Travail de fin d'études

« Réalisation d'un outil adaptatif de comptage de cellules »

Carlozzi Nicolas, 2 MINF

Date : 17 août 2009

Jury : Professeur Marc Van Droogenbroeck (co-promoteur)
      Professeur Louis Wehenkel (co-promoteur)
      Professeur Jacques Verly
      Raphaël Marée (GIGA)
Résumé

Dans le cadre de recherches contre le cancer notamment, les biologistes tentent d’évaluer l’effet de diverses molécules sur des processus biologiques tels que l’angiogenèse\(^1\). Plus le nombre de cellules est élevé et moins la molécule testée inhibe l’angiogenèse. Une partie du travail consiste donc à compter le nombre de cellules endothéliales\(^2\) sur des plaques circulaires transparentes, comme l’illustrent les images ci-dessous :

![Images illustrant le comptage de cellules endothéliales](image1.jpg) ![Images illustrant le comptage de cellules endothéliales](image2.jpg) ![Images illustrant le comptage de cellules endothéliales](image3.jpg)

Ce comptage est jusqu’à présent effectué manuellement. C’est là une tâche longue, répétitive, fastidieuse et dépendante de l’opérateur humain. Le but de ce travail est d’étudier la possibilité d’automatiser cette tâche autant que possible avec un maximum de précision.

Pour ce faire, le programme développé utilisera des outils de traitement d’images combinés à des méthodes d’apprentissage automatique. Le comptage sera aussi automatisé que possible mais nous laisserons, in fine, la possibilité à l’utilisateur de corriger les erreurs éventuelles.

Le présent rapport débute par une introduction sur le contexte général de travail, suivi d’une analyse des images à traiter. Ensuite, nous consacrerons une vaste partie à l’étude des domaines étudiés et à l’état de l’art. Enfin, nous détaillerons le processus de développement de l’outil de comptage automatique ainsi que les différents résultats obtenus.

---

\(^1\) Création de nouveaux vaisseaux sanguins par des tumeurs cancéreuses afin d’assurer leur alimentation.

\(^2\) Cellules qui forment les vaisseaux.
Remerciements

J'exprime toute ma gratitude à Monsieur Marc Van Droogenbroeck, Monsieur Louis Wehenkel et Monsieur Raphaël Marée pour m'avoir permis de réaliser ce travail de fin d'études et accordé leur confiance.

Plus particulièrement, je tiens à remercier à nouveau Monsieur Raphaël Marée pour le temps qu'il m'a consacré, sa disponibilité exemplaire et surtout pour les conseils avisés et les perspectives intéressantes qu'il a donnés à mon travail. En ce sens, je tiens également à remercier Sébastien Piéard pour ses remarques pertinentes.

Je remercie également l'ensemble des professeurs qui ont participé à ma formation universitaire et l'équipe du GIGA qui m'a aidé à mener à bien ce travail. En particulier, Geoffrey Gloire, Pascal Hubert et Julie Halkein pour les images et autres documents pratiques qu'ils m'ont fournis.

Mes remerciements vont aussi à ma famille, plus précisément à mes parents qui m'ont permis de mener des études dans des conditions optimales. Je leur serai toujours reconnaissant de l'encadrement qu'ils m'ont apporté. En ce sens, je tiens aussi à remercier Alicia pour son soutien et ses encouragements.

Enfin, c'est avec un réel plaisir que j'adresse mes remerciements à toutes les personnes ayant contribué, de près ou de loin, à la rédaction de ce mémoire.
Table des matières

Chapitre 1 - Introduction ............................................................................................................. 10
  1.1 Contexte général ............................................................................................................... 10
    1.1.1 Le GIGA, recherche et défi .................................................................................... 10
    1.1.2 Processus d’acquisition ......................................................................................... 11

Chapitre 2 - Analyse des images ................................................................................................. 13
  2.1 Introduction ..................................................................................................................... 13
    2.1.1 Algorithme naïf de segmentation de couleur ..................................................... 13
  2.2 Problématique des images ............................................................................................... 15
    2.2.1 Cas favorables ......................................................................................................... 15
    2.2.2 Cas défavorables .................................................................................................... 16
    2.2.3 Cas extrêmes ......................................................................................................... 21
  2.3 Conclusion ....................................................................................................................... 22

