

UNIVERSITÉ DE SHERBROOKE  
DÉPARTEMENT D'INFORMATIQUE

IFT 159

Devoir #4 — Automne 2009

**Analyse, conception et programmation**

**Attention :** Cette spécification sert premièrement à faire l'analyse/conception et ENSUITE la programmation de VOTRE analyse/conception.

Le but de ce travail pratique est de faire l'analyse d'un problème de difficulté moyenne, de faire la conception de sa solution en utilisant un diagramme structurel, puis de programmer cette solution. Cette programmation illustrera, en plus des notions de séquence, sélection et itération, le concept de tableau (donc pas de «string»), la récursivité et les paramètres de sortie. Le but est aussi de créer un programme **correct** et, optionnellement, **robuste**. Un programme robuste se verra attribué un bonus de 20%.

Ce travail pratique se divise en deux phases :

1. La phase 1 consiste à faire l'analyse et la conception complète du problème et des sous-problèmes identifiés. **Elle est à remettre au plus tard le mardi 3 novembre 2009 dans les casiers au sous-sol du D4.**
2. La phase 2 consiste à programmer votre solution élaborée lors de la phase 1. **Elle est à remettre au plus tard le mardi 10 novembre 2009.** La remise devra être faite grâce à la commande *turnin*.

---

**Sujet : Comparaison d'images**

On veut bâtir un programme général qui, en plus de calculer le bruit dans une image de télédétection et de les comparer, affiche une ou plusieurs listes d'images avec leurs caractéristiques respectives selon un ordre établi à partir de différents critères.

Le nombre maximum d'images pouvant être traité est de 10 (pour faciliter le jeu de données et la correction). Le nombre courant d'images n'est pas connu. Chaque image est connue via son identificateur de fichier qui est une chaîne de caractères composée d'au plus 20 caractères (à valider pour la robustesse). La dimension de chaque image est de 200x200 pixels (ne pas valider). Chaque pixel est représenté par un entier sur 8 bits (caractère). Vous devez calculer le bruit de chacune des images («moyenne», «variance», «skewness» et «kurtosis»). De plus, lorsque cela est requis, vous devez comparer les images.

Les critères permettant d'établir l'ordre de tri pour l'affichage des images sont :

- La «moyenne», la «variance», le «skewness», le «kurtosis» ou la «ressemblance». Pour la ressemblance, vous devez d'abord obtenir l'image de référence à laquelle toutes les autres seront comparées.
- L'ordre croissant ou décroissant des valeurs du critère précédent. Si deux valeurs sont identiques, elles figureront dans l'ordre de leur apparition.

Les critères, sur le champ et l'ordre, devront **obligatoirement** être (à valider pour robustesse) :

«moyenne»	pour trier suivant le moyenne
«variance»	pour trier suivant la variance
«skewness»	pour trier suivant le skewness
«kurtosis»	pour trier suivant le kurtosis
«ressemblance»	pour trier suivant le niveau de ressemblance
«fin»	pour marquer la fin du traitement
«c» ou «C»	pour trier suivant l'ordre croissant
«d» ou «D»	pour trier suivant l'ordre décroissant

Le programme lit premièrement les noms des fichiers contenant les images. Le programme lit ensuite les images et calcule le bruit de chacune des images. Un nom de fichier spécial («fin») indique la fin de la liste d'images. Ensuite, pour chaque liste que l'on veut produire, le programme lit le critère avec lequel on veut trier les données et l'ordre dans lequel les données doivent être triées. Le critère «fin» indique la fin du traitement.

Pour les fins de robustesse, vous devez valider chacune des entrées et la relire si elle est invalide. Afin de faciliter la validation des données, toutes les chaînes de caractères (nom de fichier et critère) terminent par un «.».

Le tri **devra être un tri sélection récursif**. Dans un tri sélection, on sélectionne la plus petite valeur (ou la plus grande), puis on l'échange avec la première (ou la dernière) valeur [Pour le faire récursif, il est souhaitable de toujours l'échanger avec la dernière, dans l'état actuel de nos connaissances]. Cette première (dernière) valeur se trouve donc dans sa position définitive. On répète ce procédé sur le reste du tableau. [ Exemple : soit la suite 4-8-2-6. La plus grande valeur est 8. On l'échange avec la dernière valeur. La suite devient 4-6-2-8. Il reste à mettre en ordre la suite 4-6-2.]

### **Idée :**

Afin de ne pas modifier l'ordre des données dans les différents vecteurs, prendre un vecteur de travail dans lequel on inscrit les indices, et les permutations se font sur ce vecteur de travail.

Exemple :

- soit un vecteur de 5 valeurs qui contient : 12 - 4 - 8 - 15 - 1
- le vecteur de travail va contenir au début : 0 - 1 - 2 - 3 - 4
- une fois le tri en ordre croissant effectué  
le vecteur de travail va contenir : 4 - 1 - 2 - 0 - 3  
et le vecteur de valeurs sera inchangé.

