LEFORT Vincent

Sélection et détection de zones dans des images paramétriques

Stage effectué du 29/03/04 au 05/06/04 au sein du Laboratoire de Microscopie Electronique Analytique de l'Université de Reims Champagne-Ardenne.

Rapport de stage de DUT Informatique
IUT de Reims-Châlons-Charleville

Remerciements

Je souhaite tout d'abord remercier chaleureusement, Mr le Professeur Gérard BALOSSIER pour m'avoir accueilli dans son laboratoire, pour avoir toujours su répondre à mes demandes et pour m'avoir aider à la rédaction de ce rapport.

J'exprime ma profonde reconnaissance à Mademoiselle Christine TERRYN, Assistante Associée Docteur en physique qui a accepté d'être mon maître de stage et qui a toujours été là pour répondre à mes diverses questions.

Je remercie également Mr Jean MICHEL, Maître de conférences, d'avoir gracieusement partagé son bureau avec moi pendant ces dix semaines, ainsi que sa connexion Internet. De plus il a efficacement répondu à mes questions.

Je remercie également Mr Francis DELIGNY, ingénieur d'études dans le domaine de l'électronique, informatique, et des réseaux pour avoir répondu à mes questions concernant le système informatique présent au laboratoire.

Je tiens aussi à remercier tous les gens que j'ai côtoyés durant ces dix semaines qui ont tous été très aimables envers moi et qui m'ont permis de m'adapter rapidement à ce nouveau cadre de travail.

J'adresse aussi mes remerciements à Mme Agnès GUESNON qui m'a aidé à la rédaction de ce rapport et qui était toujours là pour répondre à mes questions.

J'aimerais aussi remercier Mr Yannick REMION, Professeur à l'IUT Reims-Châlons-Charleville pour m'avoir aidé dans mon travail concernant le traitement d'images, ainsi que Mr Nicolas CHOISET pour m'avoir aidé sur les problèmes concernant le Visual Basic.

Enfin je voudrais remercier tous mes enseignants de l'IUT Reims-Châlons-Charleville qui m'ont enseigné une grande partie des connaissances nécessaires à la réalisation de ce projet.

Résumé du stage

Ce stage a consisté à réaliser une interface graphique réalisée en Visual Basic pour sélectionner une zone dans une image puis calculer une valeur moyenne des pixels de cette zone. J'avais aussi à faire de la détection automatique de formes dans ces images.

Mot clefs : images paramétriques, Visual Basic, C++, interface graphique, détection automatique, traitement d'image, calcul de moyenne.

Sommaire

Remerciements....1
Résumé du stage....2
Sommaire....3
Liste des annexes....4
Liste des figures....5
Introduction....6
Présentation du laboratoire....7
1) **Présentation globale**....7
a) Description....7
b) Localisation....8
2) **Travaux de recherche**....9
3) **Organigramme de l'équipe de recherche**....10
Présentation de l'environnement informatique....11
1) **Présentation du contexte informatique**....11
2) **Configuration**....11
3) **Logiciels utilisés**....12
a) Logiciels de bureautique....12
b) Langages de programmation....12
c) Logiciels divers....13
Présentation du travail personnel....14
1) **Contexte du stage**....14
a) Résumé du travail demandé....14
b) Points intéressants....15
2) **Sélection de zones dans une image**....16
a) Généralités....16
b) Réalisation du travail....17
- **Sélection de la zone**....17
- **Affichage de l'image**....18
- **Calcul de la moyenne dans la zone sélectionné**....20
c) Eléments rajoutés....22
d) Conclusion sur ce travail....23
3) **Détection du film support dans les images paramétriques**....24
a) Quelques notions sur les traitements d'images utilisés....24
- **Histogramme**....24
- **Profil**....24
- **Seuillage binaire**....25
- **Combinaison d'images**....26
- **Opérateurs de voisinage**....26
b) Application des ces outils....29
- **Positionnement du travail**....29
- **Traitements effectués pour la détection du film**....29
- **Adaptation de l'interface graphique**....32
c) Conclusion....33
Conclusion....34
Glossaire....35

Liste des annexes

Annexe A : Présentation du Microscope Electronique à Balayage Transmission (MEBT)....A

Annexe B : Code du calcul de moyenne.....B

Annexe C : Exemples de combinaisons d'images.....….....D

Annexe D : Masques de convolution.....E

Liste des figures

Figure 1 : Exemple d'images à améliorer..... 14

Figure 2 : Exemple d'images contenant le film support..... 15

Figure 3 : Résultat obtenu sans et avec l'algorithme de Bresenham.....18

Figure 4 : Correspondance entre un tableau de valeur et son image.....18

Figure 5 : Exemple d'utilisation d'une pile.....20

Figure 6 : Explications du fonctionnement du calcul de moyenne..... 21

Figure 7 : Relation entre le tableau «image» et le tableau des positions utilisées..... 22

Figure 8 : Présentation du programme réalisé.....23

Figure 9 : Histogramme des niveaux de gris dans une image.....24

Figure 10 : Profil de la ligne rouge sur le zèbre.....24

Figure 11 : Exemple d'une image seuillée au niveau de gris 128.....25

Figure 12 : Exemple d'une combinaison d'image avec un opérateur moyenne..... 26

Figure 13 : Exemple de dilatation et d'érosion sur une image.....27

Figure 14 : Exemple de calcul du pixel médian.....27

Figure 15 : Résultat du filtre médian sur une image bruitée.....28

Figure 16 : Explication sur le filtre écart-type..... 28

Figure 17 : Fonctionnement de la convolution.....28

Figure 18 : Masque de convolution de Sobel suivant l'axe des X et des Y.....29

Figure 19 : Explication sur le seuillage à partir de la moyenne dans l'histogramme..... 30

Figure 20 : Résultat de la détection de contour.....30

Figure 21 : Explications sur le rajout de certains pixels de l'image initiale..... 31

Figure 22 : Résultat de la détection du film support..... 31

Figure 23 : Présentation des outils de détection du film..... 32

Introduction

Après un nombre important d'envois de lettres de candidature et de Curriculum Vitae dans la Marne et dans la région parisienne, j'ai reçu un avis positif du Laboratoire de Microscopie Electronique Analytique de la faculté des sciences de Reims. J'ai eu par la suite un entretien avec le directeur de ce laboratoire qui ma proposé de travailler sur des images afin de créer un programme qui permettait de sélectionner des zones à l'intérieur de ces images pour y effectuer des opérations sur les pixels sélectionnés ; Il ma proposé un autre travail qui consistait à détecter automatiquement des zones de ces images.

Ce travail m'a tout de suite intéressé car il me permettait de mettre en application l'enseignement que j'avais reçu lors de ces deux années à l'*IUT* (1) de Reims-Chalôns-Charleville site de Reims et encore plus spécifiquement lors de ma deuxième année d'IUT car ce stage mettait en application le cours de traitements d'images que j'ai reçu.

Dans un premier temps, je présenterai le laboratoire : sa localisation géographique, les travaux de recherches qui y sont développés ainsi que le personnel.

Dans un second temps, je présenterai l'environnement informatique présent sur les lieux du stage (les matériels et les logiciels utilisés).

Dans un troisième et dernier temps, je développerai mon travail personnel : les méthodes, les résultats et les problèmes rencontrées lors de ce stage effectué pendant ces dix semaines au laboratoire.

(1) : Voir Glossaire page 35

Présentation du laboratoire

1) Présentation globale

a) Description

Mon stage a été réalisé au sein du Laboratoire de Microscopie Electronique Analytique (LMEA) de l'Université de Reims Champagne-Ardenne.

Ce laboratoire est une composante de l'unité INSERM ERM 0203 «Interfaces Biomatériaux/Tissus Hôtes». Les membres de ce laboratoire sont tous des physiciens travaillant en étroite collaboration avec les biologistes et odontologues de l'ERM 0203 situés sur le campus de faculté de médecine.

L'unité ERM 0203 a été crée en 2002 par l'Institut National de la Santé et de la Recherche Médicale (INSERM). Cette reconnaissance a été confirmée pour 4 ans, donc pour toute la durée du contrat quadriennal 2004-2007. Cette unité est une des composantes de l'Institut Fédératif de Recherche (IFR 53) qui regroupe plusieurs laboratoires INSERM, CNRS et équipes d'accueil (EA) de l'Université.

Cette unité a été reconnue dans le cadre des structures E.R.I.T.M de l'INSERM. Ce sigle signifie «Equipe de Recherche et d'Innovation Technologique et Méthodologique ». Le but assigné à ces structures est de développer des méthodes innovantes d'investigation et de caractérisation dans le domaine biomédical.

Pour ce qui concerne le LMEA, c'est dans le domaine de la microanalyse des cellules, des tissus et des biomatériaux que l'accent est porté. Les méthodologies développées concernent les microspectrométries X et de pertes d'énergie des électrons (EELS : Electron Energy Loss Spectroscopy) associées aux cryométhodes. Leurs mises en œuvre nécessitent l'utilisation d'un microscope électronique moyenne tension (300 kV) fonctionnant en mode transmission (MET) et balayage transmission (Voir Annexe A).

Un des points forts des méthodologies innovantes développées au sein du LMEA concerne l'imagerie chimique ou paramétrique des échantillons biologiques, des biomatériaux et des interfaces biomatériaux/tissus hôtes. C'est ainsi par exemple qu'on citera la mise en œuvre de l'imagerie paramétrique de la mesure locale de la concentration en eau intracellulaire par *spectrométrie EELS*[(1)] avec une résolution latérale de 200 nanomètres. Cette méthodologie permet d'identifier les évolutions des concentrations en eau à l'échelle des compartiments subcellulaires liés à des effets pathologiques ou inflammatoires.

(1) : Voir Glossaire page 35

b) Localisation

Le Laboratoire de Microscopie Electronique Analytique où j'ai effectué mon stage est situé au pôle technologique Farman à Reims.

Il est situé 21 rue Clément Ader.
Pour y accéder, vous pouvez voir ci-contre le plan d'accès.

Historique de la carte :

1. Rue Clément Ader
2. Avenue Henri Farman / N44
3. Université, Campus Scientifique, IUT
4. Parc Pommery
5. Avenue Nicephore Niepce / N244
6. Sortie Autoroute Reims / Cormontreuil
7. Parc des Expositions

Les rues oranges indique l'itinéraire à suivre pour arriver au LMEA à partir de la sortie d'autoroute

Le laboratoire se trouve à l'extérieur du campus du moulin de la housse.
Ce site a été créé en 1989 grâce à un cofinancement de la Région Champagne Ardenne, du Département de la Marne et de la ville de Reims pour accueillir la première unité INSERM créée à Reims ; il s'agissait de l'unité 314.
Par un cofinancement Etat-INSERM le laboratoire s'est équipé du microscope électronique à balayage transmission. Celui-ci a dû être installé sur une dalle antisismique pour l'isoler des vibrations extérieures (vibrations dues à la proximité du périphérique).

2) Travaux de recherche

Les travaux de recherche de l'ERM 0203 ont pour but d'étudier les interfaces biomatériaux/tissus hôtes. Les biomatériaux étudiés concernent les revêtements prothétiques et les matériaux de comblement osseux. L'approche vise d'une part, à élaborer et caractériser des revêtements phosphocalciques par électrodéposition, des bioverres macro ou nanoporeux et d'autre part, à fonctionnaliser ces matériaux implantables en site osseux en les dotant de propriétés biologiques spécifiques qui résulteront en une stimulation de l'activité ostéoblastique et de la production de la matrice osseuse. C'est ainsi que les chercheurs de l'ERM 0203 s'orientent vers la conception de nouvelles générations de matériaux biomimétiques, via l'ingénierie tissulaire. Dans ce but, ils élaboreront des matériaux, substituts osseux et/ou revêtements prothétiques, par adjonction de peptides qui auront été préalablement incorporés dans des microcapsules afin de permettre une libération contrôlée et prolongée des produits actifs dans le temps.

Plus spécifiquement, les travaux réalisés au LMEA consistent à étudier à l'échelle submicrométrique les interfaces Biomatériaux/tissus hôtes. Ces interfaces étant de très petites dimensions, c'est la raison pour laquelle les spectrométries qu'ils emploient font appel aux sondes électroniques (microscopie électronique) dont on peut réduire la taille à quelques nanomètres. Ainsi les spectrométries qu'ils développent donnent des informations sur des zones très localisées aux interfaces.

Ces travaux de recherche s'effectuent suivant deux axes principaux :

1- l'influence des caractéristiques physico-chimiques des matériaux bioactifs sur le processus de bioactivité *in vitro* et *in vivo*;

2- l'étude de la modulation de l'activité cellulaire par les biomatériaux en développant :

- la détection d'altérations cellulaires précoces engendrées par les biomatériaux;
- l'étude des interactions cellules/matrice extracellulaire/biomatériaux, visant à évaluer l'implication des ostéoblastes et des macrophages humains dans les processus de réaction inflammatoire et de dégradation matricielle.