Chapitre 3 - Domaines étudiés ................................................................................................. 23
  3.1 Préambule ....................................................................................................................... 23
  3.2 Apprentissage automatique ............................................................................................ 23
    3.2.1 Apprentissage supervisé ....................................................................................... 24
  3.3 Classification automatique d’images .............................................................................. 25
    3.3.1 Les arbres de décision ......................................................................................... 25
    3.3.2 Ensembles d’arbres ............................................................................................... 28
    3.3.3 Pixit ....................................................................................................................... 29
  3.4 Traitement numérique des images ................................................................................... 32
    3.4.1 Pré-requis ............................................................................................................... 32
    3.4.2 Espaces colorimétrique ......................................................................................... 34
    3.4.3 Les filtres .............................................................................................................. 36
    3.4.4 Segmentation ......................................................................................................... 40
Chapitre 4 - État de l’art

4.1 Introduction

4.2 Recherche bibliographique

4.3 Thèse d’E. Glory

4.3.1 Evaluation de la qualité d’une segmentation d’images couleur

4.3.3 La segmentation d’images cytologiques en couleur

4.3.4 La classification des composantes connexes

4.3.5 La segmentation des agrégats

4.3.6 Conclusions E.Glory

Chapitre 5 - Développement de l’outil de comptage automatique

5.1 Introduction

5.1.1 Choix d’implémentation

5.2 Stratégie générale

5.3 Segmentation initiale

5.3.1 Seuillage manuel

5.3.2 Seuillage automatique

5.3.3 Choix d’une composante de couleur

5.3.4 Binarisation de l’image

5.4 Classification automatique des composantes connexes

5.4.1 Phase d’apprentissage

5.4.2 Phase de prédiction

5.5 Segmentation des agrégats

5.5.1 Technique développée

5.6 Phase d’édition - corrections
Chapitre 6 - Tests et évaluations .................................................................................. 81

6.1 Introduction........................................................................................................... 81
6.2 Mesures................................................................................................................ 81
6.3 Résultats et interprétations.................................................................................. 84

1. Expérience IK9-8M............................................................................................... 85
2. Expérience NT24HA.............................................................................................. 86
3. Expérience svl2-8M............................................................................................. 87
4. Expérience IK9-6M............................................................................................... 88
5. Expérience « agrégats »...................................................................................... 89
6. Expérience « coloration ».................................................................................... 91

6.4 Temps d’exécution et complexité ......................................................................... 94

Chapitre 7 - Conclusion ............................................................................................... 95

7.1 Résumé................................................................................................................ 95
7.2 Perspectives / améliorations............................................................................... 96
# Table des figures

## Chapitre 1 : Introduction

- Figure 1.1-1 : Acquisition des images couleur .......................................................... 11
- Figure 1.1-2 : Technique des chambres de Boyden ................................................ 11

## Chapitre 2 : Analyse des images

- Figure 2.1-1 : Modélisation de l'espace RGB sous la forme d'un cube .......................... 14
- Figure 2.2-1 : Cas favorable pour le comptage ............................................................ 15
- Figure 2.2-2 : Cas favorables : résultats et taux d'erreur ........................................ 15
- Figure 2.2-3 : Image originale d’agrégat ................................................................. 16
- Figure 2.2-4 : Image Binaire obtenue après segmentation ....................................... 16
- Figure 2.2-5 : Exemple d’une seule composante connexe ! .................................. 16
- Figure 2.2-6 : Intervalle de circularité : 0.00 - 1.00 ..................................................... 17
- Figure 2.2-7 : Intervalle de circularité : 0.40 - 1.00 ..................................................... 17
- Figure 2.2-8 : Pas de distinction entre cytoplasme et noyau .................................. 18
- Figure 2.2-9 : Comptage erroné des pores (flèches vertes) ..................................... 19
- Figure 2.2-10 : Les flèches vertes indiquent les taches interprétées comme cellules .... 19
- Figure 2.2-11 : Exemples d’artefacts ..................................................................... 20
- Figure 2.2-12 : Les cellules profondes dans le gel sont ignorées lors du comptage .... 20
- Figure 2.2-13 : Image floue impliquant une composante connexe trop grande .......... 21
- Figure 2.2-14 : Cas extrêmes pour la segmentation .............................................. 21

