

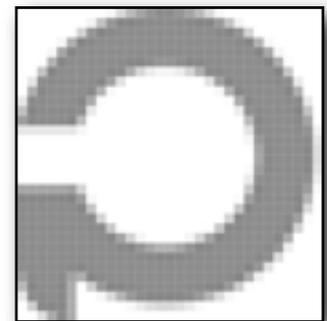
# Sélection et Déroulage d'une image



## Préambule

### Stockage informatique des images

La représentation la plus simple et la plus utilisée pour stocker des images en informatique est ce qu'on appelle la carte de pixels (en anglais  *pixmap / pixelmap*  mais bien souvent on utilise  *bitmap*  par abus de langage). Un pixel (abréviation de  *picture element* ) représente la plus petite unité que l'on peut afficher sur un écran. Le principe d'une carte de pixels est d'associer une couleur à chacun de ces pixels. De plus, il n'est pas nécessaire de stocker les coordonnées de chaque pixel puisqu'elles sont facilement accessibles lorsque l'on connaît les dimensions de l'image. Ainsi si notre image possède une largeur  $L$  et une hauteur  $H$  les couleurs de chacun des pixels seront stockés dans un tableau à une dimension (= un vecteur) où les  $L$  premières valeurs représenteront les couleurs de la première ligne de pixels, les  $L$  valeurs suivantes représenteront les couleurs de la deuxième ligne...



*Zoom sur le logo de l'école.  
On distingue les pixels avec  
une couleur unique par pixel*

### Couleur d'un pixel

Le codage d'une couleur est stocké sur un ou plusieurs octets. Le format que nous utiliserons pour ce projet est le format **PGM** (*Portable GrayMap*) qui est une image noir & blanc où l'intensité de blanc est codée sur 1 octets (0 = pixel "noir", 255 = "pixel blanc"). Dans la suite du sujet et par abus de langage, nous utiliserons "couleur" pour désigner l'intensité de blanc d'un pixel.

### En résumé

De manière générale, un fichier image contient donc une entête qui regroupe les informations essentielles de l'image (format, largeur, hauteur) et un bloc de données qui regroupe les couleurs des pixels de l'image.

## Description du Projet

Dans la plupart des logiciels de retouches d'images existent des outils qui permettent de sélectionner une zone de l'image sans avoir à sélectionner les pixels un par un. Les plus courants sont:

- L'outil *rectangle de sélection*: on spécifie le bord inférieur gauche et le bord supérieur droit et tous les pixels appartenant à ce rectangle sont sélectionnés.

- L'outil *ellipse de sélection*: sont sélectionnés tous les pixels vérifiant l'équation ci-dessous. C, de coordonnées  $(x_c, y_c)$ , représente le centre de l'ellipse et **a** et **b** représentent respectivement le grand rayon et le petit rayon.

$$\frac{(x - x_c)^2}{a^2} + \frac{(y - y_c)^2}{b^2} \leq 1$$

- L'outil *baguette magique*: il permet de facilement sélectionner une zone colorée uniforme sans être obligé de tracer le contour. Cet outil fonctionne de la manière suivante: à partir d'un pixel sélectionné P de couleur c, les pixels voisins de P qui sont de la même couleur vont être ajoutés à la sélection, ainsi que les voisins des voisins... On obtient au final une zone de l'image pour laquelle tous les pixels ont la même couleur. De plus, il est possible d'intégrer une tolérance T (exprimée en pourcentage) pour ajouter un pixel si sa couleur est comprise entre  $[c \times (1 - T); c \times (1 + T)]$ , ce qui permet d'avoir un détournage plus permissif et de prendre en compte la mauvaise qualité d'une image.

---

## Cahier des Charges

### Réalisation

L'objectif de ce projet est de réaliser un programme qui effectue quelques opérations élémentaires de retouche d'images. Ce programme, qui fonctionne en ligne de commandes, doit permettre à l'utilisateur de:

1. charger une image (par souci de simplicité, toutes les images sont de taille **800 x 600**),
2. d'appliquer différents outils de sélection.
3. d'étendre ou de réduire une sélection en utilisant un autre outil (où le même outil mais avec d'autres caractéristiques).
4. quand l'utilisateur est satisfait de la sélection, remplacer la couleur des pixels sélectionnés par celle d'une autre image.
5. sauvegarder le résultat de ces transformations.

---

### Travail demandé

Votre travail est constitué de deux étapes:

1. Analyser le problème afin de proposer une méthode de résolution pour chacune des fonctionnalités demandées.
2. Effectuer la conception de cette méthode et son implantation en C.

### Travail à rendre

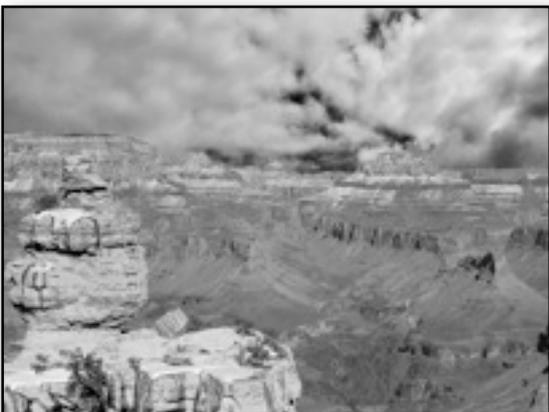
Vous devrez remettre par mail (à [jeremie.dequidt@polytech-lille.fr](mailto:jeremie.dequidt@polytech-lille.fr) et [laure.gonnord@polytech-lille.fr](mailto:laure.gonnord@polytech-lille.fr)) une archive **tgz** au plus tard le **16 janvier 2011 avant minuit** (un malus de 5 points sera appliqué par jour de retard):

1. le code source commenté et correctement indenté de votre programme C (voir les consignes sur le site de Laure Gonnord)

2. un rapport structuré au format **PDF** (police 11points, interligne simple) dans lequel apparaissent les points suivants:
- introduction et conclusion,
  - cahier des charges,
  - analyse du problème (méthodes de résolution et éventuels choix effectués),
  - description des structures utilisées pour la résolution,
  - description des principaux algorithmes (pseudo-code ou arbres programmatiques),
  - avancement du projet et problèmes rencontrés.

**Note:** Il est inutile d'imprimer le rapport et il est inutile de s'inspirer du projet des IMA3 de l'année dernière.

### Illustration



*Exemple d'utilisation: à partir de l'image en haut à gauche, l'outil baguette magique a été appliqué sur un pixel du ciel. Les pixels issus de la sélection ont été remplacés par ceux de l'image en haut à droite. Le résultat est l'image de gauche.*

### Code Source et Données

Une archive vous est fournie pour commencer le projet. Elle se trouve à l'adresse <http://dequidt.plil.net/docs/ProgStructuree/> et contient un squelette de programme C. Ce programme vous est utile car il contient des fonctions qui permettent de charger et de sauvegarder des images dans des fichiers. Dans l'archive mentionnée se trouve deux images (PGM) que vous pouvez utiliser pour le projet. Cependant n'hésitez pas à utiliser d'autres images que vous transformerez au bon format en utilisant les logiciels **display** ou **gimp**.