Pour ce faire les chercheurs du LMEA ont mis en œuvre des développements méthodologiques spécifiques qui concernent :

- **Les ultracryométhodes :** Pour conserver l'intégrité du matériel biologique cellulaire (morphologie et composition ionique), les ultracryométhodes sont développées (réalisation de coupes froides hydratées comportant la cryofixation(fixation par le froid) à 77 K suivie d'une cryoultramicrotomie (coupe à froid) et du cryotransfert(observation à froid) dans le microscope).

- **L'optimisation de la détection d'éléments trace et cartographie chimique en spectrométrie EELS :** Sur une base constituée d'un MEBT 300 kV couplé à un spectromètre EELS, l'équipe a développé un dispositif fonctionnant en mode "spectre-image" permettant de réaliser des images chimiques avec une résolution latérale pouvant atteindre quelques dizaines de nanomètres.

- **La mesure locale de la concentration en eau à l'échelle subcellulaire par spectrométrie EELS** : La mesure des concentrations locales en eau dans les

compartiments cellulaires est basée sur les variations fines enregistrées dans les spectres EELS dans laquelle les composés macromoléculaires (protéines, acides nucléiques, sucres, lipides) et l'eau présentent chacun une signature spectrale significative. La cartographie de la distribution en eau est réalisée à l'aide de la technique des spectres-images (les spectres sont enregistrés point par point, ils subissent ensuite une série de traitement ce qui permet de reconstruire l'image paramétrique de cette zone). Couplée à la *spectroscopie X* (1), elle permet d'établir les concentrations ioniques dans les cellules

3) Organigramme de l'équipe de recherche

Liste nominative des enseignants-chercheurs statutaires :

Nom, Prénom	Fonction
LAURENT-MAQUIN Dominique	Directrice PU-PH
BALOSSIER Gérard	Codirecteur PU
BENHAYOUNE Hicham	Maître de conférences
FAURE Joël	PU
LAQUERRIERE Patrice	Maître de conférences
MICHEL Jean	Maître de conférences
BRISSET Lucien	PU-PH
LORIMIER Sandrine	Maître de conférence PH
GELLÉ Marie-Paule	Maître de conférence PH
TERRYN Christine	Assistante Associée
BANCHET Vincent	Attaché Temporaire d'Enseignement et de Recherche

PU : Professeur Université.
PH : Praticien Hospitalier.

Liste nominative des Ingénieurs, Techniciens, Administratifs :

Nom, Prénom	Fonction
DELIGNY Francis	Ingénieur d'Etude Informatique/Electronique
NICOLAS Jean-Louis	Ingénieur de Recherche en Electronique
PERCHET Alain	Ingénieur d'Etude en Mécanique
WORTHAM Laurence	Ingénieur d'Etude en Microscopie
BRUNET-MICHEL Monique	Secrétaire
BOUTHORS Sylvie	Ingénieur d'Etude en Biologie
GUILLAUME Christine	Technicienne en Biologie
LALUN Nathalie	Adjoint Technique en Biologie
BAYDAR Cendrine	Secrétaire

De plus, dans ce laboratoire, il y a un doctorant en 1ère année de thèse et un stagiaire en DEA.

(1) : Voir Glossaire page 35

Présentation de l'environnement informatique

1) Présentation du contexte informatique

Au sein du laboratoire, il n'existe pas à proprement parler de service informatique. Mais le personnel du laboratoire se sert de l'outil informatique sous toutes les formes.

De plus Monsieur Francis DELIGNY qui est ingénieur d'études dans le domaine de l'électronique, informatique, et des réseaux, est chargé d'assurer la maintenance du parc informatique des deux laboratoires présents sur le site.

2) Configuration

Dans le laboratoire, chaque personne possède son propre ordinateur configuré de manière similaire. En général, les ordinateurs possèdent un processeur Intel Pentium (III ou IV) avec un lecteur CD et un lecteur disquette, et pour certains un graveur CDR/CDRW et un écran 17 pouces.

L'environnement utilisé est Microsoft Windows, en général Windows 2000 hormis un poste nécessitant Windows 98 pour pouvoir faire fonctionner le logiciel d'acquisition des spectres EELS dont l'interface spectromètre/ordinateur a été réalisé au sein du laboratoire.

Historiquement, sont présents sur le site deux réseaux internes, le premier a été mis en place dans les années 90 à l'aide de câble BNC puis a été abandonné. En effet, au fur et à mesure de la progression de l'équipement informatique des personnels du laboratoire, la longueur du câble réseau nécessaire dépassait la longueur maximale de 185m. Un nouveau réseau par câblage RJ45 (le RJ45 est une norme de câble) a donc dû être mis en place.
Le réseau BNC n'est plus beaucoup utilisé mais toujours opérationnel, il est notamment possible de se connecter au serveur pour pouvoir récupérer des données.
Actuellement chaque ordinateur est relié par un réseau câblé en RJ45 connecté à un serveur interne (avec un processeur Intel Pentium IV à 2.8 Ghz, 1Go de RAM, deux disques dur : un 60 Go et un 40 Go tournant sous Microsoft Windows 2000 Server). Sur ce serveur, chacun dispose de son propre compte de connexion qui permet de pouvoir récupérer ces données centralisées sur le serveur lorsque l'on se connecte sous un autre ordinateur que le sien.

Un réseau intranet permet de se connecter à son propre ordinateur à partir d'un autre ordinateur du laboratoire, mais aussi de transmettre des données à quelqu'un d'autre sans être obligé de passer par un support comme les CD-ROM ou les disquettes, ou encore de faire des copie de disque dur à disque dur,.... Celui-ci sert surtout pour transférer des données acquises par le microscope et les spectromètres sur les ordinateurs personnels.

Chaque ordinateur relié au réseau possède une connexion Internet très haut débit, en effet le laboratoire est relié à la faculté des sciences par une connexion à fibre optique (connectée au réseau métropolitain avec un débit de 100 Mbit/S). La sécurité et les accès

Internet sont gérés par le CRI (Centre des Ressources Informatiques). Il s'agit d'un service spécifique se trouvant à la faculté des sciences qui gère le réseau Internet, en s'occupant notamment de la sécurité du réseau en mettant en place par exemple des *firewalls*(1), des antivirus, des *anti-spams*(1), etc... ainsi que le réseau interne à la faculté.

3) Logiciels utilisés

a) *Logiciels de bureautique*

La plupart de logiciels de bureautique utilisés ici sont les logiciels du pack Microsoft Office ; en effet, tous les ordinateurs du laboratoire sont sous un environnement Windows.

- *Microsoft Word :* Il s'agit d'un logiciel de traitement de texte. Ce logiciel d'utilisation pratique et d'interface conviviale, permet d'effectuer des mises en pages de qualité pour divers documents : articles scientifiques, rapport d'activités, thèse, cours, note de services...

- *Microsoft Excel :* Il s'agit d'un tableur utilisé pour réaliser des graphiques, des tableaux et des statistiques. Ce logiciel est notamment utilisé pour mettre en page ou effectuer des calculs statistiques sur les résultats des expériences mises en place au laboratoire.

- *Microsoft Power Point :* Logiciel de Présentation Assisté par Ordinateur (PAO). Il est utilisé pour préparer des exposés scientifiques présentés lors de congrès ou de séminaires, ainsi que pour certains cours. Ce logiciel convivial et simple d'utilisation permet de créer des présentations dynamiques et vivantes.

Ces logiciels ont été choisis car les formats de documents correspondants («.doc» pour Microsoft Word, «.xls» pour Microsoft Excel et «.ppt» pour Microsoft Power Point) sont lisibles par la plupart des ordinateurs extérieurs au laboratoire. Cela est utile pour pouvoir créer des documents sur un ordinateurs et de pour pouvoir les utiliser sur un autre.
De plus, des ordinateurs ayant des systèmes d'exploitations Linux ou même des Macintosh ont des logiciels permettant de lire ce type de fichiers.

b) *Langages de programmation*

Dans le laboratoire, plusieurs langages de programmation sont utilisés. Les interfaces d'acquisition des spectres EELS ont été développées en C. Ainsi les premiers programmes de traitement de données ont également été développés en C pour pouvoir assurer une compatibilité entre ceux-ci. Ces derniers ont été réalisés avec BORLANC C qui permet de créer des interfaces graphiques.
Plus récemment, de nouveaux programmes de traitement ont été réalisés en Visual Basic à l'aide du logiciel Microsoft Visual Basic. En effet ce logiciel permet de créer une interface utilisateur/machine plus conviviale avec des menus déroulant, des *boîtes de dialogues* (1), des barres de défilement…

(1) : Voir Glossaire page 35

La programmation est utilisée ici pour réaliser les logiciels de traitements de données enregistrées par le microscope (des images) et/ou les spectromètres EELS et X (des spectres). Au laboratoire, un de ces logiciels permet de déterminer la concentration des ions dans les cellules à partir des spectres X. Un autre permet la mesure de la concentration massique en eau dans les cellules à partir des spectres EELS ou du microscope électronique (images).

c) *Logiciels divers*

Quelques autres logiciels sont couramment utilisés dans le laboratoire :

- *Microsoft Internet Explorer :* Il s'agit d'un logiciel permettant d'aller sur Internet pour effectuer par exemple des recherches bibliographiques.

- *Eudora mail :* Il s'agit d'un logiciel pour recevoir ces emails. Ce logiciel a été préféré à des logiciels comme Microsoft Outlook pour des raisons de sécurité. En effet, ce dernier ouvrait les mails systématiquement ce qui pouvait être dangereux si l'email était porteur de virus.

- *EDAX :* Logiciel d'acquisition de spectre X qui a été vendu avec le spectromètre X.

- *Logiciel d'acquisition :* Logiciel qui permet l'acquisition des spectres EELS. C'est un logiciel qui a été crée au laboratoire en langage C.

Présentation du travail personnel

1) Contexte du stage

Ce travail a été effectué dans le cadre de la suite d'une thèse de doctorat soutenue au laboratoire. Cette thèse avait pour objet de développer une mesure de la concentration massique en eau par l'acquisition d'images (issu du microscope) et par spectroscopie EELS.
La mesure de la concentration en eau couplée à la mesure des concentrations ioniques permet de mettre en évidence certaines altérations des cellules biologiques (maladies...).

Les résultats se présentent sous forme d'images paramétriques (reconstruites à partir des concentrations en eau déterminées par la méthode précédente).
Pour obtenir les valeurs moyennes des concentrations en eau dans les différents compartiments de la cellule (noyau, cytoplasme, granules, …), on effectuait manuellement la moyenne en pointant pixel par pixel la zone intéressante.

L'objectif de ce stage a donc été de réaliser une interface graphique permettant de délimiter la zone d'intérêt à l'aide de la souris puis d'effectuer automatiquement la moyenne sur les pixels sélectionnés.

Dans le cadre de ce stage j'ai travaillé avec Mademoiselle TERRYN, qui a développé ces différentes méthodes de mesures de la concentration en eau.

Pour réaliser ce travail, j'ai eu à ma disposition un ordinateur possédant un processeur Intel Pentium III avec 195 Mo de RAM, Windows 2000 Professionnel comme système d'exploitation, un lecteur de CD-ROM, un lecteur de disquettes et possédant des ports USB.

a) Résumé du travail demandé

Dans un premier temps, j'ai dû réaliser une interface graphique permettant d'afficher une image stockée dans un fichier au format brut (Voir Page III.2.a) et sélectionner à l'aide de la souris une zone de cette image sur laquelle nous allons effectuer un calcul de moyenne.

Dans un deuxième temps, il m'a été demandé d'améliorer la qualité de certaines images comme celles présentées ci-dessous (Fig. 1).

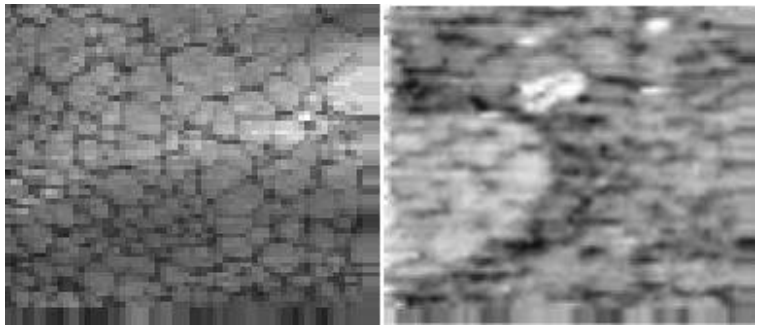


Fig. 1 : Exemple d'images à améliorer.

Ce travail nécessitant des connaissances et compétences techniques d'un niveau élevé par rapport au programme de *DUT* (1) Informatique, ce travail n'a pas pu être réalisé.

Afin de poursuivre la première tâche proposée, nous souhaitons appliquer ce même logiciel aux images paramétriques obtenues à partir des spectres EELS. Dans ce type d'images, les pixels correspondant au film support (Network-Like Support Film)(Fig. 2) ne doivent pas être pris en compte pour effectuer les moyennes afin éviter d'avoir des résultats entachés d'erreur. Le but de ce nouveau travail est donc de détecter ce film support avant d'effectuer la moyenne.

