

Compte rendu TP1 INF304

Deuxième partie :

Exercice 1

3. Le programme crée une image avec le nom que l'on donne comme deuxième argument qui est une copie de l'image donnée en premier argument.

Exercice 2

transformation_image.c : c'est le main qui vérifie que les arguments sont corrects et qui appelle les fonctions des sous-programmes récupérant l'image initiale, la modifiant et en la ré-écrivant dans un nouveau fichier.

image_io.c : définit les fonctions de lecture et d'écriture de l'image.

operations_image.c : contient les fonctions qui modifient une image (initialement uniquement la fonction d'inversion).

type_image.c : définit le type pixel, la structure image puis les fonctions qui permettent de les manipuler (l'initialisation d'une image et la récupération de valeurs particulières d'une image).

Chaque fichier est accompagné d'un .h qui permet de définir des variables globales, des types et déclare les fonctions utilisées dans le .c.

Après avoir complété le programme *inverse_video*, le programme crée une image avec le nom que l'on donne comme deuxième argument qui est une copie modifiée de l'image donnée en premier argument. Chaque pixel est alors inversé c'est-à-dire qu'un pixel sombre devient clair et inversement.

Exercice 3

Les transformations ajoutées dans le programme sont :

- *changer_luminosité* : la fonction prend en argument un décalage qui sera la valeur ajoutée à chaque pixel. Une valeur positive éclaircira l'image et inversement.
- *noir_et_blanc* : la fonction passe l'image en noir et blanc au lieu de nuances de gris, à partir d'une certaine valeur de pixel on change la valeur du pixel soit en blanc, soit en noir.

```

#include "operations_image.h"

/* Calcule l'image "inverse" (négative) de l'image im1, et place le
   résultat dans l'image im2 */

void inverse_video(Image *im1, Image *im2) {
    Pixel p, p_inverse;
    int l = largeur(im1);
    int h = hauteur(im1);
    int max = maxval(im1);
    /* Indices de parcours de boucles */
    int i, j;

    /* Initialisation de l'image résultat, de mêmes largeur, hauteur
       et valeur max que l'image donnée en paramètre */
    init_image(l, h, max, im2);

    /* Parcours de chaque pixel de chaque ligne */
    for (i = 0; i < l; i++) {
        for (j = 0; j < h; j++) {
            /* A COMPLÉTER
               sélection du pixel (i,j) dans l'image Im :*/
            p = lepixel(im1, i, j);
            /*calcul de son "inverse video" :*/
            p_inverse = max - p;
            /*et affectation du résultat dans la nouvelle image im2 :*/
            affecter_pixel(im2, i, j, p_inverse);
        }
    }
}

void changer_luminosite(Image *im1, Image *im2, int decalage/*entre
-max et max*/) {
    Pixel p, p_modifie;
    int l = largeur(im1);
    int h = hauteur(im1);
    int max = maxval(im1);
    /* Indices de parcours de boucles */
    int i, j;

    /* Initialisation de l'image résultat, de mêmes largeur, hauteur
       et valeur max que l'image donnée en paramètre */
    init_image(l, h, max, im2);

```

```

/* Parcours de chaque pixel de chaque ligne */
for (i = 0; i < l; i++) {
    for (j = 0; j < h; j++) {
        p = lepixel(im1, i, j);
        p_modifie = p + decalage;
        if (p_modifie > 255){
            p_modifie = 255;
        }
        if (p_modifie < 0){
            p_modifie = 0;
        }
        affecter_pixel(im2, i, j, p_modifie);
    }
}

void noir_et_blanc(Image *im1, Image *im2) {
    Pixel p, p_modifie;
    int l = largeur(im1);
    int h = hauteur(im1);
    int max = maxval(im1);
    /* Indices de parcours de boucles */
    int i, j;

    /* Initialisation de l'image resultat, de mêmes largeur, hauteur
       et valeur max que l'image donnée en paramètre */
    init_image(l, h, max, im2);

    /* Parcours de chaque pixel de chaque ligne */
    for (i = 0; i < l; i++) {
        for (j = 0; j < h; j++) {
            p = lepixel(im1, i, j);
            if (p>=128){
                p_modifie = 255;
            }
            else
            {
                p_modifie = 0;
            }
            affecter_pixel(im2, i, j, p_modifie);
        }
    }
}

```

```
}  
}
```