ou noire et on écrit un pixel blanc dans l'image finale.

Ecrire un programme python qui convertit une image PPM (couleur) en image PBM (Noir ou blanc) ne contenant que le contour des objets. On pourra prendre comme image le chat.