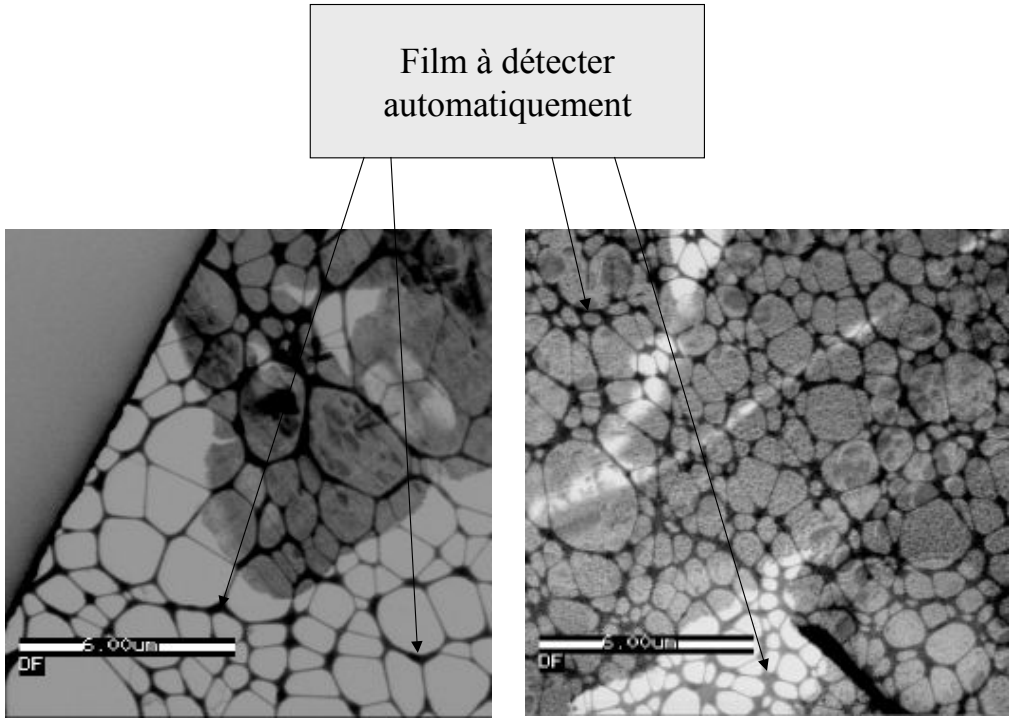


Fig. 2 : Exemple d'images contenant le film support.

b) Points intéressants

Plusieurs points intéressants sont à souligner :

- Tout d'abord j'ai appris à créer des programmes interactifs et conviviaux à l'aide de Visual Basic.
- Ensuite j'ai travaillé sur des outils de traitements d'images que l'on avait étudiés et développés en cours.

De plus tous ces travaux m'ont permis de mettre en pratique beaucoup de connaissances qui m'ont été enseignées durant ces deux années d'IUT Informatique.

(1) : Voir Glossaire page 35

2) Sélection de zones dans une image

Le but de ce travail était de pouvoir sélectionner une zone d'intérêt (compartiments : noyau, cytoplasme, granules...) dans une image paramétrique de concentrations massiques en eau pour effectuer un calcul de moyenne des niveaux de gris dans la zone sélectionnée à l'aide de la souris. Ce qui éviterait d'effectuer ce travail judicieux manuellement en prenant quelques pixels de la zone et ensuite d'effectuer le calcul de moyenne.

a) Généralités

Une image peut être assimilée à un tableau deux dimensions de pixel. Un pixel peut être assimilé à une couleur (qui est un triplet de valeur Rouge-Vert-Bleu) ou de niveau de gris (c'est à dire une seule valeur).

Les images qui me sont fournies sont des images paramétriques reconstruites à partir des images acquises au microscope à l'aide d'un logiciel créer par Mlle TERRYN Christine, mon maître de stage. Il s'agit d'images en niveau de gris de 512x512 pixels. Elles sont au format RAW, c'est-à-dire un fichier contenant une suite de caractères qui représente le contenu binaire de l'image. En effet un niveau de gris étant compris entre 0 et 255 (le noir étant désigné par 0 et le blanc par 255). Ces images sont codées en caractères non signés. Un caractère signé est un nombre codé sur 1 octet (soit 8 bits) ce qui représente un nombre compris entre -127 et 128, pour un caractère non signé, la plage s'étend de 0 à 255.

Le fichier contient donc une succession de nombres codés en binaire sur des caractères représentant le niveau de gris (pour une image couleur d'une succession de 3 caractères représentant successivement la valeur du rouge, du vert et du bleu).

Ce type de format est un format très basique qui ne contient que les informations de couleur des pixels. Il faut connaître la taille et le type d'images (couleur ou niveau de gris) pour pouvoir s'en servir.

b) Réalisation du travail

- **Sélection de la zone**

Ayant réalisé durant mon année universitaire une application qui consistait à tracer des points lorsque l'on bougeait la souris en laissant un bouton appuyé, je retrouvais dans le travail demandé le même type de problème. La différence porte sur le langage de programmation utilisé puisque au lieu du Visual Basic, j'ai utilisé la librairie graphique OpenGL qui permet de créer des scènes en 2D ou en 3D avec la librairie GLUT qui permet de pouvoir afficher à l'écran les scènes créées.

Notion de programmation événementielle :

Pour pouvoir programmer sous un environnement graphique comme l'API Windows, QT, GTK++,... (sont des *GUI* (1)), il faut comprendre leurs modes de fonctionnement. Lorsqu'un événement se produit (clic de souris, mouvement de la souris, utilisation d'une touche du clavier,...) le système d'exploitation indique à tous les programmes qui sont ouverts l'événement qui vient de se produire et chacun d'entre eux réagit en fonction de l'événement. Par exemple, l'application graphique que j'ai créée va réagir au clic du bouton gauche de la souris, au relâchement du bouton gauche et au déplacement de celle-ci lorsque l'événement se produira sur la zone active mon programme.

Fonctionnement de la gestion des événements par Visual Basic :

Lorsque l'on crée une application graphique avec Visual Basic, il signale à chaque objet les événements auxquels il doit réagir. Un exemple : si je crée un *objet* «*Text Box*»(1) on peut le faire réagir différemment à l'événement «Clic» ou «Double Clic».

Les événements utilisés afin de me servir de la souris pour sélectionner une zone lorsque l'on clique sur mon objet «*Picture Box*»(1) appelé «pb» sont les suivants :

- J'utilise la fonction «pb_Mousedown» lorsque je clique sur un bouton de la souris. Les paramètres de cette fonction sont : un identifiant du bouton appuyé et la position (X,Y) de la souris par rapport à mon objet.
- J'utilise aussi la fonction «pb_MouseUp» appelée lorsque l'on lâche le bouton de souris, les paramètres de cette fonction sont un identifiant du bouton appuyé et la position (X,Y) à laquelle on a lâché le bouton.
- La troisième fonction utilisée est celle qui est appelée lorsque la souris bouge, cette fonction est «pb_MouseMove» qui indique la nouvelle position de la souris.

L'idée est plutôt simple : pour pouvoir délimiter une zone à la souris, lorsque l'on clique sur le bouton de gauche (le bouton de souris choisi pour délimiter la zone), je mémorise dans une *variable*(1) l'événement : bouton enfoncé. Lorsque la souris bouge et que le bouton est enfoncé alors je sauvegarde la position de la souris ; puis je dessine un point à cet endroit

(1) : Voir Glossaire page 35

jusqu'à ce que l'on relâche le bouton de souris. On obtient au final une zone comme le montre la figure 3a. Le problème est que cette zone présente une délimitation discontinue et que pour effectuer le calcul de moyenne il me fallait une zone fermée. J'ai donc décider de codé un algorithme de tracé de droite (*algorithme de Bresenham*(1)) qui permettait de tracer une ligne entre deux points et de connaître tous les points formant cette zone. J'obtenais donc une zone fermée comme le montre la figure suivante (Fig. 3b).

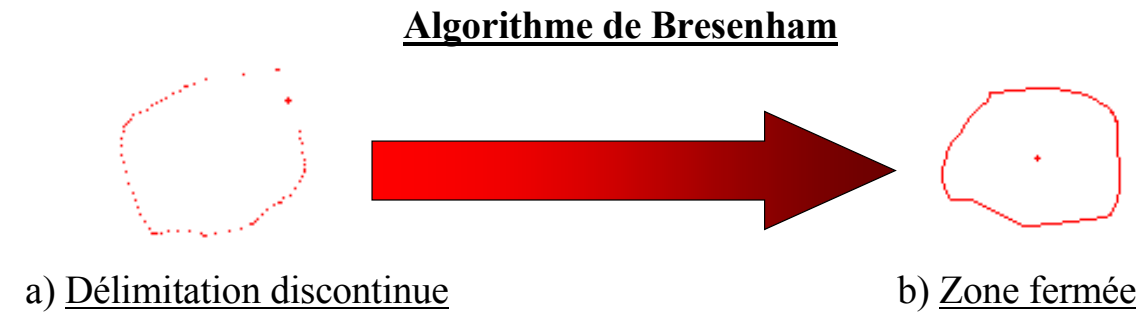


Fig. 3 : Résultat obtenu sans et avec l'algorithme de Bresenham.

- **Affichage de l'image**

C'est cette étape de réalisation qui m'a posé le plus de difficultés.

Pour pouvoir afficher et garder en mémoire les valeurs des pixels, j'ai stocké dans un tableau à deux dimensions la valeur de chaque pixel de l'image à traiter. Par exemple pour une image 4x4 pixels, je la stocke dans un tableau 4x4 comme le montre la figure suivante (Fig. 4).

Fig. 4 : Correspondance entre un tableau de valeur et son image.

Je lis le fichier contenant les valeurs des pixels valeur par valeur que je stocke dans un tableau (à l'endroit correspondant au pixel lu).

(1) : Voir Glossaire page 35

Le chargement est un peu long (environ 6.5s (1)) car en Visual Basic je n'ai pas trouvé d'autre méthode pour lire mon fichier que de le lire caractère par caractère, contrairement à d'autres langages comme le C++ où je peux lire tout mon fichier d'un coup et le stocker directement dans mon tableau.

C'est l'affichage de l'image qui était stocké dans mon tableau qui a posé quelques difficultés. La première solution a été d'afficher l'image point par point à l'écran. Cette méthode fonctionnait très bien mais elle s'avère coûteuse en temps (plus de deux minutes (1)). Je me suis alors aperçu que si l'on cachait la zone que l'on devait afficher et qu'on affiche ensuite l'image d'un seul bloc, cela était moins long (environ 7s (1)). Mais cela prenait tout de même trop de temps puisque à chaque réaffichage de la fenêtre, il fallait attendre 7s (1) pour pouvoir visualiser l'image.

J'ai donc cherché une autre solution. Je savais qu'il existait une fonction appelée «LoadPicture» qui permettait de charger dans mon objet «*PictureBox*». De plus cette fonction est une fonction de Visual Basic, cela rend ainsi l'opération d'affichage de l'image plus rapide.

Cette fonction nécessite le passage en paramètre d'une image de type bitmap (.bmp), curseur (.cur), icône (.ico), les fichiers avec codage des répétitions (.rle), les métafichiers (.wmf), les métafichiers étendus (.emf), les fichiers GIF (.gif) et les fichiers JPG (.jpg). Or le format des images que j'utilise est au format brut (.RAW). Ce format n'était pas supporté pas Visual Basic il a donc fallu trouver une solution ; j'ai donc décider de créer un programme réalisé en C++ qui me permettait de convertir mon image RAW en un autre format. En effet le C++ est un langage plus rapide pour effectuer des opérations de ce type car l'on accède directement à la mémoire de l'ordinateur ce qui permet d'accélérer certaines opérations qui seraient plus coûteuses au niveau temps en Visual Basic.

J'ai donc effectué des recherches sur Internet pour connaître la structure des fichiers bitmap par exemple. J'ai alors essayé de convertir mes images RAW en BMP ce qui la demandé un certain temps avant de réussir à obtenir une image bitmap qui ressemblait à l'image de départ. Lorsque j'essayais de charger cette image avec la fonction LoadPicture, cela engendrait une erreur lors de l'exécution car elle ne reconnaissait pas mon format d'images alors que des logiciels comme *XnView©* (2) réussissait à la lire. J'ai donc abandonné l'idée de créer des images BMP car j'avais déjà passer beaucoup de temps et cela ne fonctionnait pas.

