

Cours de programmation séquentielle

TP noté de programmation séquentielle - Arbres quaternaires

1 Buts

- Mise en place d'une structure d'arbre quaternaire.
- Traitement d'images: transformation par symétrie, compression.
- Manipulation de fichiers.

2 Énoncé

Réaliser un programme qui effectue la lecture d'une image (définie en 256 niveaux de gris) stockée dans un fichier de type PGM (*Portable Gray Map*). Créer un arbre quaternaire permettant de la mémoriser. Prévoir ensuite les fonctionnalités suivantes:

- Une symétrie : proposer à l'utilisateur une symétrie verticale, horizontale ou centrale, puis réécrire l'image transformée.
- Une compression : après avoir demandé à l'utilisateur un seuil pour la variance ou l'écart-type entre les pixels réécrire l'image compressée.

Les images seront sauvegardées sur le disque en format PGM et visualisées par exemple avec la commande `display` (`eog` fait également l'affaire).

3 Marche à suivre

- Mettre en place une structure d'arbre quaternaire afin de stocker une image en mémoire sans perte d'information.
- Effectuer la lecture de l'image stockée dans un fichier PGM. Elle est de dimension 512×512 pixels et définie en 256 niveaux de gris. Stocker les pixels de celle-ci dans une matrice.
- Remplir l'arbre quaternaire à partir de la matrice. Prévoir la transformation inverse pour passer de l'arbre vers la matrice.
- Implémenter les transformations nécessaires sur l'arbre pour effectuer une symétrie verticale, horizontale ou centrale.
- Implémenter une compression de cet arbre basée sur l'algorithme suivant:
 1. On divise l'image en 4.
 2. Sur chacune des parties on calcule la moyenne et l'écart-type des valeurs des pixels.
 3. Si l'écart-type des valeurs est plus grande qu'une valeur entrée par l'utilisateur, on redivise le quadrant en 4.

La variance $\text{var}(X)$ ou l'écart-type $\text{std}(X)$ pour un ensemble de n pixels $X = (X_1, \dots, X_n)$ sont donnés par

$$\text{var}(X) = \frac{1}{n} \sum_{i=1}^n X_i^2 - \left(\frac{1}{n} \sum_{i=1}^n X_i \right)^2, \quad \text{std}(X) = \sqrt{\text{var}(X)}. \quad (1)$$

- Lors de l'écriture de l'image compressée, chaque feuille de l'arbre doit uniquement stocker la moyenne des pixels contenus dans le quadrant de l'image.
- Le résultat des transformations sera écrit dans un fichier PGM pour être visualisé.

4 Travail à rendre

- Ce travail est à réaliser individuellement.
- Vous devez rendre votre code sur votre répo git au plus tard le 25.05.2021 à 23h59 (la version antérieure ou égale à cette date sera récupérée sur votre git).
- Tout le monde n'ayant pas le même nom de repo git, veuillez écrire l'adresse de votre git, ainsi que votre nom sur le wiki sur **Cyberlearn**. N'oubliez pas de laisser votre répo public pour qu'on puisse le cloner.
- Suite à ce rendu, vous devrez effectuer une présentation orale de votre travail d'une durée d'au maximum 20 min avec des diapositives.
- La présentation orale aura lieu les mercredi 26 mai et 2 juin pendant les heures de cours.
- Vous serez notés sur la présentation et sur le code que vous avez rendu (pondération 1/4 - 3/4).