## Chapitre 3 : Domaines étudiés

- Figure 3.2-1 : Apprentissage supervisé ................................................................... 24
- Figure 3.3-1 : Induction d'un arbre de décision à partir d'un ensemble d'apprentissage .... 26
- Figure 3.3-2 : Sur et Sous - apprentissage .............................................................. 28
- Figure 3.3-3 : Extraction de sous-fenêtres, redimensionnement et étiquetage .......... 30
- Figure 3.3-4 : Construction d'un ensemble de T arbres aléatoires ......................... 30
- Figure 3.3-5 : Phase de prédiction de la méthode «PixIT» ..................................... 31
Figure 3.4-1 : Spectre lumineux (couleurs visibles par l’œil humain)........................................................................32
Figure 3.4-2 : Histogramme d'une image en niveaux de gris..................................................................................33
Figure 3.4-3 : Espace RGB et synthèse additive.........................................................................................................34
Figure 3.4-4 : Espace HSI. ..................................................................................................................................................35
Figure 3.4-5 : Différentes formes d’éléments structurant symétriques........................................................................38
Figure 3.4-6 : Dilatation morphologique......................................................................................................................38
Figure 3.4-7 : Erosion morphologique. ........................................................................................................................39
Figure 3.4-8 : Dilatation d’une image en niveaux de gris (exemple 1). ........................................................................39
Figure 3.4-9 : Résultat du seuillage de Ridler................................................................................................................41
Figure 3.4-10 : Relief du gradient de l'image (bassins versants, minima et LPE).........................................................41

Chapitre 4 : Etat de l'art

Figure 4.3-1 : Evaluation des critères de Liu et Borsotti. ..........................................................................................47
Figure 4.3-3 : Critère C : Choix automatique de l'espace de couleur. ......................................................................50
Figure 4.3-4 : Critère C : Choix automatique du seuil de segmentation....................................................................50
Figure 4.3-5 : Schéma général de l’algorithme de segmentation des agrégats. .........................................................54
Figure : 4.3-6 : Comparaison de surface entre la composante et l'ellipse.................................................................55
Figure 4.3-7 : Algorithme de fusion..........................................................................................................................56

Chapitre 5 : Développement de l'outil de comptage automatique

Figure 5.2-1 : Chaîne de traitement des images. ........................................................................................................59
Figure 5.2-2 : Stratégie générale pour le dénombrement de cellules. .......................................................................59
Figure 5.3-1 : Binarisation de l'image originale. ........................................................................................................61
Figure 5.3-2 : Evaluations de différents algorithmes de seuillage automatique. .....................................................62
Figure 5.3-3 : Sensibilité des cônes aux longueurs d'ondes lumineuses.................................................................63
Figure 5.3-4 : Représentation d’une image couleur dans............................................................................................65
Figure 5.3-5 : Elément structurant du filtre médian. (Rayon = 5pixels).................................................................67
Figure 5.3-6 : A gauche l'image originale. A droite l'histogramme...........................................................................68
Figure 5.4-1 : Matrice de confusion après validation croisée..................................................................................70
Figure 5.5-1 : Forme et taille diverses des agrégats. ...............................................................................................72
Figure 5.5-2 : Sur-segmentation de la LPE classique ................................................................. 73
Figure 5.5-3 : Agrégat comportant 10 noyaux cellulaires ............................................................ 74
Figure 5.5-4 : Agrégat nettoyé ........................................................................................................... 74
Figure 5.5-5 : Application du filtre maximum sur l’agrégat ............................................................ 74
Figure 5.5-6 : Enveloppe maximale de P ............................................................................................ 75
Figure 5.5-7 : Marqueurs utilisés pour la LPE .................................................................................. 76
Figure 5.5-8 : A droite : Carte des Distances Euclidiennes + Marqueurs ........................................ 76
Figure 5.5-9 : Chaîne de traitement du comptage des agrégats .......................................................... 76
Figure 5.5-10 : Exemple du problème du cytoplasme pour le comptage par surface ....................... 78
Figure 5.5-11 : Surévaluation du nombre de noyaux dans l’agrégat .................................................. 78
Figure 5.6-1 : Interface graphique de l’outil d’édition/correction .................................................... 79