Par la suite je me suis intéressé au format GIF mais il était encore plus difficile à créer que le format BMP. J'ai donc regardé le format JPG, celui-ci est l'un des formats les plus compliqués mais il existe des bibliothèques qui permettent de lire des fichiers JPG et d'en créer. Après quelques recherches là-dessus, j'ai trouver un programme qui convertissait des images de type PGM (Portable GrayScale), PPM (Portable PixMap), JPG, BMP... vers d'autres formats comme le JPG, BMP, TIF... Pour simplifier j'ai créé des images au format PGM qui est un format très simple à créer que je convertis au format JPG avec le logiciel. Au final j'obtiens donc des images au format JPG ce qui me permettait de les afficher quasi instantanément grâce à la fonction «LoadPicture» de Visual Basic.

(1) : Le temps indiqué correspond à des mesures effectuer à l'aide de Visual Basic.
(2) : Voir Glossaire page 35

- **Calcul de la moyenne dans la zone sélectionné:**

J'ai effectué le calcul de moyenne de deux méthodes différentes qui nécessitent l'utilisation d'une pile.

Explications sur les piles : Une pile est une structure de données qui permet de stocker plusieurs éléments. Une pile est une structure de type LIFO (Last In First Out) ce qui signifie que le premier élément à rentrer sera le dernier à sortir et inversement.
Prenons l'exemple d'une pile d'assiette, la dernière assiette que l'on mettra sur la pile sera la première à sortir et la première assiette sera la dernière à sortir.

Il y a deux notions à connaître sur les piles, la notion *d'empiler* et de *dépiler*.

Lorsque l'on parle d'empiler c'est lorsqu'on rajoute une valeur en haut de la pile, lorsque l'on parle de dépiler c'est lorsqu'on enlève la valeur qui est au sommet de la pile pour pouvoir s'en servir.(Fig. 5)

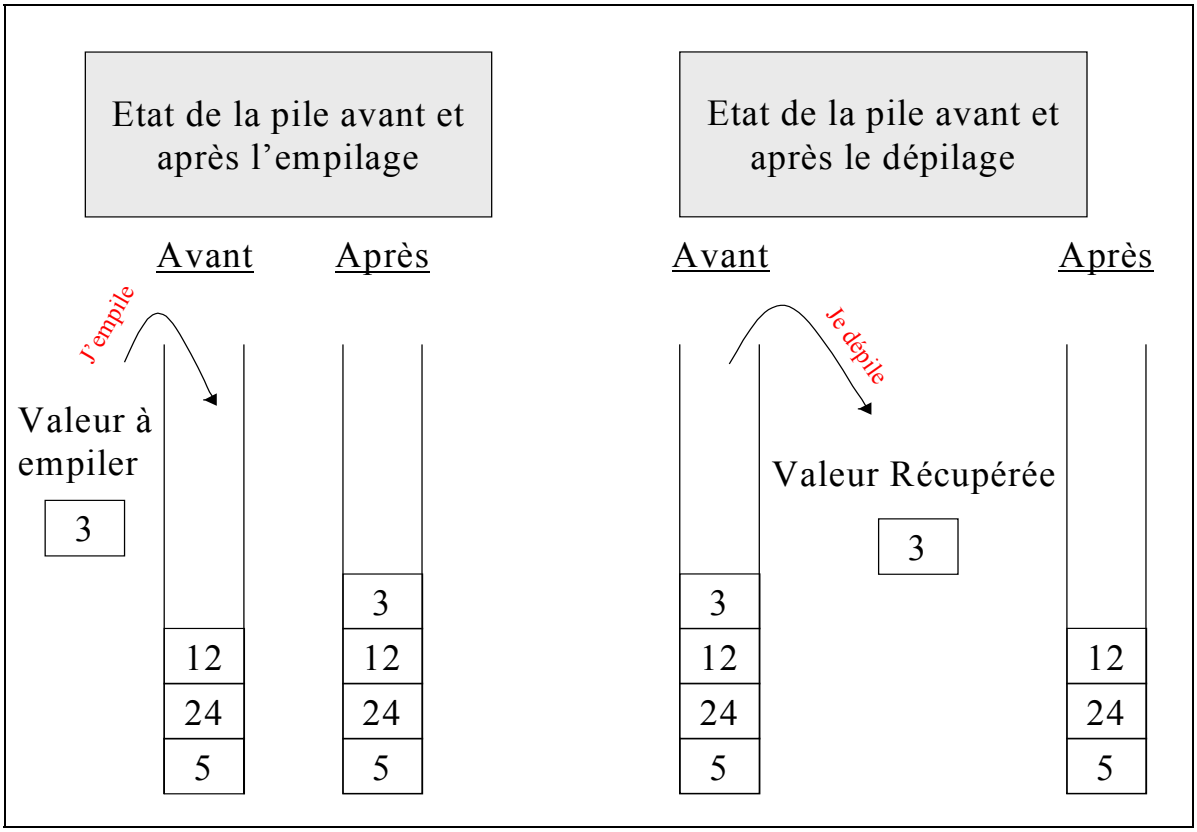


Fig. 5 : Exemple d'utilisation d'une pile

Le principe de ces méthodes est le suivant : au début ma pile est vide, j'empile la position du pixel central de la zone. Ensuite j'ai une boucle qui va s'effectuer jusqu'à ce que la pile soit vide. A chaque passage dans la boucle, je dépile la position du pixel qui se trouve au sommet de la pile. Je regarde la valeur de ce pixel dans mon tableau qui stocke l'image, puis je prends en compte cette valeur dans mon calcul de moyenne si elle n'a pas déjà été utilisée. Ensuite je regarde les positions adjacentes (en haut, en bas, à gauche, à droite) par rapport à la position que j'ai dépilée, si elles n'ont pas été encore utilisées et si elles ne font

pas partie ni du contour ni des bords, alors j'empile cette position sur la pile. Et je répète ces différentes étapes jusqu'à ce que la pile soit vide. En effet si la pile est vide, cela signifie que j'ai traité tous les pixels de ma zone. (Fig. 6)

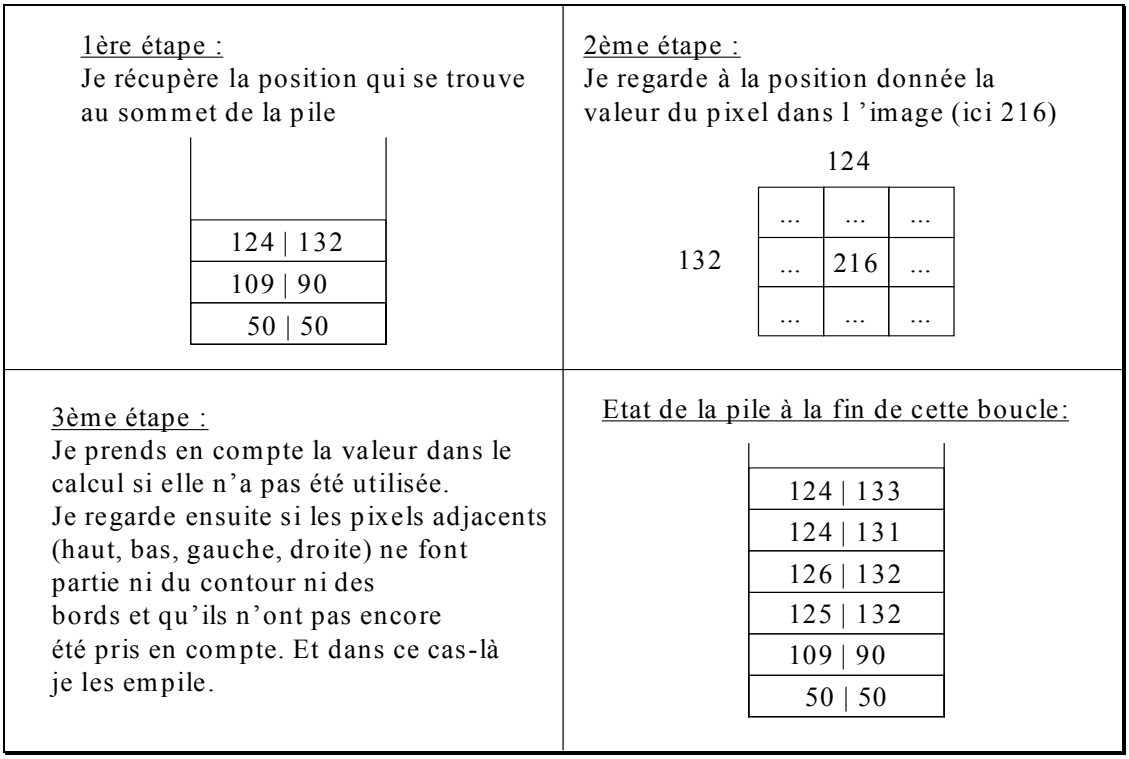


Fig 6 : Explication du fonctionnement du calcul de moyenne

La première méthode nécessite de stocker dans un tableau toutes les positions déjà prises en compte. Pour chaque nouvelle position, j'effectue une recherche pour savoir si elle est déjà présente dans le tableau. Cette méthode présente plusieurs inconvénients ; en effet, à chaque fois que je devais rajouter une nouvelle position, je devais redimensionner le tableau c'est-à-dire : créer un tableau avec une case de plus, recopier toutes les anciennes valeurs, et mettre à la fin de ce tableau la nouvelle valeur. Ceci devient vite coûteux en temps lorsqu'on arrive à des tableaux contenant un grand nombre de valeurs.

J'ai donc dû changer de méthode pour diminuer les temps de calcul.

Cette seconde méthode diffère de la première au niveau de la prise en compte des valeurs déjà utilisées (3ème étape, Fig. 6).

Cette seconde méthode est plus simple et plus efficace que la précédente ; en effet, au lieu d'utiliser des tableaux pour stocker les positions déjà prises en compte, je crée un deuxième tableau de la même taille que l'image et dans ce tableau, les cases 0 sont des cases non utilisées et les cases à 1 sont les cases indiquant soit le contour, soit que les pixels ont déjà été utilisés dans le calcul de la moyenne. (Fig. 7)

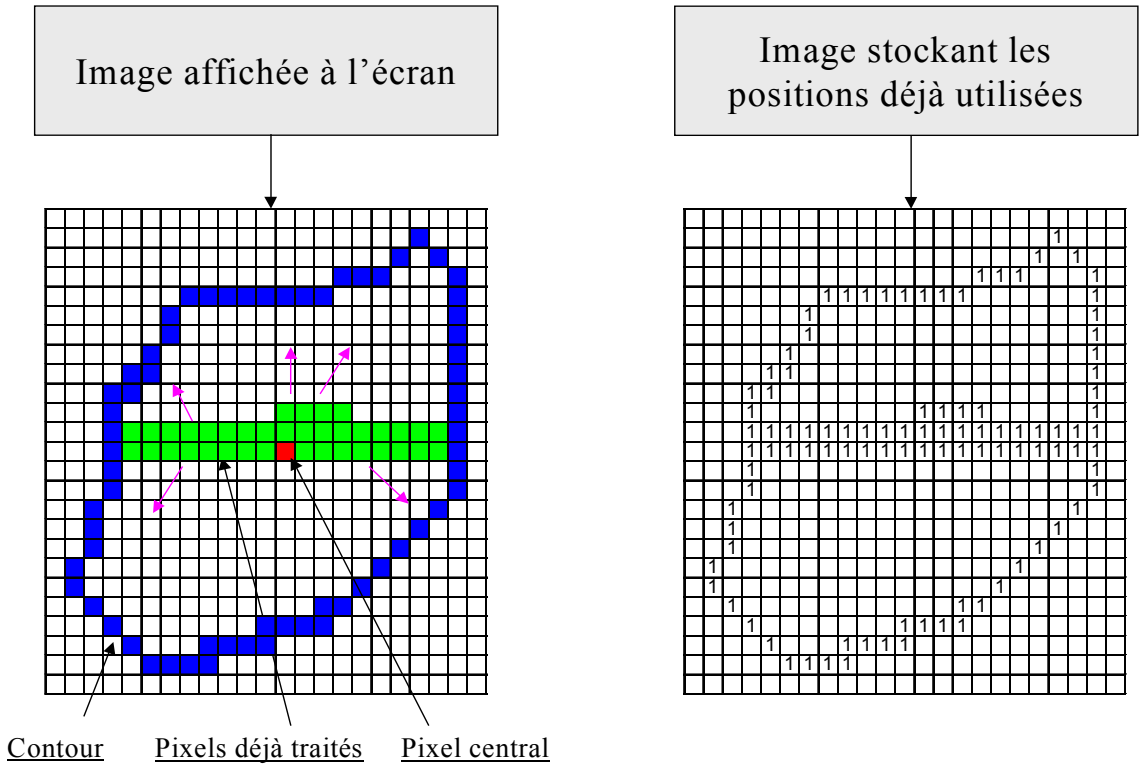


Fig. 7 : Relation entre le tableau «image» et le tableau des positions utilisées.
(Les flèches violettes → permettent de montrer qu'à partir du pixel central, les pixels déjà pris en compte vont se propager dans toute la zone).

