

TP1 - Hiver 2018

IMN 259

Analyse d'images

Date limite pour remettre votre travail : 2 février

Objectifs

1. Lecture/écriture des formats de fichier pgm et ppm.
2. Coder quelques opérations ponctuelles de base.

Description

À l'aide du code C++ fourni (fichiers tp1.cpp, MImage.h, MImage.cpp) vous devez implémenter la fonction nécessaire pour l'écriture des formats images "pgm" et "ppm" (la lecture vous est donnée). Tel que mentionné dans le cours, le format pgm est fait pour les images en niveaux de gris et ppm pour les images couleur. Votre code doit être en mesure de gérer les fichiers ASCII et binaires (formats P2, P3, P5 et P6). Pour ce faire, vous devez utiliser les fonctions *LoadImage* et *SaveImage* de la classe *MImage*. Pour plus d'information sur ces formats d'image, tapez *pgm ppm file format* dans Google. Voir en annexe une explication de la structure de stockage.

Vous devez aussi implémenter cinq opérations ponctuelles simples. Étant donnée une image I de taille $N \times M$, ces opérations sont :

1. Inversion de la couleur des pixels : $I'(i, j, k) = 255 - I(i, j, k)$ pour tous les pixels (i, j) . À noter que la variable k indique la bande de couleur ($k=0$:rouge, $k=1$:vert ou $k=2$:bleu) (fonction *Invert*).
2. Appliquer un seuil à chaque pixel (fonction *Threshold*)

$$I'(i, j) = \begin{cases} 0 & \text{si } (I(i, j, 0) + I(i, j, 1) + I(i, j, 2))/3 < \text{Seuil} \\ 255 & \text{sinon.} \end{cases} \quad (1)$$

pour les images couleur, et

$$I'(i, j, 0) = \begin{cases} 0 & \text{si } I(i, j, 0) < \text{Seuil} \\ 255 & \text{sinon.} \end{cases} \quad (2)$$

pour les images en niveaux de gris.

3. Calcul de l'intensité moyenne de chaque bande de couleur k (R, G et B pour les images couleur) :

$$\text{moy}_k = \frac{1}{NM} \sum_i^N \sum_j^M I(i, j, k) \quad (3)$$

Utilisez la fonction *Average*.

4. Calcul du contraste global de chaque bande de couleur :

$$\text{cont}_k = \sqrt{\frac{1}{NM} \sum_i^N \sum_j^M (I(i, j, k) - \text{moy}_k)^2} \quad (4)$$

où moy_k est l'intensité moyenne de la bande k de l'image I (fonction *Average*). Pour ce faire, utilisez la fonction *Contrast*.

5. Recalage des niveaux de gris de l'image (entre 0 et 255) (fonction Rescale).

$$I'(i, j, k) = 255 * \frac{(I(i, j, k) - \min_k)}{\max_k - \min_k} \quad (5)$$

où \min_k et \max_k sont les niveaux de gris minimum et maximum de la bande k de l'image. Cette fonction fait en sorte que l'intensité maximale de chaque bande de couleur de l'image résultante est 255 et l'intensité minimale est 0.

6. **SUGGESTION** : commencer par gérer les formats ASCII, ensuite implémentez les cinq opérations ponctuelles et finissez par les formats binaires.

Évaluation

Ce travail doit être fait en **équipe de DEUX**. Pour simplifier la correction, nous vous demandons de modifier le moins possible la fonction **main** et de remettre TOUS les fichiers .h, .cpp, .pgm, .ppm ainsi que le Makefile. Veuillez noter que le barème d'évaluation est disponible sur le site web du cours. Au moment de soumettre votre travail, assurez-vous que votre code compile bien sous Linux (vous n'avez qu'à taper la commande *make* dans un terminal Linux). Utilisez le système **turnin web** pour soumettre votre travail (<http://opus.dinf.usherbrooke.ca:8080/>).

Annexe

Dans le cours d'introduction à la programmation, vous avez probablement vu comment créer un tableau 2D en C++. La méthode suggérée est de créer un tableau de pointeurs, où chaque élément pointe vers un autre tableau. Bien que cette méthode soit simple et efficace, elle n'est pas idéale pour stocker une image. En pratique, les images sont stockées sous forme de tableau **1-D** où chaque ligne se suit. Ainsi, toute l'image est dans une seule zone contiguë en mémoire, ce qui fait un meilleur usage de la cache du processeur, et ainsi accélère significativement les traitements.

Dans le code fourni, une image 4×3 comme suit :

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix}$$

sera stockée en mémoire sous forme de tableau 1-D de taille 12, comme suit :

$$[a \ b \ c \ d \ e \ f \ g \ h \ i \ \dots]$$

Pour trouver l'élément h à $(x, y) = (3, 1)$, il faut calculer l'indice comme suit :

$$\text{indice} = y \times \text{largeur} + x = 1 \times 4 + 3 = 7$$

On confirme que l'élément h se trouve bien à l'indice 7 dans le tableau 1-D.