Chapitre 6 : Tests et évaluations

Figure 6.2-1 : Coefficient de Pearson (Expérience IK9-8M, cf. section 6.3) ........................................ 83
Figure 6.3-1 : Expérience IK9-8M .................................................................................................... 85
Figure 6.3-2 : Expérience NT24HA ................................................................................................ 86
Figure 6.3-3 : Expérience svl2-8M .................................................................................................. 87
Figure 6.3-4 : Expérience IK9-6M .................................................................................................... 88
Figure 6.3-5 : Expérience "Agrégats" ............................................................................................... 89
Figure 6.3-6 : Exemple de différentes colorations des images .......................................................... 91
Figure 6.3-7 : Erreur PCA : Image originale ..................................................................................... 93
Figure 6.3-8 : Erreur PCA : Image binaire ......................................................................................... 93
CHAPITRE 1

INTRODUCTION

1.1 CONTEXTE GÉNÉRAL

1.1.1 LE GIGA, RECHERCHE ET DÉFI

Ce travail est proposé dans le cadre d'une collaboration avec les équipes du GIGA\(^3\) et co-
encadré avec l’équipe du Professeur Marc Van Droogenbroeck\(^4\) et du Professeur Louis
Wehenkel\(^5\).

Le GIGA (Groupe Interdisciplinaire de Génoprotéomique Appliquée) est un centre de
recherche actif dans le monde médical et scientifique. Un des principaux pôles d’intérêt est la
recherche contre le cancer. En ce sens, le GIGA-cancer, une des six unités de recherche
thématicque du GIGA, étudie notamment les mécanismes moléculaires impliqués au cours de
l’angiogenèse.

De nombreux projets sont actuellement en cours et plusieurs s’intéressent
particulièrement au processus de l’angiogenèse que l’on retrouve notamment lors du
développement embryonnaire, mais aussi lors de la croissance des tumeurs malignes et du
développement des métastases. Les chercheurs tentent de suivre de près les inhibiteurs
préalablement définis de l’angiogenèse et s’efforcent d’en identifier de nouveaux dans le but de
mettre au point des traitements. Ils évaluent les mécanismes moléculaires qui sont à l’origine de
l’angiogenèse et de la lymphangiogenèse (comme le cancer).

Le sujet de ce mémoire intervient donc au niveau des résultats, analyses et conclusions
que les scientifiques tentent d’établir lors de diverses expériences. L’information du
dénombrement de cellules permet, entre autre, d’évaluer l’effet d’une molécule sur un processus
biologique.

\(^3\) http://www.giga.ulg.ac.be
\(^4\) http://www2.ulg.ac.be/telecom
\(^5\) http://www.montefiore.ulg.ac.be/~lwh/
1.1.2 PROCESSUS D’ACQUISITION

Le processus d’acquisition des images est schématisé par la figure suivante :

![Figure 1.1-1 : Acquisition des images couleur.](http://www.123bio.net/revues/jchabry/6d.html)

1.1.2.1 TECHNIQUE DES CHAMBRES DE BOYDEN

Les images traitées sont obtenues à partir de la technique des chambres de Boyden. Cette technique permet de mesurer la capacité de migration des cellules à travers un filtre. Deux compartiments sont séparés par une membrane microporeuse (le filtre).

![Figure 1.1-2 : Technique des chambres de Boyden.](http://www.123bio.net/revues/jchabry/6d.html)

En général, les cellules sont placées dans le compartiment supérieur et sont autorisées à migrer à travers les pores de la membrane dans le compartiment inférieur, où des agents chimiotactiques sont présents. Après un certain temps d’incubation, la membrane entre les deux compartiments est fixée et colorée, et le nombre de cellules qui ont migré vers la partie inférieure de la membrane est déterminé.
1.1.2.2 FIXATION ET COLORATION

La procédure de fixation et coloration permet :

✓ d'augmenter la visibilité des cellules,
✓ d'accentuer les particularités morphologiques des cellules,
✓ de conserver l’échantillon en vue d’expériences ultérieures.

La technique de fixation permet de stopper le processus de dégradation des cellules en les imprégnant de certaines substances. Cette étape est délicate car la structure des objets en est plus ou moins affectée. La procédure de fixation est donc source potentielle de futurs artefacts sur les images analysées.