Cette méthode est donc plus rapide, car pour savoir si les pixels adjacents à la position actuelle sont déjà «occupés», il suffit de regarder si le pixel correspondant dans l'image stockant les positions déjà utilisées s'il y à un 1 au lieu de faire une recherche dans mon tableau. De plus, pour signaler que cette position a déjà été prise en compte, il suffit alors de mettre un 1 à la position courante dans ce tableau au lieu de détruire le tableau de valeurs et le recopier dans un tableau plus grand comme dans le cas de la première méthode.
(Voir Annexe B pour le code de cette fonction).

c) *Eléments rajoutés*

Pour plus de convivialité j'ai rajouté des boites de dialogues[1] comme des barres de défilement lors du chargement des images (Fig. 8), une fenêtre d'information sur les images, sur les zones sélectionnées (Fig. 8) ; cette fenêtre contient divers renseignements comme un histogramme (Voir III.3 section Histogramme, Fig.8, Fig. 9), la moyenne de la zone (Fig. 8), le nombre de couleurs, le niveau de gris maximum et minimum...), un menu déroulant...

(1) : Voir Glossaire page 35

d) Conclusion sur ce travail

Pour ce premier travail, mon programme a parfaitement répondu aux attentes de mon maître de stage car il s'est révélé un outil précieux (surtout en gain de temps et pour la précision des calculs) pour le traitement de ces résultats. (Fig. 8)

Par ailleurs, ce travail m'a permis d'approfondir et de pratiquer le Visual Basic que je connaissais peu (en effet quelques connaissances en Visual Basic Application nous ont été enseignées pendant mon année universitaire). Ainsi j'ai pu réaliser un programme avec une interface graphique assez conviviale répondant au cahier des charges qui m'a été proposé.

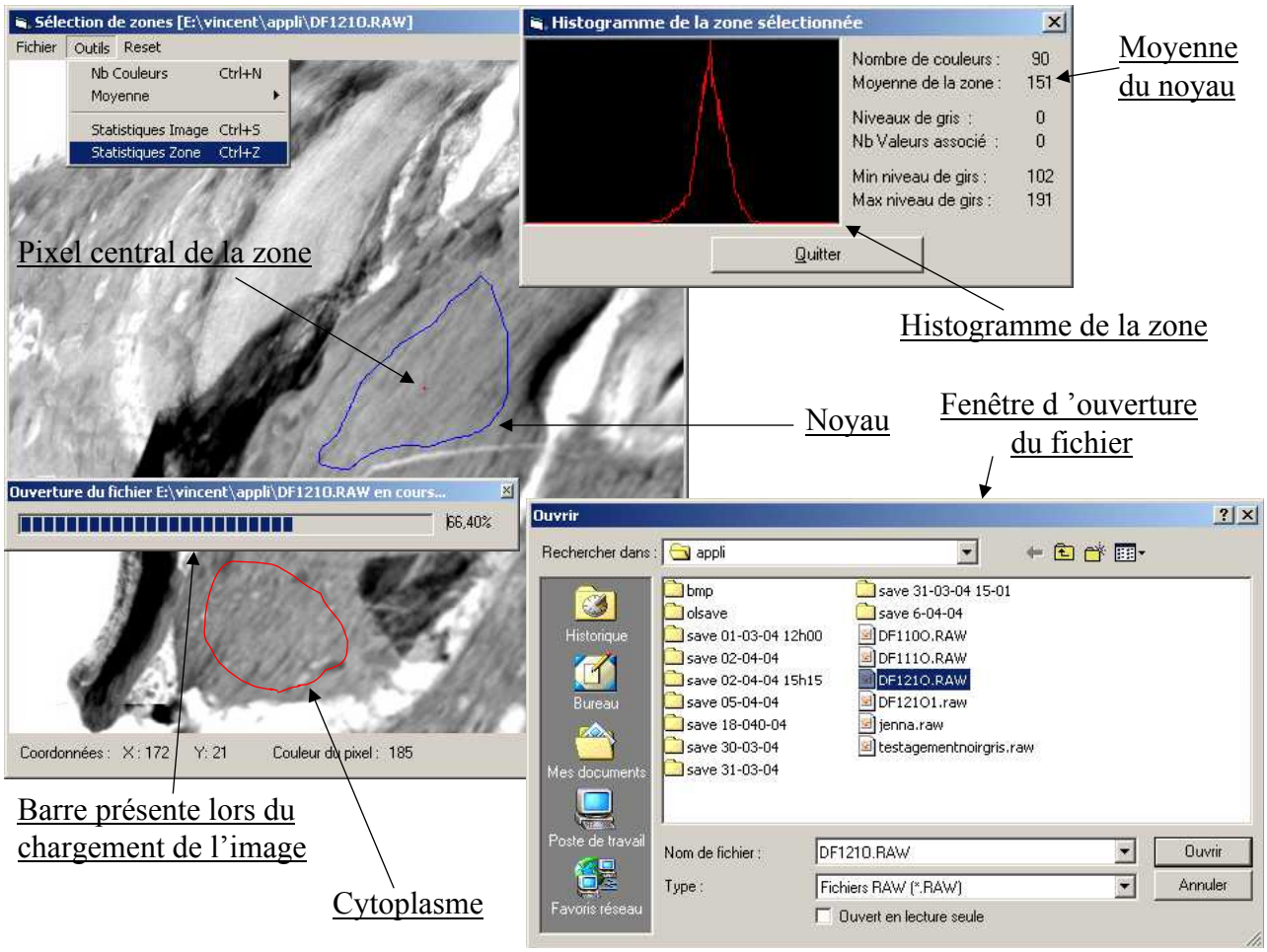


Fig. 8 : Présentation du programme réalisé.

Le temps approximatif de calcul dans le noyau est de 0,19s et de 0,12s dans le cytoplasme. Ce qui est infime comparer aux minutes passer pour effectuer ce calcul dans les différentes zones manuellement.

3) Détection du film support dans les images paramétriques

a) *Quelques notions sur les traitements d'images utilisés.*

- Histogramme

Un histogramme est un graphique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite).
Ainsi, l'histogramme d'une image en 256 niveaux de gris sera représenté par un graphique possédant 256 valeurs en abscisse, et le nombre de pixels de l'image en ordonnée. (Fig. 9)

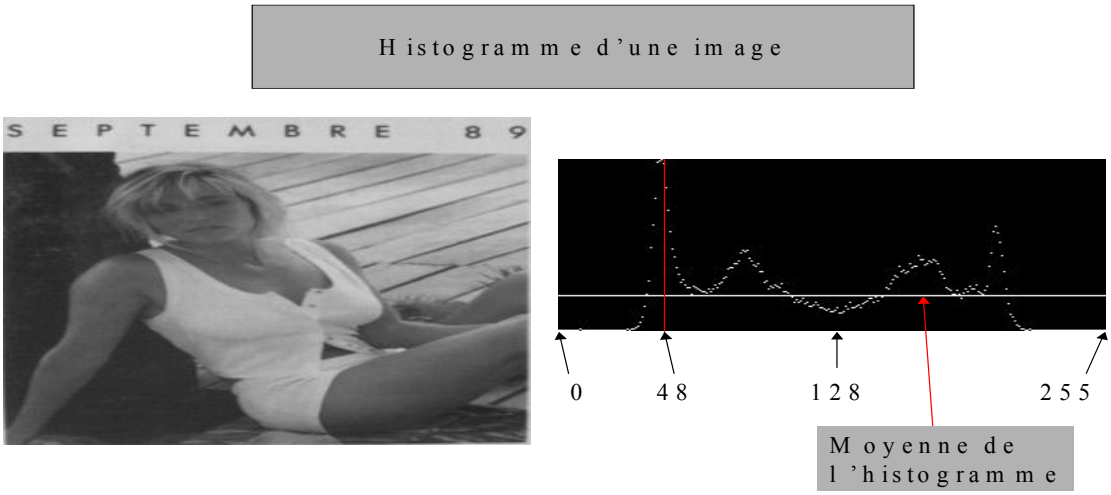


Fig. 9 : Histogramme des niveaux de gris dans une image.

Ici nous pouvons voir que le niveau de gris le plus présent est le niveau de gris 48(trait rouge). De plus sur cet histogramme nous pouvons voir une ligne horizontale blanche qui représente le nombre de valeur moyen dans l'histogramme.

- Profil

Un profil est un graphique qui correspond au niveau de gris de chaque pixel traversé par une ligne. (Fig. 10)

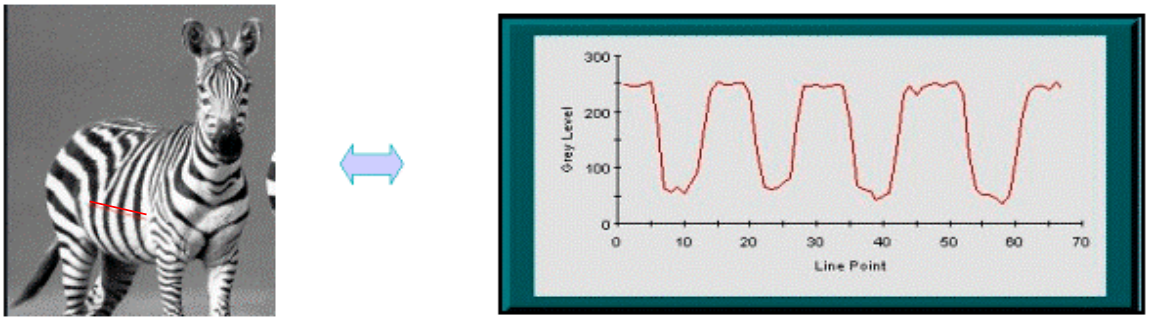


Fig. 10 : Profil de la ligne rouge sur le zèbre.

Dans cette figure, les rayures blanches sont situées très haut sur l'échelle de codage des niveaux de gris : à 250 (soit quasiment la valeur correspondant au blanc: 255). Les rayures noires ne sont pas noires mais gris foncé, car leur valeur de niveau de gris n'est pas 0 (qui correspond au noir absolu) mais environ 50.
Ce genre de profil des niveaux de gris est utile pour appréhender la "structure" des niveaux de gris d'une image avant d'effectuer des traitements dessus.

- **Seuillage binaire**

Lorsque l'on effectue le seuillage d'une image, il s'agit généralement de la rendre binaire. On définit un niveau de gris référence, ensuite on affecte la valeur maximale des niveaux de gris (255) à tous les pixels dont la valeur est supérieure ou égale à la valeur de référence. Les autres pixels prendront la valeur minimale (0).

Nous pouvons voir l'exemple sur la Figure 11 où nous avons seuillé une image au niveau de gris 128.

Fig. 11 : Exemple d'une image seuillée au niveau de gris 128

- **Combinaison d'images**

Le principe est assez simple ; par exemple pour un opérateur «moyenne», la nouvelle valeur du pixel sera la moyenne des deux précédentes. (Fig. 12)

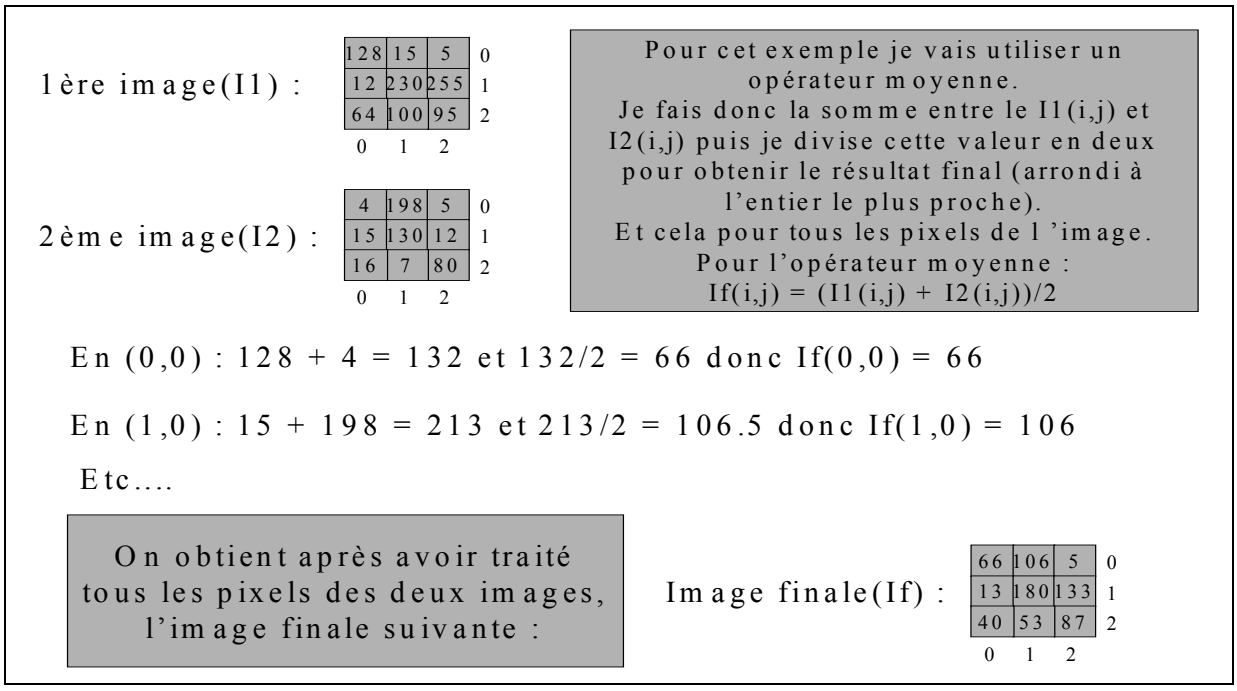


Fig. 12 : Exemple d'une combinaison d'image avec un opérateur «moyenne».

