

Algorithmique des images

TD n° 1

Manipulation simple d'images

Exercice 1. Manipulation simple d'images "Gray scale" (PGM)

Le format de fichier **Portable GrayMap** (pgm) permet d'enregistrer des images en niveaux de gris ou chaque pixel est codé par une valeur entière. Ainsi en ouvrant une image de format (pgm) préalablement enregistrée en ASCII avec un éditeur de texte type gedit (ou bien emacs) vous pourrez observer le code de chaque pixel. Par exemple en ouvrant le fichier `eye_s_asc.pgm` vous observerez le format suivant :

```
1      P2          <-- Code pour le format PGM
2      # Created by GIMP version 2.10.36 PNM plug-in
3      512 423      <-- Largeur et hauteur de l'image
4      255          <-- pixel 0
5      144          <-- pixel 1
6      144          <-- pixel 2
7      143          <-- pixel 3
8      145          <-- pixel 4
9      145          <-- pixel 5
10     144          <-- pixel 6
11     ...
```

Q- 1.1 Structure

Écrire une structure `pgm` qui contiendra la hauteur (`height`), la largeur (`width`) la valeur maximale de codage des pixels (`max_value`) ainsi qu'un pointeur sur un tableau à deux dimensions de caractères non-signés (`pixels`).

Q- 1.2 Allocation

Écrire une fonction `pgm_alloc` qui prend en paramètre la hauteur (`height`), la largeur (`width`) la valeur maximale de codage des pixels (`max_value`) et qui retourne une structure `pgm` contenant les données d'une image de taille `height x width` de pixels initialisés à la valeur `max_value`.

Q- 1.3 Libération

Écrire une fonction `pgm_free` qui prend en paramètre un pointeur sur une structure `pgm` et libère l'espace mémoire occupé par cette structure.

Q- 1.4 Lecture ASCII

Écrire la fonction `pgm_read_asc` qui prendra en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier au format ASCII à lire (`fname`) et retournant un pointeur sur une structure `pgm` contenant les informations relatives à l'image contenue dans le fichier `fname`.

Q- 1.5 Écriture ASCII

Écrire la fonction `pgm_write_asc` qui prendra en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier (`fname`) à écrire (au format ASCII) ainsi qu'un pointeur sur une structure `pgm`. La fonction retournera un entier égale à 0 si tout s'est bien passé et à 1 sinon.

Q- 1.6 Lecture BINAIRE

Écrire la fonction `pgm_read_bin` qui prendra en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier au format BINAIRE à lire (`fname`) et retournant un pointeur sur une structure `pgm` contenant les informations relatives à l'image contenue dans le fichier `fname`.

Q- 1.7 Écriture BINAIRE

Écrire la fonction `pgm_write_bin` qui prendra en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier (`fname` à écrire (au format BINAIRE) ainsi qu'un pointeur sur une structure `pgm`. La fonction retournera un entier égale à 0 si tout s'est bien passé et à 1 sinon.

Q- 1.8 Négatif

Écrire la fonction `pgm_negative` qui prendra en paramètre un pointeur `src` sur une structure `pgm` contenant l'image source et un pointeur `dst` sur une structure `pgm` contenant le négatif de l'image source.

Q- 1.9 Extraction

Écrire la fonction `pgm_extract` qui en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier de sortie (`fname`), une structure `pgm_t`, les coordonnées `dx` et `dy` indiquant le point de départ de l'image à extraire et les dimensions de l'image à extraire `width` et `height`. La fonction écrira dans le fichier `fname` une "sous-image" extraite de l'image principale.

Q- 1.10 Histogramme

Écrire la fonction `pgm_get_histogram` qui prendra en paramètre un pointeur sur une structure `pgm` et qui retournera un pointeur sur un tableau de `max_value` contenant l'histogramme des pixels de l'image.

Q- 1.11 Fichier histogramme

Écrire la fonction `pgm_write_histogram` qui prendra en paramètre un pointeur sur une structure `pgm`, un pointeur sur une chaîne de caractère `fname`. La fonction devra créer le fichier `fname` et l'histogramme de l'image sous la forme de deux colonnes (la première colonne contiendra les valeurs de 0 à `max_value`, la seconde les données de l'histogramme correspondant).

Q- 1.12 Test

Écrire un programme de test `test_pgm.c` afin de tester toutes les fonctions précédentes.

