Projet Programmation S6

Sujet n°1 : Comptage de cellules

Compte-rendu de première séance

L'objectif de ce projet est de compter les cellules présentes dans une image en niveaux de gris en les isolant par des méthodes de morphologies mathématiques.

Le projet se divise en plusieurs grandes étapes.

Nous débutons avec une image PGM en nuances de gris dont on souhaite compter le nombre de cellules.

Mais pour faciliter le traitement de l'image il est nécessaire de la seuiller et l'obtenir ainsi une image entièrement composée de noir et blanc (étape 1).

Ensuite, l'on souhaite compter seulement les cellules entièrement visibles sur l'image. Donc il faut supprimer les cellules situées au niveau des bords (étape 2).

Certaines cellules seuillées présentes des trous qu'il va falloir également supprimer pour un comptage optimal (étape 3).

On peut remarquer également que certaines cellules se touchent présentant un problème pour les compter. Cette étape constitue alors à éroder les celles pour bien les séparer (étape 4).

Enfin, la dernière étape consiste à compter les composants connexes de l'image (étape 5).

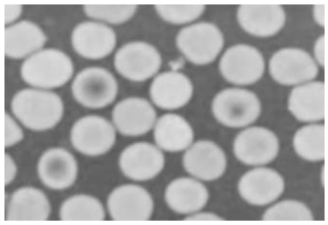

Image initiale

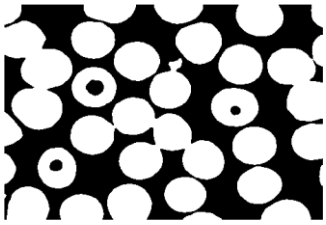

Image étape 1

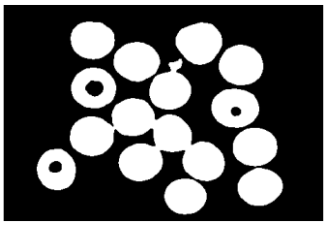

Image étape 2

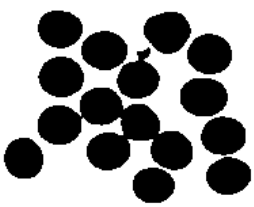

Image étape 3

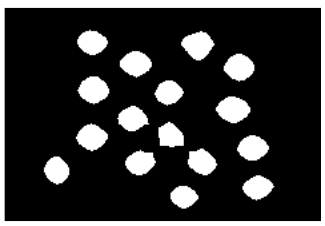

Image étape 4

(Lecture image PGM)

Objectif : lire une image au fomat PGM et la stocker en mémoire dans un tableau à deux dimensions

En entrée : image en format PGM
En sortie : tableau d'entiers à deux dimensions

*int** lecture_PGM(char* image)*

La fonction devra lire le fichier au format PGM dont on connaît la syntaxe et relevés les données le composant qui nous intéressent, à savoir la taille de l'image (nombre de lignes et nombre de colonnes) et la valeur de chaque pixel (entre 0 et 255). A partir de ces données, on les stocke alors dans un tableau à deux dimension (nombre de ligne x nombre de colonnes).

On pourra définir une structure rassemblant les informations utiles de l'image (taille et valeurs des pixels).

Etape 1 : Seuillage

Objectif : obtenir une image en noir et blanc à partir d'une image PGM (facilite le traitement de l'image car binaire)

Méthode manuelle : en entrée : image à seuiller à format PGM et le seuil
en sortie : image seuillée

char seuillage_manuel(pgm* image, int seuil)*

Méthode d'Otsu : en entrée : image à seuiller en format PGM
en sortie : image seuillée

char seuillage_Otsu(pgm* image)*

float histogramme(pgm* image)*
*int seuil_Otsu(pgm * image)*

Contrairement à la méthode manuelle, la méthode d'Otsu permet de déterminer automatiquement la valeur du seuil à partir de l'histogramme de l'image et renvoie en sortie l'image seuillée.