On peut utiliser d'autres opérateurs très courants comme l'opérateur minimum ou maximum qui, au lieu de calculer la moyenne prend soit le minimum des deux valeurs soit le maximum.
(Voir Annexe C pour le résultat sur les images).

- **Opérateurs de voisinage**

Un opérateur de voisinage permet d'affecter la valeur d'un pixel en fonction de celles de ces voisins.

Dans tous les traitements réalisés par la suite, j'ai utilisé un filtre carré de dimension 3x3 centré en (1,1) c'est-à-dire que le pixel du centre du carré aura une nouvelle valeur en fonction de ces 8 voisins adjacents.

→ **L'érosion** consiste à affecter au pixel central le minimum des pixels adjacents. Cette opération permet d'élargir les zones foncées, d'amincir les structures claires larges, de faire disparaître les détails clairs... (Fig. 13).

→ **La dilatation** consiste à affecter au pixel central le maximum des pixels adjacents. Cette opération permet d'élargir les zones claires, d'amincir les structures foncées larges, de

faire disparaître les détails foncés, connecter les parties disjointes proches, combler les trous,... (Fig. 13)
L'érosion et la dilatation consistent en des opérations d'amincissement et d'épaississement des structures.

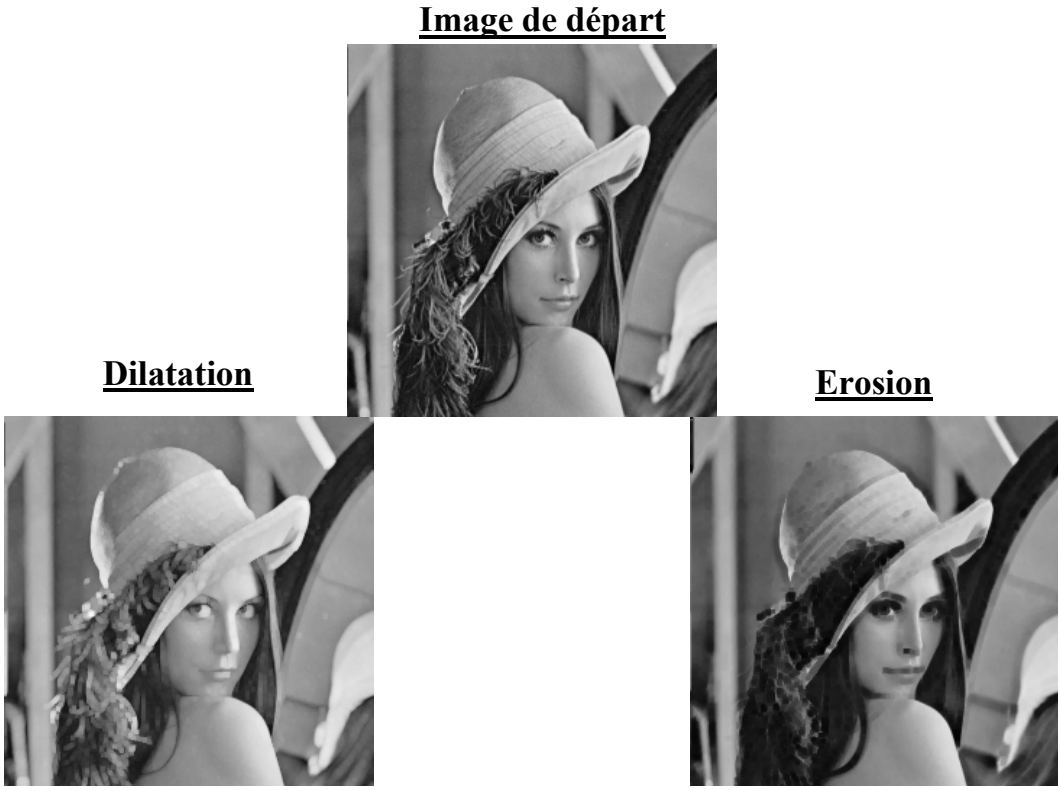


Fig. 13 : Exemple de dilatation et d'érosion sur une image.

→ **L'ouverture** consiste à effectuer sur l'image une érosion suivie d'une dilatation. Une ouverture permet le lissage des contours (adoucit les contours), la suppression de pixels isolés...
La fermeture consiste en une dilatation suivie d'une érosion..

→ **Filtre médian :** Ce filtre est un filtre réducteur des bruits impulsionnels, c'est-à-dire que les pixels aberrants dans une zone. Pour réaliser un filtre médian, il faut prendre toutes les valeurs des pixels du voisinage, les mettre dans un tableau trié puis prendre la valeur au milieu du tableau (Fig. 14).

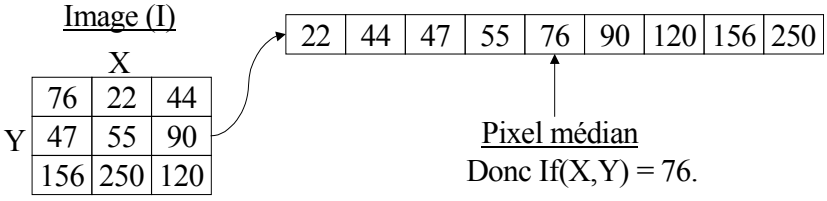


Fig. 14 : Exemple de calcul du pixel médian.

Nous pouvons observer ce que l'on appelle du bruit dans une image dans l'exemple présenté sur la figure 15.

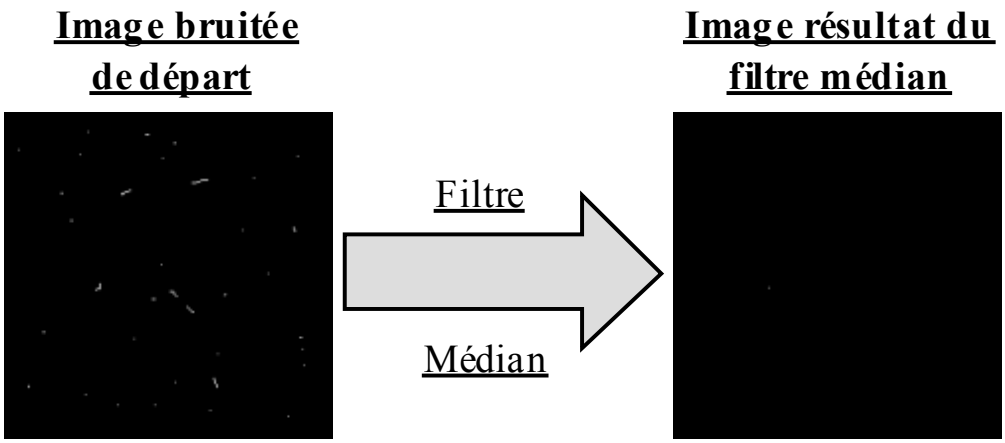


Fig. 15 : Résultat du filtre médian sur une image bruitée

→ Filtre écart-type : Ce film affecte au pixel central du voisinage, l'écart type de tout le voisinage (Fig. 16)

Image d'origine

Y \ X			
	76	22	44
	47	55	90
	156	250	120

Formule écart-type :
Val = Racine((1/N)*(Somme(x²) – Somme(x)))

I(X,Y) = Racine((1/9)*
((76²+22²+44²+47²+55²+90²+156²+250²+120²) –
(76+22+44+47+55+90+156+250+120))
= Racine((1/9)*(122766 – 860))
= 116.38
Au final I(X,Y) = 116 (Arrondi)

Fig. 16 : Explication sur le filtre écart-type.

→ Convolution : On parle de convolution lorsque que l'on affecte des coefficients au filtre. Le filtre s'appelle alors masque de convolution. L'image finale est calculée en déplaçant le masque de convolution sur tous les pixels de l'image initiale. (Fig. 17)

Masque de convolution

2	1	2
0	0	0
-2	-1	-2

Image de départ

Y \ X			
	5	134	53
	12	125	60
	40	100	12

Le pixel à la position (X,Y) vaut maintenant :
5*2+134*1+53*2+40*-2+100*-1+12*-2 = 46

Image finale

Y \ X			
	...	...	...
	...	46	...
	...	...	...

Fig. 17 : Fonctionnement de la convolution

Dans mon travail de détection du film support, j'utilise le masque de Sobel (Fig. 18). En effet celui-ci permet de faire de la détection de contours. Son but est d'amplifier les régions de fortes variations locales d'intensité correspondant aux contours.

Sobel X

1	0	-1
2	0	-2
1	0	-1

Sobel Y

1	2	1
0	0	0
-1	-2	-1

Fig. 18 : Masque de convolution de Sobel suivant l'axe des X et des Y.

Vous trouverez d'autres masques de convolution assez connus dans l'Annexe D.

b) Application des ces outils

- **Positionnement du travail**

Après avoir réalisé le programme de sélection des zones dans une image paramétrique, un autre travail ma été confié. Celui-ci a consisté à détecter le film support présent dans certaines images (Fig. 2). Dans la zone sélectionnée pour le calcul de moyenne, les pixels correspondants au film support donnent des résultats erronés, c'est pourquoi il m'a été demandé de détecter ce film support pour éviter de le prendre en compte dans le calcul de moyenne.

Ce second travail s'est déroulé en deux étapes bien distinctes. Dans un premier temps j'ai développé un programme en C++ qui me permet de détecter le film. J'ai utilisé le langage C++ car celui-ci est beaucoup plus rapide que le Visual Basic pour effectuer les opérations que j'utilisées et donc d'optimiser les traitements. De plus comme certains de ces outils de traitement d'image on été développé durant mon année scolaire, cela me permettait de les utiliser pour réaliser ce travail. Pour pouvoir détecter le film support dans les images paramétriques, j'ai utilisé la plupart des traitements explicités au III.3.a. En effet j'ai énuméré certains d'entre eux car je les ai utilisés dans mes travaux lors de mes essais de détection.

Dans un second temps j'ai adapté l'interface graphique que j'ai réalisée précédemment pour qu'elle puisse prendre en compte le film détecté.

- **Traitements effectués pour la détection du film**

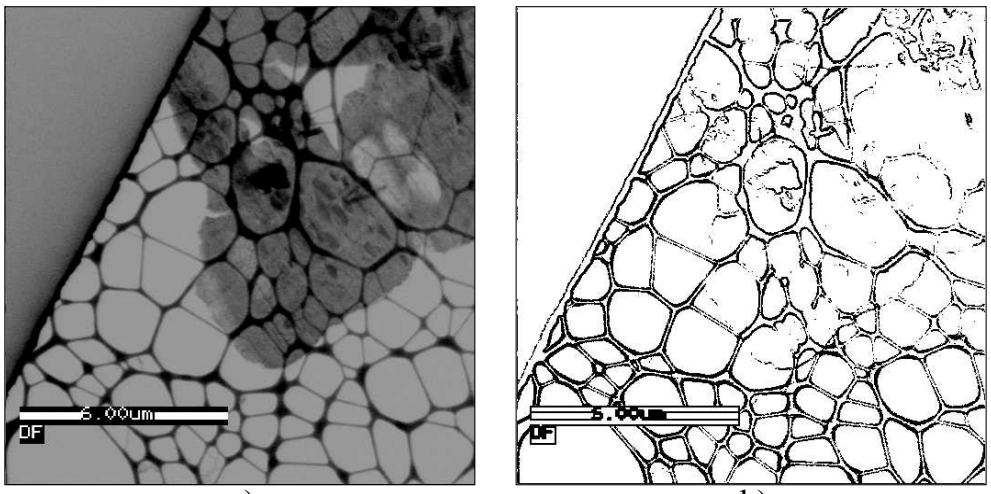
La détection du film consiste en une succession de traitements effectués sur l'image. En fait celle-ci consiste à utiliser des outils de détection de contours.

Pour détecter les contours j'ai appliqué le filtre de gradients de Sobel suivant l'axe des X puis suivant l'axe des Y. J'obtiens donc deux images différentes sur lesquelles j'effectue un seuillage par rapport à la moyenne de l'histogramme c'est à dire tous les niveaux de gris dont le nombre est inférieur à la moyenne de l'histogramme seront dessinés en noir(Fig. 19).

Fig. 19 : Explication sur le seuillage à partir de la moyenne dans l'histogramme.