Après la fixation, vient l’étape de coloration qui met en évidence certaines structures afin d'en faciliter l'étude. Dans notre cas, les scientifiques utilisent la coloration au « Giemsa », colorant constitué d’un mélange de deux autres colorants :

➢ Bleu de méthylène
➢ Eosine

Le « Giemsa » donne une coloration dans les tons violet, rose violacé, voire bleu, selon le dosage des deux colorants.

---

6 La fixation rend inactifs les enzymes qui peuvent détruire la morphologie cellulaire et durcit les structures pour qu'elles ne se modifient pas durant la coloration et les observations.
CHAPITRE 2

ANALYSE DES IMAGES

2.1 INTRODUCTION

Dans le but d'adopter la meilleure stratégie possible pour un comptage automatique, il est primordial de commencer par étudier les caractéristiques des images à analyser. L'objet de ce chapitre consistera dès lors à détailler les particularités des images proposées ainsi que les différents problèmes de comptage qui en découlent. Nous nous rendrons rapidement compte que le sujet de ce travail n'est en rien trivial !

Afin de mettre en évidence les difficultés du comptage et les erreurs typiques d'une approche simple de la solution de comptage automatique, nous avons utilisé un plugin ImageJ\(^7\) basé sur un algorithme de segmentation de couleurs. Vu sa simplicité, ce plugin se révèle être un outil rapide et intéressant pour exposer les problèmes récurrents rencontrés.

Cette analyse permettra surtout aux chercheurs de déterminer dans quelle mesure ils peuvent limiter les sources d'erreurs (contexte d'acquisition, réglages microscope, luminosité, taches, artefacts,...) et trouver des solutions d'amélioration de la qualité des images fournies. Par conséquent, et dans une perspective éventuelle d'un outil de comptage spécialisé à un seul type d'images, il sera possible de mieux évaluer les éléments paramétrables et les éléments pour lesquels l'outil doit s'adapter.

2.1.1 ALGORITHME NAÏF DE SEGMENTATION DE COULEUR

Les images sont traitées dans l'espace de couleur RGB. Des seuils minimaux et maximaux sont fixés pour chaque composante (R, G, B).

Voici les valeurs par défaut du programme :

<table>
<thead>
<tr>
<th>Seuils minimaux</th>
<th>Seuils maximaux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>Red</td>
</tr>
<tr>
<td>Green</td>
<td>Green</td>
</tr>
<tr>
<td>Blue</td>
<td>Blue</td>
</tr>
</tbody>
</table>

\[^7\]http://rsb.info.nih.gov/ij/
Chapitre 2 – Analyse des images

Remarque : La composante verte maximale autorisée est inférieure aux deux autres composantes vu que la coloration des cellules tend vers le violet.

Figure 2.1-1 : Modélisation de l’espace RGB sous la forme d’un cube.

- La première étape de l’algorithme consiste à créer une image binaire sur base de l’image traitée. Chaque pixel de l’image est testé selon ses composantes RGB. Si une des valeurs des 3 composantes du pixel ne se trouve pas dans l’intervalle correspondant, alors le pixel de l’image binaire prend la valeur 0 (noir), sinon il prend la valeur 255 (blanc).

- Une opération morphologique est alors appliquée à l’image binaire : l’ouverture\(^8\).

- Pour terminer, on utilise la commande « Analyze Particles » du logiciel ImageJ. Cette commande permet de compter et mesurer les composantes connexes d’une image binaire\(^9\). Ces dernières sont prises en considération à condition de respecter 2 critères :
  a. Un intervalle de taille des composantes détectées (nombre de pixels)
  b. Un intervalle de circularité (formule :\(\text{circularité} = \frac{4\pi \times \frac{\text{Surface}}{\text{Périmètre}^2}}{2}\)). Cet intervalle varie de 0.00 à 1.00. Une valeur de 1.00 indiquant un cercle parfait.

Toute composante ayant une taille ou une circularité en dehors des intervalles définis est ignorée. Par défaut, nous avons déterminé empiriquement un intervalle de taille de \([100..\infty]\), ce qui signifie que seules les composantes dont l’aire est inférieure à 100 pixels sont rejetées. De même, nous avons fixé l’intervalle de circularité à \([0.10..1]\) afin de ne pas rejeter trop d’agrégats.

