Td1 : Visualisation d'images sous X-window

Découvrir l'utilisation des programmes

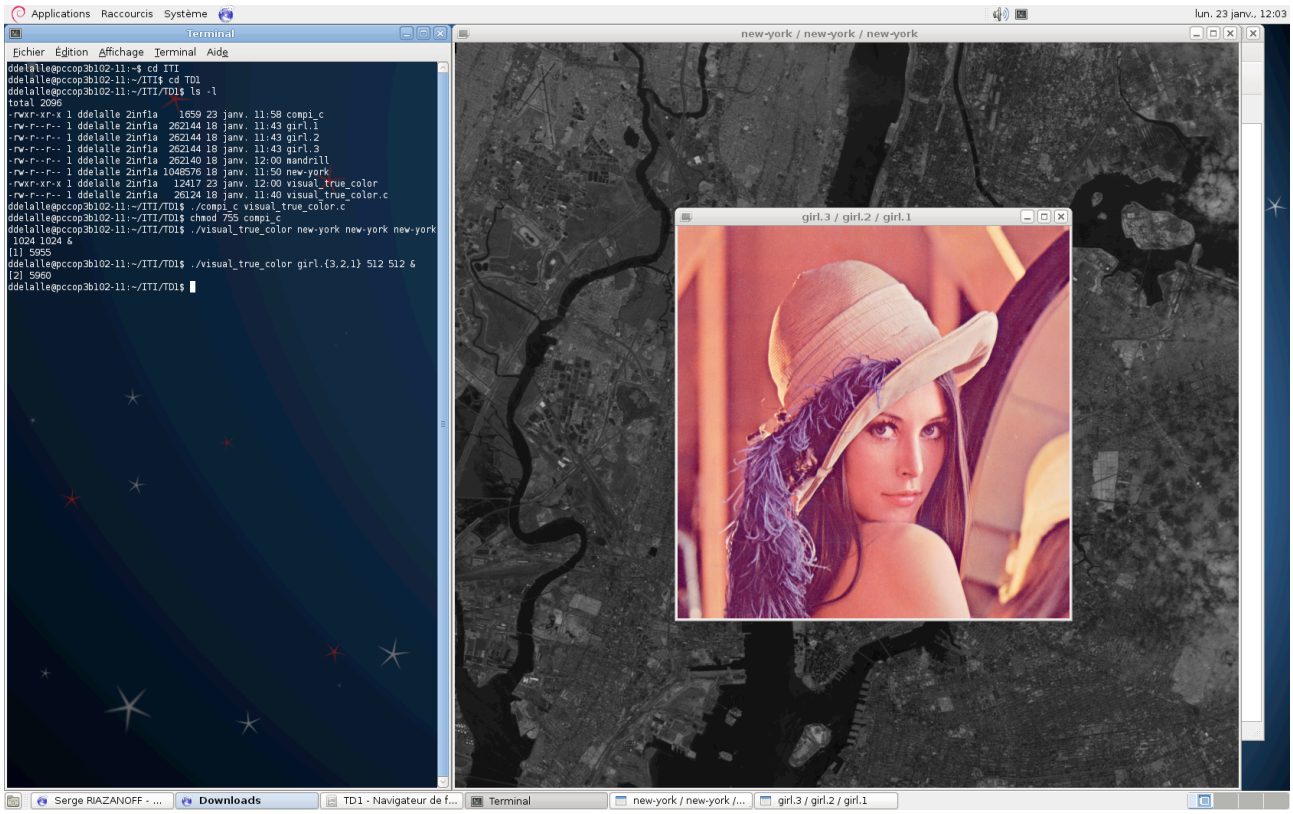
1. Léna sur Central Park.

Après avoir télécharger les fichiers et les images correspondant à Léna et Central Park, commençons tout d'abord par afficher Léna (512*512).

Pour ce faire, il faut écrire dans le terminal :
./compi_c visual_true_color.c
chmod 755 compi_c
./visual_true_color girl.{3,2,1} 512 512 &

Affichons maintenant l'image de Central Park (1024*1024).
./visual_true_color new-york new-york new-york 1024 1024 &