Puis je recommence ces opérations sur l'image initiale avec une rotation de 180° ce qui me permet de détecter un nombre plus important de contours. Au final j'obtiens donc 4 images binaires que je recombine à l'aide d'un opérateur minimum. (Fig. 20).

Image initiale Image résultat



a) b)

Fig. 20 : Résultat de la détection de contour.

Dans la plupart des cas, j'effectue sur l'image résultat une ouverture suivie d'un filtre médian. L'ouverture relie les contours et le filtre médian permet de débruiter l'image car des pixels n'appartenant pas à un contour peuvent avoir été détecté ce qui permet de les supprimer.

Ensuite je rajoute à cette image (Fig. 20. b), tous les pixels dont le niveau de gris est inférieur au premier niveau de gris est supérieur à la moyenne de l'histogramme de l'image initiale (Fig. 20 a). (Fig. 21)

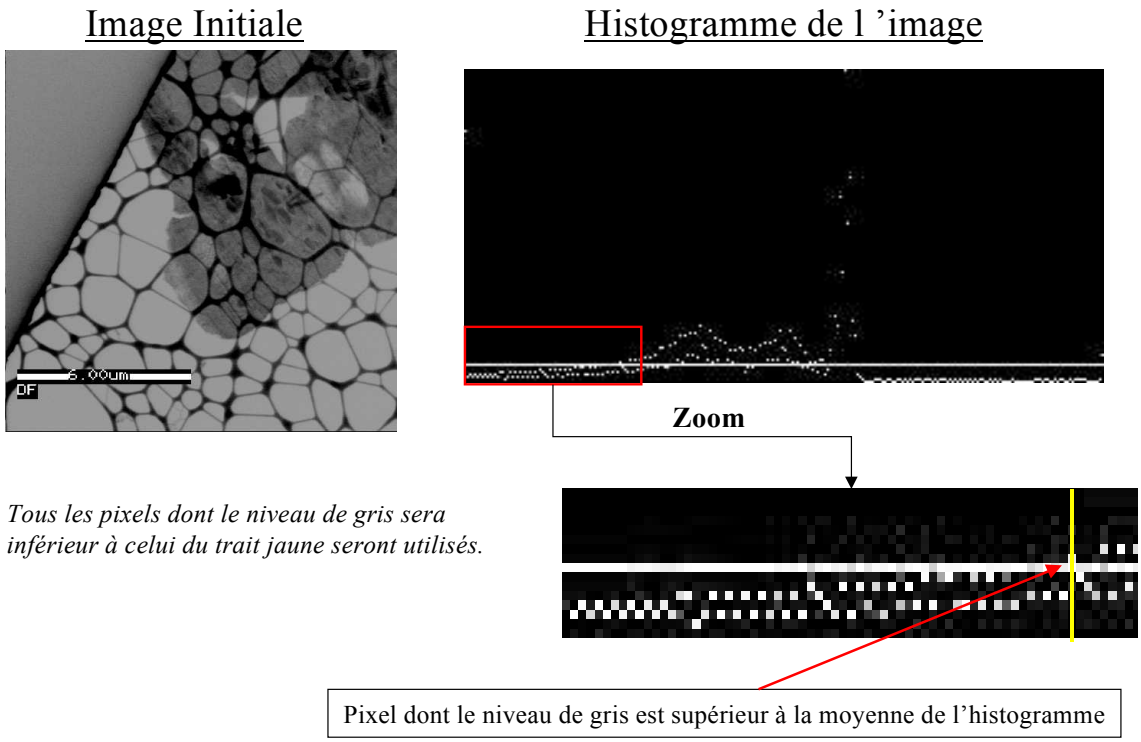


Fig. 21 : Explication sur le rajout de certains pixels de l'image initiale.

Au final en recombinant l'image avec un opérateur minimum de la figure 20 b à celle obtenue avec la méthode de la figure 21, j'obtiens des résultats plutôt satisfaisants. (Fig. 22)

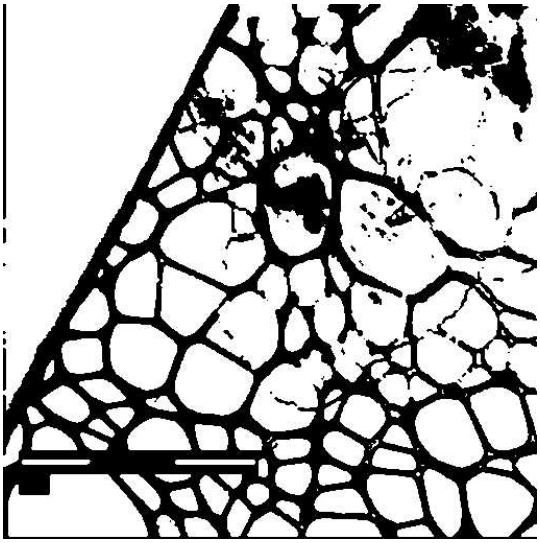


Fig. 22 : Résultat de la détection du film support.

Ces traitements peuvent ne pas être suffisants ou superflus suivant le type d'image traitée, c'est pourquoi, lorsque j'ai adapté le code de l'interface graphique pour qu'il prenne

en compte le film support, j'ai fait en sorte que l'on puisse rajouter ou supprimer certains traitements (Voir point suivant).

- **Adaptation de l'interface graphique**

Après avoir réalisé l'étape de détection du film support, j'ai dû adapter l'interface graphique que j'avais créée précédemment pour que celle-ci prenne en compte les pixels appartenant au film support pour éviter de s'en servir dans le calcul de moyenne.

Cette adaptation n'a pas nécessité beaucoup de modifications, en effet, il a juste fallu rajouter un tableau de la taille de l'image permettant de stocker les pixels à ne pas prendre en compte et adapter un peu la fonction de calcul de moyenne. Il a fallu rajouter un test dans cette fonction pour dire que si le pixel courant fait partie du film alors il ne doit pas être pris en compte. J'ai rajouté aussi une boite de dialogue permettant de sélectionner les traitements à effectuer sur l'image pour la détection du film (Fig. 23).

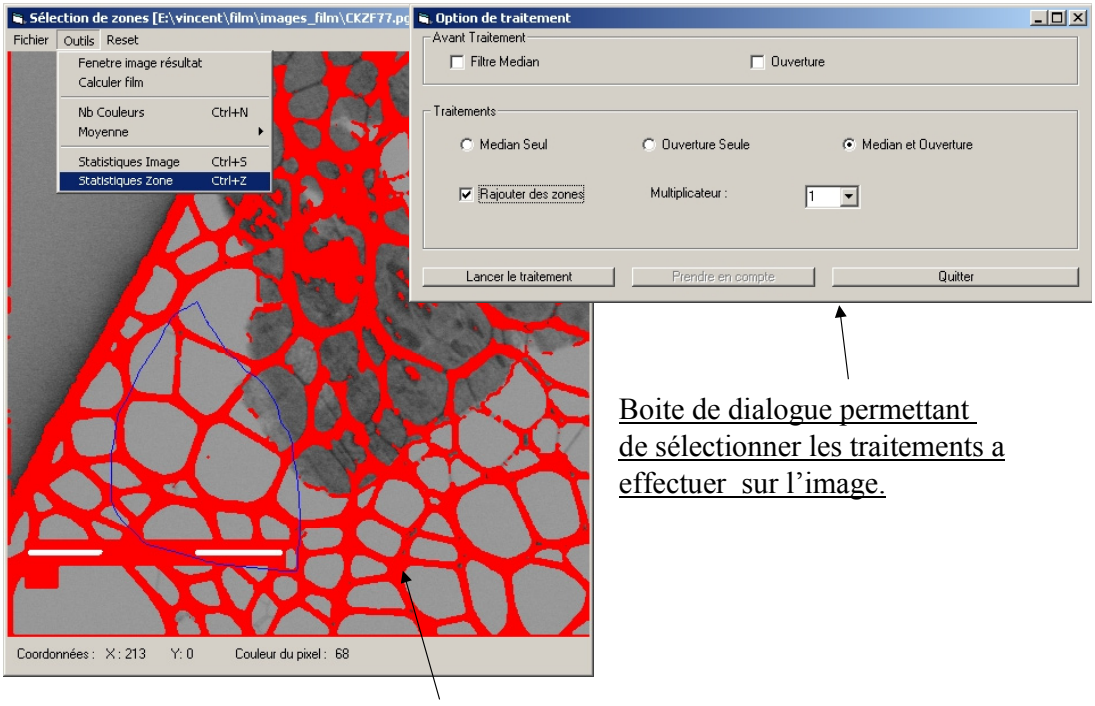


Fig. 23 : Présentation des outils de détection du film.

Un problème est apparu car, en Visual Basic lorsque l'on exécute un programme extérieur avec la fonction «Shell» (ce qui est le cas lorsque j'appelle le programme qui détecte le film ou encore celui qui convertit les images), ceux-ci sont lancés en mode synchrone. Cela signifie que le programme extérieur va être exécuté en même temps que l'application qui l'a lancé. L'idéal serait que le programme exécuté se termine avant que le programme principal passe à l'instruction suivante. En effet, lorsque j'appelle le programme qui détecte le film support, celui-ci crée une image binaire (où le noir représente le film), que je charge ensuite

dans mon tableau qui stocke le film; si le programme essaie de charger l'image qui aurait du être créée avant que celle-ci ne le soit par l'autre programme alors cela produit une erreur. Il faut donc que le programme chargé de créer l'image où est présent le film ait fini son exécution avant que l'application ne charge cette image.
Pour résoudre ce problème j'ai cherché dans la documentation de Visual Basic s'il était possible de lancer un programme et d'attendre la fin de l'exécution de celui-ci. Comme il n'existait pas de fonction permettant de faire cela j'ai effectué des recherches sur Internet et j'ai trouvé une fonction qui permettait de le faire et de résoudre ce problème.

c) Conclusion

Ce travail de détection du film support a été très intéressant de par sa nature, en effet il m'a permis de comprendre un peu mieux le traitement d'images ainsi que sa difficulté. En outre ce travail de détection du film contour m'a pris beaucoup de temps et ne fonctionne pas encore avec certaines images. De plus j'aimerai continuer encore mes études dans le domaine de l'imagerie en faisant, par exemple du traitement d'images; c'est pourquoi ce travail a été très intéressant pour moi.

J'approfondirai cette partie de mon rapport lors de la soutenance orale.

Conclusion

Ce stage a été très bénéfique pour moi car il m'a permis d'acquérir de nouvelles connaissances telles que l'apprentissage du Visual Basic pour créer des programmes visuels conviviaux et interactifs ou encore des méthodes de traitements d'images.

De plus, ce stage dans un laboratoire de recherche m'a permis de visualiser le travail des chercheurs ainsi que des méthodes de travail utilisées pour essayer de comprendre et de guérir certaines maladies.

Il s'agissait en réalité du premier projet que je réalisais seul. En effet, j'ai travaillé seul pendant toute la durée de ce stage. J'ai réalisé ces travaux entièrement selon le cahier des charges précis proposé par le laboratoire, j'ai pris en charge ces travaux de l'analyse jusqu'à la réalisation finale.

Je suis globalement satisfait des résultats finaux qui ont parfaitement su répondre aux attentes du laboratoire, bien que de nombreuses modifications et améliorations soient encore possibles telles que l'optimisation du code et des méthodes de réalisation.

L'apprentissage à l'IUT m'a permis l'acquisition de nombreuses connaissances grâce à un enseignement de qualité, et j'aimerais poursuivre mes études en faisant une licence professionnelle dans le domaine de l'imagerie numérique. J'espère par cette formation acquérir un maximum de connaissances qui me donneront de sérieux atouts lors de mon engagement dans le monde du travail.

(1) : Voir Glossaire page 35

Glossaire

Algorithme de Bresenham : C'est un algorithme (procédure informatique qui solutionne un problème point par point) qui permet de tracer une droite entre deux points dans un espace de dimension 2. Il a été développé par Jack E. Bresenham en 1962.

Anti-spams : Logiciel qui évite de recevoir des spams dans sa boite aux lettres électronique. Un spam est un email (généralement publicitaire) que l'on reçoit dans notre boite aux lettres sans l'avoir demandé.

Boites de dialogue : Petites fenêtres flottantes apparaissant à l'écran. Celles-ci permettent de mettre en relation l'utilisateur et le programme.

DUT : Diplôme Universitaire de Technologie.

Firewall : Qui peut être traduit «mur de feu», «pare-feu». C'est une barrière permettant d'isoler un ordinateur d'un réseau, pour éviter tout piratage.

G.U.I. (Graphic User Interface) : Ils permettent de créer des applications graphiques qui mettent en relation l'homme et la machine.

IUT : Institut Universitaire de Technologie.

Microscopie électronique : La différence avec la microscopie optique, c'est qu'au lieu de traverser les échantillons avec des photons, ceux-ci sont traversés avec des électrons. La principale caractéristique est le fort grandissement (jusqu'à X 1000000) qui permet de travailler sur des échantillons très petits.