\(^8\) L’ouverture est la composition de deux opérations morphologiques élémentaires : une érosion suivie d’une dilatation.
\(^9\) L’algorithme utilisé est décrit à cette adresse : http://rsbweb.nih.gov/ij/docs/menus/analyze.html#ap
Chapitre 2 – Analyse des images

2.2 PROBLÉMATIQUE DES IMAGES

2.2.1 CAS FAVORABLES

Les cas favorables décrivent une situation idéale où toutes les images présenteraient un contraste net de couleur entre les cellules et l’arrière plan. Les parasites de l’image (pores, artefacts, taches,...) seraient très peu présents et les agrégats de cellules quasi inexistants. La qualité de la segmentation est d’autant plus grande que les couleurs des objets d’intérêt contrastent avec le reste de l’image.

Exemple :

Figure 2.2-1 : Cas favorable pour le comptage.

Afin de donner un premier aperçu de la précision de l’algorithme que nous venons de présenter, nous avons choisi manuellement 5 images que nous jugeons être des cas favorables et, comme en témoigne le tableau ci-dessous, la segmentation décrite précédemment donne un taux d’erreur moyen inférieur à 4%, ce qui est très satisfaisant.

<table>
<thead>
<tr>
<th>Nom de l’image</th>
<th>Comptage manuel</th>
<th>Comptage automatique</th>
<th>Taux d’erreur (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2102008.jpg</td>
<td>94</td>
<td>89</td>
<td>5,319</td>
</tr>
<tr>
<td>101008.jpg</td>
<td>134</td>
<td>123</td>
<td>8,209</td>
</tr>
<tr>
<td>120908.jpg</td>
<td>151</td>
<td>147</td>
<td>2,649</td>
</tr>
<tr>
<td>24h.jpg</td>
<td>386</td>
<td>380</td>
<td>1,554</td>
</tr>
<tr>
<td>IK9-10-6M(2).jpg</td>
<td>449</td>
<td>450</td>
<td>0,223</td>
</tr>
<tr>
<td><strong>Taux d’erreur moyen</strong></td>
<td><strong>3,591</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.2-2 : Cas favorables : résultats et taux d’erreur.

Malheureusement, les cas favorables, sans être exceptionnels, ne sont pas les cas les plus couramment rencontrés. Comme nous allons le voir dans la section suivante, une bonne partie des images présentent des agrégats et artefacts qui rendent le comptage automatique plus délicat. Cependant, on retrouve souvent des parties de l’image avec des caractéristiques favorables.
2.2.2 CAS DÉFAVORABLES

2.2.2.1 PROBLÈMES LIÉS À L’APPROCHE NAÏVE DU PROGRAMME

2.1.1.1 AGRÉGATS

Le problème des agrégats est sans doute le plus difficile et le plus important à résoudre car il est la source principale d’erreurs lorsque l’on compare les résultats d’un comptage manuel avec ceux d’un comptage automatique.

Nous allons ici présenter un exemple typique de mauvaise interprétation du programme sur des images comportant de nombreux agrégats de cellules.

Afin de comprendre pourquoi les agrégats posent problème, il est pertinent de s’intéresser à l’image binaire obtenue après la phase de segmentation de couleur. En voici une illustration :

![Figure 2.2-3 : Image originale d’agrégat.](image1.png)  
![Figure 2.2-4 : Image Binaire obtenue après segmentation.](image2.png)

Avec une telle image binaire, le rassemblement des composantes connexes donne une interprétation complètement erronée, comme le montre l’exemple ci-dessous :

![Figure 2.2-5 : Exemple d’une seule composante connexe.](image3.png)
Chapitre 2 – Analyse des images

Lors du comptage, la composante connexe rouge sera comptée comme une seule cellule, ce qui est évidemment inadmissible ! Limiter la taille des composantes connexes à compter n’est pas une solution car trop d’agrégats seraient alors ignorés !

Une solution possible à ce problème serait d’effectuer un comptage de surface, c.-à-d. calculer le rapport entre la surface totale des particules mesurées et la surface estimée d’une cellule. Le problème majeur étant évidemment l’estimation de la surface d’une cellule pour des images telles que la figure 4 ! Néanmoins, cette comparaison sera proposée dans le chapitre 5, à la section 5.5.1.3.