## Notes importantes

- On rappelle ici que l'on ne doit utiliser que des tableaux à une ou deux dimensions (vecteur et matrice), et que l'on **ne doit pas utiliser les fonctions de la bibliothèque <string>**. L'utilisation de la bibliothèque <string> sera pénalisée.
  - **On demande, optionnellement, de bâtir un programme robuste. Un bonus sera accordé pour la robustesse.**
  - La correction est basée, entre autres, sur le respect de votre analyse préliminaire. Si, lors de votre programmation, vous devez modifier votre analyse et conception préliminaire, **mentionnez-le**.
- 

## Exemple de jeu de données

```
image1.  
image2.  
image3.  
image4.  
image5.  
image6.  
image7.  
fin.  
variance.  
C  
kurtosis.  
D  
ressemblance.  
c  
image3.  
fin.
```

---

## Remise de la partie analyse

Remettre :

- sur des feuilles séparées mais brochées ensemble et clairement identifiées (noms, matricules **et codes d'usager**), votre analyse globale du problème, votre diagramme structurel, les analyses et les algorithmes du problème et des sous-problèmes identifiés.
- la feuille de calcul d'effort (que vous pouvez récupérer sur le site WEB dans la section

documents).

---

## **Soumission du programme**

Vous devez me soumettre avant la date limite, à l'aide de la commande

**turnin -cift159 -ptp4 tp4**

un seul **répertoire** portant OBLIGATOIREMENT le nom **tp4** qui contient le programme demandé (**tp4\*.cpp**), le fichier **modifications.txt**, la documentation, trois fichiers tests (**test1**, **test2**, **test3**) avec leurs images et le fichier d'estimation d'effort . Étant donné que la correction est partiellement automatisée, une mauvaise soumission entraîne une mauvaise note (possiblement 0). Il n'y a pas de recorection.

Votre programme devra respecter les normes de programmation du département. Les fichiers tests devront contenir des tests significatifs. La documentation se fera sous forme de commentaires dans le programme en utilisant les normes de *doxygen*. Le fichier contenant le calcul de l'effort doit **OBLIGATOIREMENT** porter le nom **calculEffort.txt**.

Si nécessaire, vous pouvez utiliser plusieurs fichiers : les noms suggérés sont «tp4\_n.cpp» où n varie de 1 au nombre nécessaire. Tous les fichiers doivent se retrouver dans le répertoire **tp4**.

Pour la bonne utilisation du train de travail ultérieur à votre remise utiliser «endl» plutôt que «\n» pour changer de ligne.

Pour vous éviter de mauvaises surprises, faites un test d'essai dans les mêmes conditions que le train de travaux. Vous devez faire imprimer votre fichier source (ou vos fichiers sources) et exécuter votre programme comme suit :

```
./a.out < donnees > resultats
```

où «donnees» et votre fichier de donnees et «resultats» votre fichier de réponses. Faites imprimer ce dernier. Évidemment «a.out» est le fichier qui contient votre programme exécutable.

Pour vérifier si le programme a bien été soumis, vous pouvez faire la commande :

**turnin -v -cift159 -ptp4**

# La télédétection

La télédétection est la discipline scientifique qui regroupe l'ensemble des connaissances et des techniques utilisées pour l'observation, l'analyse, l'interprétation et la gestion de l'environnement à partir de mesures et d'images obtenues à l'aide de plates-formes aéroportées, spatiales, terrestres ou maritimes. Comme son nom l'indique, elle suppose l'acquisition d'information (images), sur un objet par un capteur situé à distance de l'objet.

Dans un monde idéal, les images de télédétection prises par les capteurs satellites devraient être parfaites. Cependant, dans la réalité, les images sont affectées par les erreurs provenant des instruments de mesure (capteur, satellite, station de réception, etc.), des conditions d'observation (angle de visée, paramètres orbitaux, conditions atmosphériques, etc.) et des erreurs de manipulation humaine. Toutes ces perturbations impliquent des erreurs aléatoires (ou accidentelles), grossières ou systématiques. Ces erreurs n'apportent pas d'information utile. Ce sont des artéfacts liés au processus d'acquisition et qualifiés de *bruit*. De façon générale, le *bruit* pour un spécialiste de traitement de signal ou d'image, est une constituante de la donnée observée qui ne permet pas de caractériser l'information recherchée.

En fait, le bruit perturbe l'analyse. Il est donc intéressant d'étudier et de caractériser ce bruit intrinsèque afin de l'extraire. Le bruit des images se caractérise par des paramètres mathématiques que l'on appelle «moments statistiques». Dans le cas général, les 4 premiers moments statistiques suffisent à caractériser le bruit.