Picture Box : Objet de Visual Basic qui permet de pouvoir dessiner des objets ou des images à l'intérieur.

Spectroscopie EELS : Détection de l'énergie perdue par les électrons après la traversée de l'échantillon. Cela permet ici de déterminer la concentration en eau dans l'échantillon biologique étudié en microscopie électronique.

Spectroscopie X : Détection des rayons X émis suite au bombardement électronique d'un échantillon. Cela permet d'obtenir les concentrations ioniques des éléments présents dans l'échantillon biologique étudié en microscopie électronique.

Text Box : Objet de Visual Basic qui permet de saisir ou d'afficher du texte.

Variable : Elément permettant de stocker une valeur.

XnView© : Logiciel de visualisation d'images.

Annexe A : Présentation du Microscope Electronique à Balayage Transmission (MEBT)

Microscope Electronique à Balayage Transmission PHILIPS CM 30

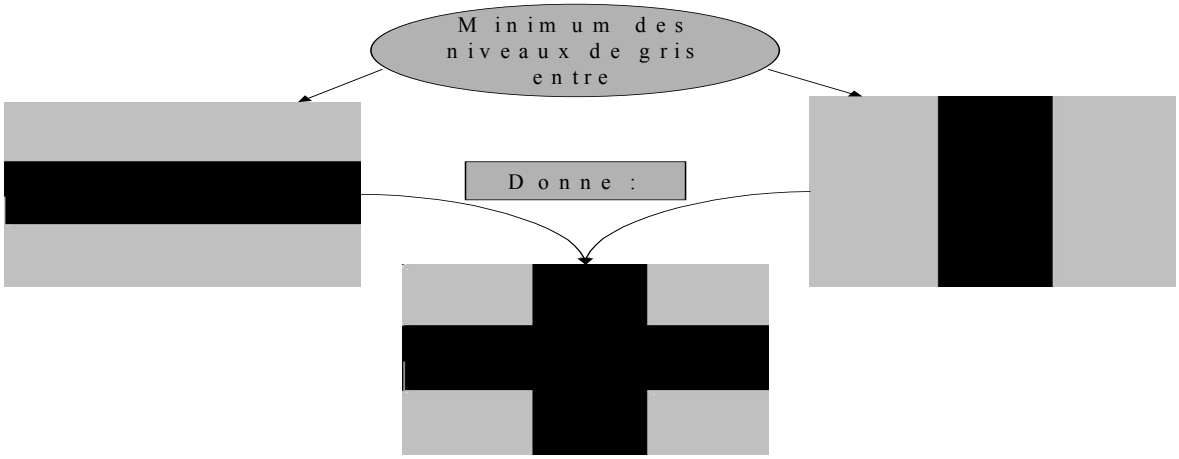
- microscope analytique
- porte-objets
- porte-objets froids
- spectrométrie par sonde électronique d'échantillons minces (épaisseur < µm) par transmission et par balayage.
- diffraction électronique

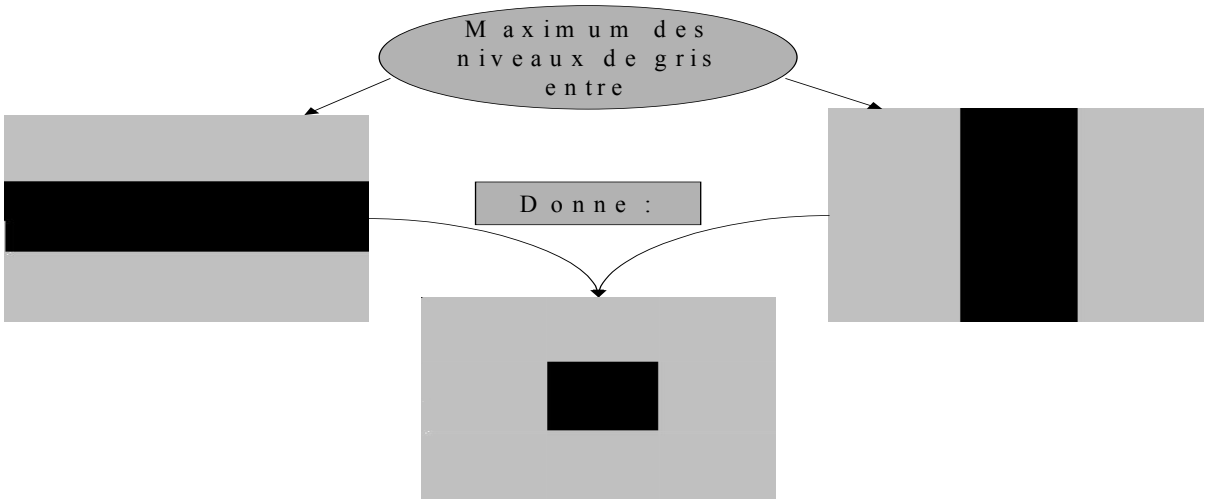
Annexe B : Code du calcul de moyenne

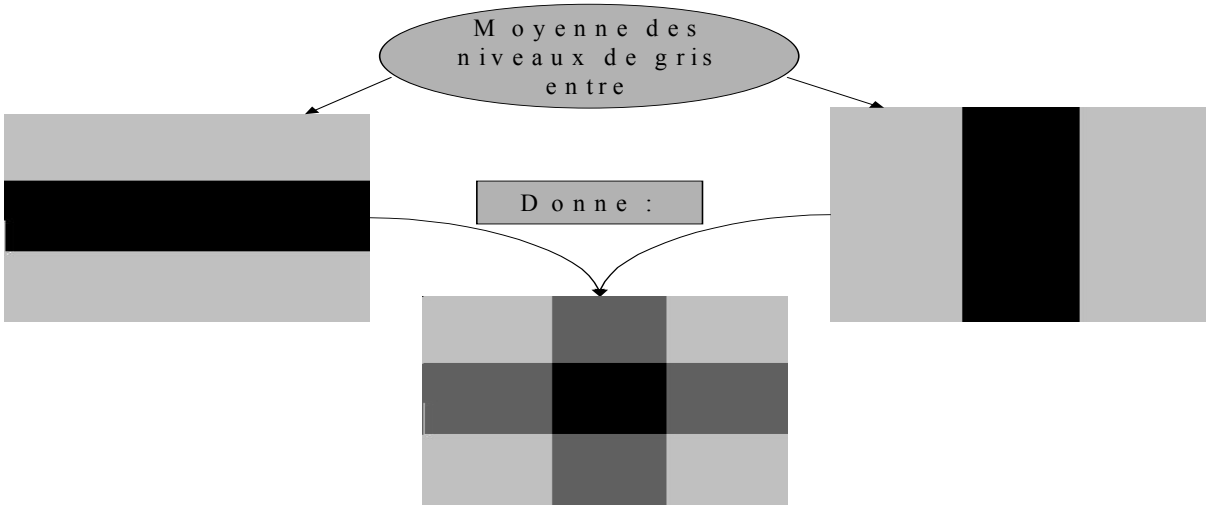
```
Private Function moyzone() As Byte
   Dim X As Single, Y As Single

   Dim somm As Long, nbval As Long 'pour le calcul de moyenne
   somm = 0 'stock la somme des valeurs des pixels
   nbval = 0 'stock le nombre de pixels pris en compte

   Dim pile() As Single 'pile qui stocke les positions où il faut passé
   Dim longpile As Long
   longpile = 2
   ReDim pile(longpile - 1)
   pile(longpile - 2) = ptsx 'on empile la position central a la zone X
   pile(longpile - 1) = ptsy 'on empile la position central a la zone Y

   While longpile <> 0 'tant que la pile n'est pas vide
      X = pile(longpile - 2) 'on depile la valeur X du sommet de la pile
      Y = pile(longpile - 1) 'on depile la valeur Y du sommet de la pile
      longpile = longpile - 2 'la pile est plus petite de deux éléments


       'ici c'est logique mais difficile à expliquer
      If longpile = 0 Then 'ici si la pile est vide
         ReDim Preserve pile(longpile) 'je la redimensionne a 0 élément (car vide)
      Else                 'sinon
         ReDim Preserve pile(longpile - 1) 'je la redimensionne a la longueur-1 car
en vb si la pile est de longueur 1 on peut mettre 2 éléments.
      End If

      If deja(X, Y) = 0 Then 'on vérifie que le pixel ne soit pas déjà pris en compte
         deja(X, Y) = 1 'on signale qu'on la pris en compte

         somm = somm + img(X, Y) 'on rajoute la valeur du pixel
         nbval = nbval + 1 'on dit qu'il y a un pixel de plus qui à été pris en compte
      End If

      'on regarde en (X+1,Y)
      If valide(X + 1, Y) Then 'on vérifie que l'on reste bien dans l'image
         If deja(X + 1, Y) = 0 Then 'que le pixel n'est pas déjà pris en compte
            longpile = longpile + 2 'on augmente la pile de deux valeurs (un pour X
et l'autre pour Y)
            ReDim Preserve pile(longpile - 1) 'on redimensionne la pile en gardant
les valeurs a l'intérieur
            pile(longpile - 2) = X + 1 'on affecte les valeurs ici X
            pile(longpile - 1) = Y 'et la Y
         End If
```

```
            End If

            'on regarde en (X-1,Y)
            If valide(X - 1, Y) Then
               If deja(X - 1, Y) = 0 Then
                  longpile = longpile + 2
                  ReDim Preserve pile(longpile - 1)
                  pile(longpile - 2) = X - 1
                  pile(longpile - 1) = Y
               End If
            End If

            'on regarde en (X,Y-1)
            If valide(X, Y - 1) Then
               If deja(X, Y - 1) = 0 Then
                  longpile = longpile + 2
                  ReDim Preserve pile(longpile - 1)
                  pile(longpile - 2) = X
                  pile(longpile - 1) = Y - 1
               End If
            End If

            'on regarde en (X,Y+1)
            If valide(X, Y + 1) Then
               If deja(X, Y + 1) = 0 Then
                  longpile = longpile + 2
                  ReDim Preserve pile(longpile - 1)
                  pile(longpile - 2) = X
                  pile(longpile - 1) = Y + 1
               End If
            End If

         Wend
         moyzone = somm / nbval 'le résultat est égale a la somme des valeurs divisé par
le nombre de valeur
End Function
```

Annexe C : Exemples de combinaisons d'images







Annexe D : Masques de convolution

- Filtres de gradients (suivant l'axe des X, pour obtenir suivant l 'axe des Y il faut inverser lignes/colonnes)

Prewitt

1	0	-1
1	0	-1
1	0	-1

Sobel

1	0	-1
2	0	-2
1	0	-1

Croix

0	0	0
1	0	-1
0	0	0

- Filtres Laplaciens

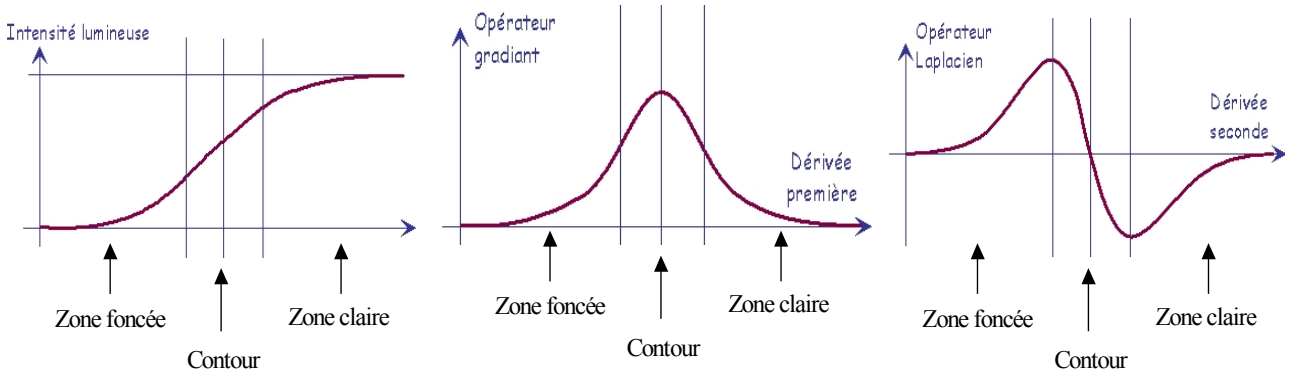
Carré

0	1	0
1	-4	1
0	1	0

Octogonal

1	1	1
1	-8	1
1	1	1

Ces filtres ont pour but d'augmenter le contraste et de mettre en évidence les contours.
On définit un contour comme une discontinuité locale de l'intensité lumineuse.
Les techniques permettant de détecter un contour sont basées sur l'utilisation:

- de filtres gradients,
- de filtres Laplaciens.

Les contours sont les lignes des maximums locaux pour les filtres de gradient.
Pour les filtres Laplaciens, les contours seront les suites de points à 0.