

# Miniprojet Algo avancé

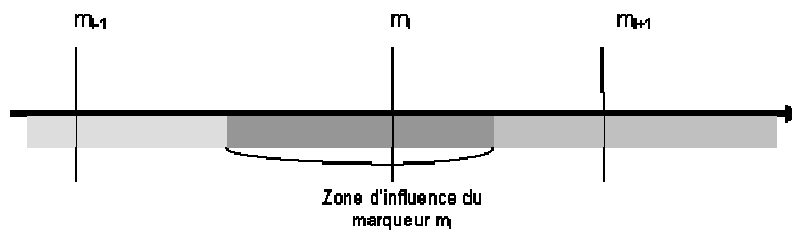
Réduction des couleurs d'une image  
Optimisation par la programmation dynamique  
(à effectuer en binôme)

**But :** A partir d'une image en 256 niveaux (gris,RVB) nous cherchons à diminuer le nombre de couleurs vers une quantité bien inférieure (8,16...). Le but est d'effectuer cette simplification au mieux c'est à dire en préservant la qualité de l'image de départ.

**Notation :** Nous notons  $h(i)$ , la fonction qui à tout niveau  $i$  associe le nombre de pixels ayant ce niveau. L'histogramme d'une image est le diagramme en bâtons qui représente cette fonction. Notons  $g(i)$  la fonction qui à un niveau  $i$  dans  $[0,255]$  fait correspondre le nouveau niveau dans l'image simplifiée. Nous choisissons comme expression arithmétique de l'erreur la fonction :

$$erreur = \sum_{i=0}^{255} h(i) \times |g(i) - i|$$

**La méthode :** La quantité de niveaux :  $n$  dans l'image simplifiée est choisie par l'utilisateur. Nous allons procéder de la manière suivante. Notre méthode de simplification consiste à positionner  $n$  marqueurs sur les niveaux 0 à 255. La zone d'influence d'un marqueur correspond à tous les niveaux qui lui sont les plus proches. Ainsi la zone d'influence d'un marqueur  $m_i$  est définie de la manière suivante :  $[(m_{i-1} + m_i) / 2, (m_i + m_{i+1}) / 2]$ . Sur cette zone, la valeur de la fonction  $g$  est donnée par celle du marqueur. Nous obtenons donc :



**Programmation dynamique** Etudiez et concevez une méthode positionnant les  $n$  marqueurs afin de minimiser l'erreur. Cette résolution de l'optimalité doit être effectuée par une méthode de programmation dynamique.

**Développement** Programmez votre méthode en tenant compte des initialisations. Modifiez là afin de pouvoir retrouver l'ensemble des marqueurs sélectionnés pour atteindre l'optimum. Vous pouvez maintenant appliquer votre méthode sur une image en niveaux de gris :



Image originale en 256 niveaux



Simplification en 16 niveaux (Round)



Simplification en 16 niveaux (Prog. Dyn.)

La fonction basique de simplification choisie est  $\text{int}(\text{int}(\text{niveau}/(256.0/n)) * (256.0/n))$

**Qualité de la méthode** En quoi cette méthode peut être considérée comme meilleure ? Pourquoi pour la plupart des images, la qualité de l'image produite par la fonction basique et la qualité de l'image issue de notre simplification seront assez proches ? Quel défaut flagrant pénalise la méthode basique ? Pourquoi n'est-il pas présent dans notre méthode ? Cette méthode teste-t-elle toutes les fonctions  $g(i)$  possibles ?

**Complexité** Démontrez que l'algorithme obtenu est en  $O(n^3)$ . Expliquez comment réduire cette complexité à  $O(n^2)$  par un pré calcul.

**Vérification** Testez sur une image en niveaux de gris en prenant trois marqueurs. Le résultat sera vérifié en programmant une fonction de type brute force qui testera toutes les positions possibles des trois marqueurs.

**La couleur** Nous décomposons une image couleur en trois images en niveaux de gris (rouge, vert et bleu). Traitez indépendamment chaque couche et reconstruisez le résultat final :



Image originale 24 bits RVB



Simplification à 6 bits (Round)



Simplification à 6 bits (Prog. Dyn.)

**Rendu :** Vous rendre un fichier nom1-nom2.rar à l'adresse [bond@polytech.unice.fr](mailto:bond@polytech.unice.fr) qui contiendra

- un dossier expliquant votre démarche et qui contient une analyse de la complexité
- les sources de votre programme ainsi que des solutions pour les exemples à traiter (ils seront disponibles à partir du 15 mars dans le dossier <http://www-local.essi.fr/~bond/si4/Calc/Miniprojet>).

**Date limite :** Lundi 22 avril 2013, 9 heures.