

Projet 2: traitement d'images

17 avril 2005

1 But du projet

Le but du projet est de réaliser des modifications sur une image contenue dans un fichier au format PPM (Portable PixMap). Ces modifications porteront sur la couleur ou la forme des images. De plus, on impose que vous utilisiez une certaine représentation en mémoire des images, basée sur des listes chaînées (pointeurs).

2 Fichier d'image

Une image peut être considérée comme une grille de points (ou pixels) ayant chacun sa propre couleur. Une couleur peut être décrite comme un mélange de rouge, de vert et de bleu (exactement comme un écran de télévision). Le nombre de points par ligne est appelé la largeur de l'image et le nombre de lignes sa hauteur.

On va utiliser un format de fichier d'image très simple à créer. Un fichier PPM est composé de la façon suivante. Sur la première ligne, il y a écrit **P3**. Sur la deuxième ligne il y a écrit le caractère # suivi d'un commentaire, par exemple le nom du fichier. Sur la troisième ligne, il y a la largeur et la hauteur de l'image, séparées par un espace, sur la quatrième ligne, il y a un nombre entier qui est le plus grand nombre utilisable dans la représentation des couleurs. C'est généralement une puissance de 2 moins 1. Supposons que ce soit le nombre 15. Sur les lignes suivantes, il y a la couleur de chaque point donnés par 3 nombres compris entre 0 et 15, séparés par un espace (ou plusieurs espaces) représentant respectivement la quantité de rouge, de vert et de bleu présents dans le mélange. Les fins de ligne du fichier sont considérées comme de simples espaces. Les lignes doivent en principe comporter moins de 70 caractères.

Pour dire qu'un point est rouge, on donnera les valeurs 15 0 0. Cela donne la couleur obtenue en mélangeant le maximum de rouge (15) avec pas du tout de vert ni de bleu. Le vert s'obtient avec les nombres 0 15 0. Le bleu avec 0 0 15. Le jaune avec 15 15 0. Le noir avec 0 0 0. Le blanc avec 15 15 15. En mettant moins de 15, par exemple, 12 12 0, on obtient un jaune plus foncé.

Les points sont donnés dans l'ordre suivant : de gauche à droite et de haut en bas. Le premier point donné est celui qui est en haut à gauche, puis le deuxième point de la première ligne et ainsi de suite jusqu'au point le plus à droite de la première ligne. Ensuite viennent les points de la seconde ligne, etc.

3 Exemple de fichier image

```
P3
#truc.ppm
4 4
15
0 0 0      0 0 0      0 0 0      15 0 0
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 15 0     0 0 0      0 0 0      0 0 0
```

On a une image de taille 4 points par 4 points (4 lignes, 4 colonnes). Il y a donc 16 points en tout. Chaque point est décrit par 3 entiers, il y a donc $16 \times 3 = 48$ nombres entre 0 et 15. Les trois premiers points de la première ligne sont noirs. Le dernier est rouge. Le quatrième point de la première colonne est jaune.

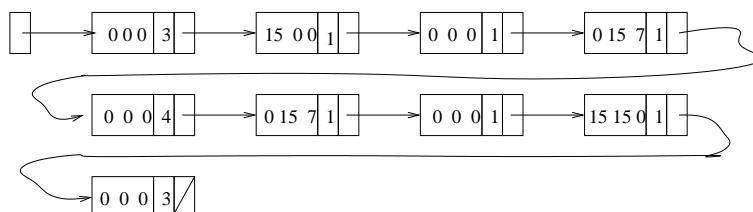
Vous pouvez taper ce fichier avec un éditeur de texte (nedit, edit). Il faut lui donner le nom truc.ppm. Vous pouvez ensuite voir l'image en utilisant un visionneur d'images (la page <http://www.webattack.com/Freeware/gmm/fwviewers.shtml> référence une palette de visionneurs d'images gratuits). Sur les machines du CNAM et la plupart des distributions LINUX, la commande `xv` permet de visualiser les images au format PPM.

L'image donnée en exemple est toute petite. Il est très long de taper à la main le contenu d'une grande image, d'où l'intérêt d'un programme qui manipule des images déjà existantes (vous en trouverez une collection sur le site web du cours). Java ne permettra pas d'afficher l'image, seulement de lire et d'écrire le fichier la contenant.