Les 4 premiers moments statistiques sont les suivants :

– *la moyenne*

$$\bar{x} = \frac{1}{N} \sum_{j=1}^N x_j \quad (1)$$

– *la variance*

$$Var(x_1...x_N) = \frac{1}{N} \sum_{j=1}^N (x_j - \bar{x})^2 \quad (2)$$

– *le skewness*

$$Skew(x_1...x_N) = \frac{1}{N} \sum_{j=1}^N \left[ \frac{x_j - \bar{x}}{\sigma(x_1...x_N)} \right]^3 \quad (3)$$

$$\sigma(x_1...x_N) = \sqrt{Var(x_1...x_N)} \quad (4)$$

– le kurtosis

$$Kurt(x_1 \dots x_N) = \left\{ \frac{1}{N} \sum_{j=1}^N \left[ \frac{x_j - \bar{x}}{\sigma(x_1 \dots x_N)} \right]^4 \right\} - 3 \quad (5)$$

(N correspond au nombre de pixels,  $x_j$  est la valeur du pixel  $j$ )

Par exemple, sur l'image de la figure 1, nous avons simulé une image d'un capteur satellite avec 4 bruits différents sur 4 zones. Les résultats de l'analyse statistiques du bruit sur les 4 zones de l'image sont donnés dans le tableau suivant :

Zone	Moyenne	Variance	Skewness	Kurtosis
1	49.96	101.05	-0.004	3.04
2	100.37	402.26	0.004	3.05
3	50.14	905.37	0.0	2.90
4	198.73	1400.80	-0.38	2.69

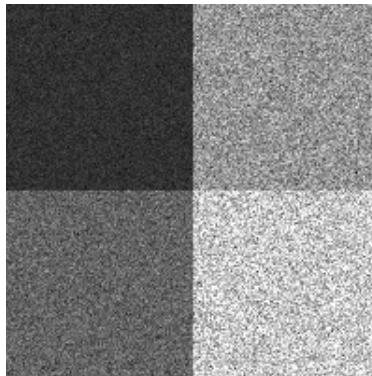


FIG. 1 – Cette image est composée de 4 bruits différents sur 4 zones de dimension 100x100 pixels, la dimension totale de l'image est donc de 200x200 pixels

Un autre élément important en télédétection consiste à déterminer si deux images sont semblables. Cette comparaison se fait grâce au coefficient de corrélation linéaire ( $r$ ). Ce coefficient est une mesure de probabilité (donc comprise entre 0 et 1) qui permet de calculer une possible corrélation entre 2 images. Cette corrélation correspond à un facteur de « ressemblance » visuellement parlant. La formule suivante permet de calculer la corrélation «  $r$  » entre deux images :

type	positive	négative
Corrélation parfaite	$0,98 < r \leq 1$	$-0,98 > r \geq -1$
Corrélation forte	$0,80 < r \leq 0,98$	$-0,80 > r \geq -0,98$
Corrélation moyenne	$0,60 < r \leq 0,80$	$-0,60 > r \geq -0,80$
Corrélation faible	$0,35 < r \leq 0,60$	$-0,35 > r \geq -0,60$
Corrélation nulle	$0,35 < r \leq 0$	$0 \geq r \geq -0,35$

TAB. 1 – Coefficient de corrélation et type de corrélation

$$r = \frac{\sum_j (x_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_j (x_j - \bar{x})^2} \sqrt{\sum_j (y_j - \bar{y})^2}} \quad (6)$$

Lorsque la corrélation est trouvée, on se sert du résultat afin de déterminer si les images sont semblables. La table 1 montre comment extraire cette information.

Par exemple, prenons les images du satellite Landsat montrées à la figure 2. L'image (a) et l'image (b) représentent la même zone d'étude avec 11 années de décalage. L'image (a) et l'image (c) représentent 2 zones d'étude différentes. Le coefficient de corrélation entre l'image (a) et l'image (b) est de 0.87, ce qui signifie que ces images ont une corrélation très forte (c'est très probablement la même image). Le coefficient de corrélation entre l'image (a) et l'image (c) est de 0.10, ce qui signifie que leur corrélation est nulle (ce ne sont pas les mêmes images).

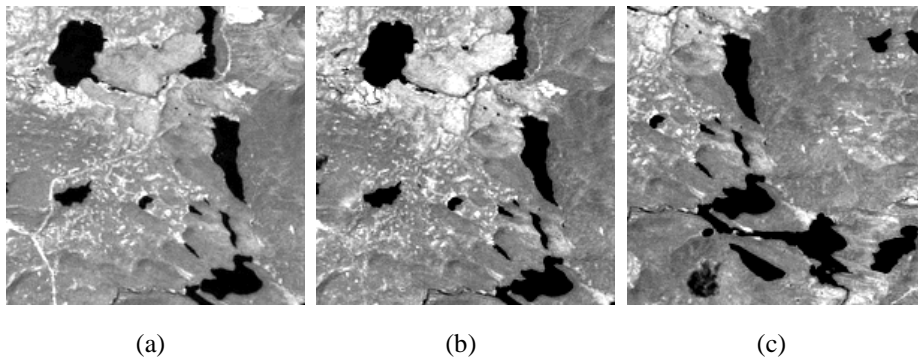


FIG. 2 – (a) image de Landsat TM5, année 1988 ; (b) image de Landsat TM5, année 1999 ; (c) image de Landsat TM5, année 1998. La dimension des images est de 200x200 pixels