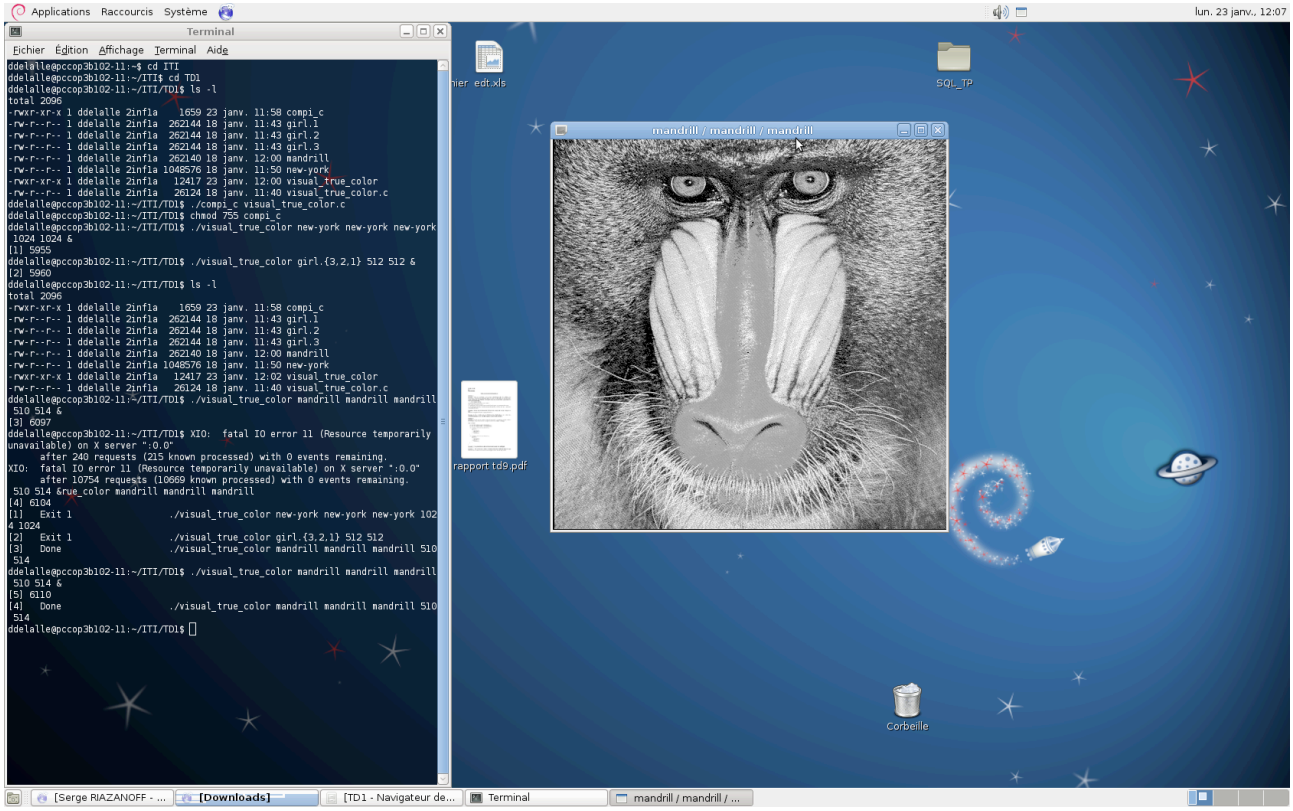
On obtient alors l'image suivante :



2. Gare au Mandrill

Ne connaissant pas la taille de l'image il faut déterminer celle-ci.
Une image étant, en général, carrée, son nombre de lignes = nombres de colonnes = $\sqrt{N}$ ou N correspond au nombre de pixels de l'image.
Ici N = 262140 donc $\sqrt{N} \approx 512$
Or si l'on affiche cette image en 512*512 celle-ci est décalée (l'image est en diagonale)
en suivant les diagonales, on constate qu'en partant du coin supérieur haut de la fenêtre, on arrive au 2/3 de la bordure droite. Cela veut donc dire qu'on a un décalage de 2 pixels.
Au final on trouve que l'image doit être de la taille suivante : 510 * 514

On l'affiche donc avec la commande suivante :
./visual_true_color mandrill mandrill mandrill 510 514 &



Examiner les fichiers
3. Couleur du pixel (0,0)

on va utiliser les commandes suivantes :
man od : permet d'afficher le manuel (documentation) de la commande od.
od -x girl.1| more => 7d
od -x girl.2| more => 89
od -x girl.3| more => e2

En écriture hexadécimal, nous obtenons :
R = e2 = 226 B = 7d = 125 V = 89

Configuration du serveur X-Window
4. Mapping des couleurs
Dimensions de l'écran : 1680 lignes * 1050 colonnes
Visual utilisé par défaut : 0x21 class : TrueColor
profondeur d'affichage : 8 bytes

Masque d'affichages des composants :
Rouge : 0xff0000 : 11111111 00000000 00000000
Vert : 0xff00 : 00000000 11111111 00000000
Bleu: 0xff : 00000000 00000000 11111111

Nombre de bits utilisés et quel est le décalage à gauche :
Rouge : 24 bits – 16 bits

Vert : 24 bits – 8 bits
Bleu : 24 bits – 0 bit

Modifier le programme visual_true_color.c
5.Couleur et représentation interne du pixel (125,17)
// a priori il faut recuperer les valeurs de la ligne 64017 (125*512 + 17)

6. Comparaison des programmes « visual_true_color » et « skelet »
a.
Le programme skelet possède deux appels différents.
Le premier correspond à une image couleur et le deuxième correspond à une image en noir et blanc.
Avec le second, il n'y a donc qu'un seul nom d'image à donner en paramètre.
Par ex pour afficher l'image du mandrill, au lieu d'écrire :
./visual_true_color mandrill mandrill mandrill 510 514 &
comme on le faisait précédemment, on va maintenant écrire :
./skelet mandrill 510 514 &
en fonction du nombre de paramètres, le programme skelet en déduit qu'il s'agit d'une image couleur ou noir et blanc et la traite en fonction.
Cependant, pour une image couleur rien ne change, on doit spécifier le nom des trois images correspondant chacune à un filtre de couleur.

b.
Dans visual_true_color, les frame buffer s'appellent :
- datimage
- buf_red
- buf_green
- buf_blue

et dans skelet, ils s'appellent :
- origin_frame_buffer
- processed_frame_buffer

c.
Les tâches d'initialisation de frame buffer ont été reportées dans une méthode car avec ce procédé on peut traiter à la fois le fait que l'image soit en couleur ou en noir et blanc.
En effet, dans les deux cas, des tableaux semblables (mais avec des contenus différents) ont été créés afin de contenir les valeurs des pixels. Afin d'éviter la duplication de code on fait appelle à une méthode pour initialiser les frame buffer.

d.
// je n'ai pas pu tester cette partie car je travaille sous mac à mon domicile.
a priori il faut changer MAX_COLOR 0
et dans la ligne en dessous :

```
#define nint(float_value)  (((float_value)-(int)(float_value) > 0.5)?         \
                        (int)(float_value)-1 : (int)(float_value))
```

et peut etre aussi la partie :

```
/* Analyze the way Red, Green and Blue component are mapped in frame buffer
en remplacant les +1 par des -1... a voir
```