4 Structure de données

Cette partie concerne la structure de donnée que l'on vous demande d'utiliser pour la représentation en mémoire d'une image PPM. On va considérer une image comme une séquence de points dont les couleurs sont notées par trois entiers. Autrement dit, dans notre représentation (comme dans le fichier PPM), on ne prend pas en compte les fins de lignes. On définit un segment comme une séquence de points consécutifs de même couleur. Pour coder un segment, il faut les trois entiers de la couleur plus le nombre de points. On vous demande de représenter une image comme un enregistrement contenant la largeur et la hauteur de l'image, l'entier de la quatrième ligne (valeur maximale d'une couleur) et la liste chaînée de segments.

Exemple : la liste des segments correspondant au fichier truc.ppm donné ci-dessus sera représentée en mémoire de la façon suivante.



Vous devrez définir deux sous-programmes permettant de transférer le contenu d'un fichier dans une liste de segments et le contenu d'une liste dans un fichier. Vous éviterez autant que possible

d'utiliser une structure intermédiaire (comme un tableau).

5 Modification de l'image

5.1 Opérations obligatoires

On vous demande de créer des sous-programmes permettant de réaliser chacune des opérations suivantes :

- Foncer ou éclaircir toutes les couleurs ayant une dominante rouge, bleue ou verte. La dominante d'une couleur est l'une des trois couleurs primaire (rouge, vert, bleu), celle qui a l'entier le plus grand. Pour foncer ou éclaircir une couleur, on ajoute ou on retranche un certain entier à chacun des trois entiers la décrivant. Par exemple, si on veut foncer les dominantes rouges, on changera par exemple la couleur 10 5 6 en 14 9 10. En revanche 2 15 10 restera inchangée parce que cette couleur n'a pas une dominante rouge mais une dominante verte.
- Passer en noir et blanc, en utilisant des niveaux de gris. Les gris sont obtenus en donnant aux trois nombres la valeur de la moyenne des trois valeurs d'origine.
- Afficher la taille courante de l'image (largeur x hauteur), en nombre de pixels.
- Découper et garder seulement une partie de l'image, entre les lignes $l1$ et $l2$ et les colonnes $c1$ et $c2$. Les valeurs $l1$, $l2$, $c1$ et $c2$ sont précisées par l'utilisateur, qui peut se baser sur le nombre de lignes et de colonnes de l'image, affichées par l'opération précédente. Si cela vous semble trop difficile, vous pouvez programmer une variante simplifiée, où l'image est découpée entre les lignes $l1$ et $l2$, en gardant toutes les colonnes.

5.2 Opérations optionnelles

En plus des opérations obligatoires vous pourrez réaliser une ou plusieurs des opérations suivantes.

- Créer le négatif d'une image. Pour cela, on change chaque couleur dans son complémentaire.
- Elargir une image.
- Foncer ou éclaircir les couleurs comprises dans une certaine plage de couleurs, caractérisées par des intervalles de couleurs primaires (ou autre caractérisation).
- Agrandir ou rapetisser une image dans les deux dimensions (c'est beaucoup plus difficile que les opérations précédentes).
- Incruster dans l'image des rectangles d'une certaine couleur (c'est également une opérations difficile).
- incruster une image sur un nouveau fond. Il s'agit de remplacer dans l'image à incruster un fond d'une couleur unie (par exemple blanc) par des couleurs venant d'un autre image (voir l'exemple à la fin de ce document). Pour simplifier la tâche, on peut imposer que les deux images aient la même taille. C'est là l'opération la plus difficile.

Vous pourrez également imaginer et réaliser d'autres transformations.

6 Interface

Vous réaliserez un menu proposant de lire une image contenue dans un fichier, d'écrire une image transformée dans un fichier et de réaliser les différents opérations.

En option, vous pourrez proposer d'avoir plusieurs images en mémoire à un moment donné.

7 Informations pratiques

Vous devrez rendre le programme lors de la semaine du 1er juin, à votre groupe de TP habituel.

Vous pouvez poser vos questions sur le projet à vos enseignants de TP. Des renseignements supplémentaires seront publiés sur la page web du cours APA.

8 Exemples de transformation

Nous vous conseillons de traiter des images de taille relativement réduite. Nous donnons des exemples de fichiers PPM téléchargeables sur la page web.

Nous illustrons ici le type de résultat que nous attendons de votre programme. Nous partons de l'image suivante :



dont vous pouvez trouver le fichier PPM [ici](#) et en version compressée (zip) [ici](#).

La transformation sur les couleurs donne le résultat suivant :

dominante rouge



dominante verte



dominante bleue



La mise en noir et blanc :



L'élargissement :



Et voici l'image obtenue en enchaînant élargissement, fonçage du bleu et passage en noir et blanc :



En découpant entre les lignes 7 et 108 et les colonnes 22 et 220, on obtient :



Pour terminer, une incrustation :