Exercice 2. Manipulation simple d'images "Couleurs" (PPM)

Le format de fichier **Portable PixMap** (ppm) permet d'enregistrer des images en couleur où chaque pixel est codé en RGB. Le système RGB code un pixel par 3 octets, chaque octet prend donc une valeur entre 0 et 255, donnant respectivement le ton de rouge, vert et bleu. Ainsi en ouvrant une image de format (ppm) préalablement enregistrée en ASCII avec un éditeur de texte type `gedit` (ou bien `emacs`) vous pourrez observer le code RGB de chaque pixel. Par exemple en ouvrant le fichier `eye_s_asc.ppm` vous observerez le format suivant :

```
1 P3          <-- Code pour le format PPM
2 # Created by GIMP version 2.10.36 PNM plug-in
3 512 423     <-- largeur et hauteur
4 255        <-- Valeur Max d'une composante couleur
5 181        <-- code R du pixel 0
```

```

6      133      <-- code G du pixel 0
7      110      <-- code B du pixel 1
8      181      <-- code R du pixel 1
9      133      <-- code G du pixel 1
10     ...

```

où les pixels sont numérotés ligne par ligne. Le pixel du coin supérieur gauche est à la position 0, son voisin de droite est à la position 1 et ainsi de suite :

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14

```

1      struct rgb
2      {
3          unsigned char r, g, b;
4      };
5      typedef struct rgb rgb_t;

```

Pour simplifier, nous supposons que les valeurs des composantes RGB sont limitées à 255.

Q- 2.1 Structure

Écrire une structure ppm qui contiendra la hauteur (height), la largeur (width), une valeur maximale (max_value) et un pointeur sur un tableau à deux dimensions de structure rgb (pixels).

Q- 2.2 Allocation

Écrire une fonction ppm_alloc qui prend en paramètre la hauteur (height), la largeur (width) la valeur maximale de codage des pixels (max_value) et qui retourne une structure ppm contenant les données d'une image de taille height x width de pixels initialisés à la valeur max_value.

Q- 2.3 Libération

Écrire une fonction ppm_free qui prend en paramètre un pointeur sur une structure ppm et libère l'espace mémoire occupé par cette structure.

Q- 2.4 Lecture ASCII

Écrire la fonction ppm_read_asc qui prendra en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier au format ASCII à lire (fname) et retournant un pointeur sur une structure ppm contenant les informations relatives à l'image contenue dans le fichier fname.

Q- 2.5 Écriture ASCII

Écrire la fonction ppm_write_asc qui prendra en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier (fname) à écrire (au format ASCII) ainsi qu'un pointeur sur une structure ppm. La fonction retournera un entier égale à 0 si tout s'est bien passé et à 1 sinon.

Q- 2.6 Lecture BINAIRE

Écrire la fonction ppm_read_bin qui prendra en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier au format BINAIRE à lire (fname) et retournant un pointeur sur une structure ppm contenant les informations relatives à l'image contenue dans le fichier fname.

Q- 2.7 Écriture BINAIRE

Écrire la fonction `ppm_write_bin` qui prendra en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier (`fname` à écrire (au format BINAIRE) ainsi qu'un pointeur sur une structure `ppm`. La fonction retournera un entier égale à 0 si tout s'est bien passé et à 1 sinon.

Q- 2.8 Négatif

Écrire la fonction `ppm_negative` qui prendra en paramètre un pointeur `scr` sur une structure `ppm` contenant l'image source et un pointeur `dst` sur une structure `ppm` contenant le négatif de l'image source.

Q- 2.9 Extraction

Écrire la fonction `ppm_extract` qui en paramètre un pointeur sur une chaîne de caractères contenant le nom du fichier de sortie (`fname`), une structure `ppm_t`, les coordonnées `dx` et `dy` indiquant le point de départ de l'image à extraire et les dimensions de l'image à extraire `width` et `height`. La fonction écrira dans le fichier `fname` une "sous-image" extraite de l'image principale.

Q- 2.10 Histogramme

Écrire la fonction `ppm_get_histogram` qui prendra en paramètre un pointeur sur une structure `ppm` et qui retournera un pointeur sur un tableau à deux dimensions (`3,=max_value=`) contenant les histogrammes des trois composantes RGB des pixels de l'image.

Q- 2.11 Fichier histogramme

Écrire la fonction `ppm_write_histogram` qui prendra en paramètre un pointeur sur une structure `ppm`, un pointeur sur une chaîne de caractère `fname`. La fonction devra créer le fichier `fname` et l'histogramme de l'image sous la forme de quatre colonnes (la première colonne contiendra les valeurs de 0 à `max_value`, les trois colonnes suivantes les données de l'histogramme des trois composantes correspondant).

Q- 2.12 Conversion

Écrire la fonction `ppm_to_pgm` qui prendra en paramètre un pointeur sur une structure `ppm` et un pointeur sur une structure `pgm`. La fonction convertira l'image `ppm` en image `pgm`.

Q- 2.13 Test

Écrire un programme de test `test_ppm.c` afin de tester toutes les fonctions précédentes.