La fonction *histogramme* prend en entrée l'image et renvoie un tableau de float compris entre 0 et 1, correspondant à la distribution des valeurs des pixels de l'image (entre 0 et 255).

La fonction *seuil_Otsu* prend en entrée l'image et renvoie le seuil à partir de son histogramme. La valeur du seuil est la valeur pour laquelle la variance inter-classe sera maximale.

Remarque : *pgm* serait le type correspondant à l'image en format PGM.

(Opérations booléennes)

Objectif : implémenter les opérations booléennes “OR” et “XOR”, qui seront utilisées dans les fonctions suivantes.

- OR :
    - En entrée : 2 integer prenant les valeurs 0 ou 1 (noir ou blanc)
    - En sortie : 1 integer prenant la valeur
        - 1 si au moins l’une des deux entrées vaut 1;
        - 0 sinon
- XOR :
    - En entrée : 2 integer prenant les valeurs 0 ou 1 (noir ou blanc)
    - En sortie : 1 integer prenant la valeur
        - 1 si une entrée exactement vaut 1
        - 0 sinon

On pourra aussi prendre en entrée 2 tableaux de même dimension et renvoyer un tableau de même dimension que les tableaux d’entrée, et faire les opérations décrites plus haut pour chaque éléments des tableaux.

On passera les paramètres par valeur pour les conserver en mémoire

(Opérations morphologiques basiques)

- Erosion
- Dilatation
- Reconstruction

Etape 2 : Suppression des cellules au bord

Objectif : changer la valeur des éléments du tableau d’entrée de 1 (blanc) à 0 (noir) pour les cellules au bord en utilisant les fonctions reconstruction et XOR

- En entrée : la matrice représentant l’image dont on veut supprimer les cellules au bord
- En sortie : La matrice représentant l’image sans les cellules au bord

On passera les paramètres d’entrée par valeur pour les conserver en mémoire.

Etape 3 : Bouchage de trous

Objectif : changer la valeur des éléments du tableau d’entrée de 0 (noir) à 1 (blanc) pour les cellules contenant un “trou” en utilisant les fonctions reconstruction et inversion (change les pixels blancs en noirs et inversement)

- En entrée : la matrice représentant l’image dont on veut supprimer les trous
- En sortie : La matrice représentant l’image sans trous

On passera les paramètres d’entrée par valeur pour les conserver en mémoire.

Etape 4 : Erosion manuelle

Objectif : rendre les cellules plus petites pour éviter qu'elles ne se touchent et pouvoir les compter correctement, en utilisant la fonction érosion.

- En entrée : la matrice représentant l'image dont on veut diminuer les cellules de taille
- En sortie : La matrice représentant l'image avec les cellules diminuées

On pourra aussi passer en entrée le nombre d'érosion que l'on veut faire.

On passera les paramètres d'entrée par valeur pour les conserver en mémoire.

Etape 5 : Comptage des composantes connexes

Objectif : compter les cellules en utilisant un algorithme de recherche en profondeur dans les pixels d'une image

- En entrée : la matrice dont on veut compter les cellules
- En sortie : le nombre de cellules

On passera les paramètres d'entrée par valeur.

Répartition du travail :

Une personne s'occupera de toute la partie 1, l'autre des parties 2 et 3. Les parties 4 et 5 seront traitées ensemble après une mise en commun des travaux préliminaires.

Une première mise en commun sera réalisée la semaine du 13 Mars. On travaillera en parallèle les parties sur le format PGM et les opérations booléennes et morphologiques. Les parties 1, 2 et 3 devraient être finalisées pour la semaine du 27 Mars, la partie 4 pour le 3 Avril et la partie 5 pour le 17 Avril. La qualité et la performance du code seront probablement traitées de manière linéaire tout au long du projet. Selon notre avancée nous envisagerons peut-être de traiter la version avancée du projet. La séance 7 sera consacrée à la préparation de la soutenance.