2.1.1.1.2 CIRCULARITÉ

L’intervalle de circularité qu’il faut utiliser avec la commande « Analyze Particles » s’avère contraignant. En effet, la forme d’une cellule variant d’une image à l’autre, et parfois même d’une cellule à l’autre sur la même image, fixer un seuil de circularité a comme conséquence fâcheuse d’ignorer les cellules « pas assez rondes ».

2.1.1.1.3 CYTOPLASME – NOYAX

Ce problème est essentiellement présent sur des images sombres où le contraste entre les cellules et l’arrière plan est peu marqué. Le cytoplasme apparaît plus sombre et, là où l’œil humain le distingue du noyau, la segmentation de couleur ne fait pas la différence et nous revenons alors à un problème d’agrégats comme le montre la photo suivante :
Les cercles verts indiquent les noyaux tandis que les cercles rouges indiquent les noyaux + le cytoplasme. Même si l'image présente une densité forte de cellules, on remarque que c'est à cause du cytoplasme que l'image binaire contient de si grandes composantes connexes. Ce problème est évidemment dû aux seuils de segmentation fixés manuellement. Bien entendu, nous pourrions adapter ces seuils pour chaque image et obtenir ainsi de très bons résultats mais cela engendrerait une perte de temps considérable et enlèverait tout intérêt à ce travail d'automatisation.

2.2.2.2 PROBLÈMES LIÉS AU CONTEXTE D'ACQUISITION

2.1.1.4 PORES

Le gel, à travers lequel les cellules migrent, comporte de nombreux pores qui sont parfois considérés comme des cellules. Pour les images bien contrastées, le problème est négligeable. Cependant, pour les images sombres, certains pixels, appartenant à un pore, se situent dans l'intervalle de couleur autorisé. Comme nous le constatons sur la figure suivante, il est même difficile de distinguer si les composantes sont des noyaux ou des pores tachés.
Chapitre 2 – Analyse des images

Figure 2.2-9 : Comptage erroné des pores (flèches vertes).

2.1.1.5 TACHES – ARTEFACTS – PARASITES

En plus des pores, il n'est pas rare qu'une image comporte des taches de coloration, artefacts ou autres parasites. Ce sont néanmoins les taches qui perturbent le plus le comptage puisqu'elles sont colorées et donc interprétées comme cellule. Néanmoins, n'importe quel parasite est susceptible d'être mal interprété. Voici un exemple :

Figure 2.2-10 : Les flèches vertes indiquent les taches interprétées comme cellules.

Ci-dessous, une série d'exemples présentant différents parasites qu'il est possible de rencontrer sur les images et qui peuvent perturber le comptage :
Chapitre 2 – Analyse des images

2.1.1.6 PROFONDEUR DES CELLULES DANS LE GEL

Il arrive que certaines cellules soient plongées plus profondément dans le gel que les autres. Ces cellules sont alors très peu colorées (trop claires) et donc confondues avec l’arrière plan. Notons que sur l’image suivante, les seules particules détectées sont des pores.

Figure 2.2-11 : Exemples d’artefacts.

Figure 2.2-12 : Les cellules profondes dans le gel sont ignorées lors du comptage.
Chapitre 2 – Analyse des images

2.1.1.7 FLOU – FOCUS

Nous l’avons déjà remarqué sur plusieurs exemples, bon nombres d’images traitées sont floues. Un focus adapté lors de chaque acquisition améliorerait sensiblement le contraste et, par la même occasion, la qualité de la segmentation.

Figure 2.2-13 : Image floue impliquant une composante connexe trop grande.

2.1.1.8 LUMINOSITÉ – CONTRASTE

Nous l’avons déjà évoqué et illustré à plusieurs reprises, le manque de luminosité influence les résultats. Vu le contraste peu prononcé de certaines images, la qualité de la segmentation est loin d’être optimale. N’oublions pas qu’une bonne luminosité renforce l’amplitude du contraste et donc facilite la distinction des cellules par rapport à l’arrière plan.

2.2.3 CAS EXTRÊMES

Dans certains cas, il ne faudra pas espérer obtenir un dénombrement satisfaisant, tout simplement car certaines images sont presque impossibles à traiter à cause de la densité de cellules.

Dans ce cas, un comptage par surface\(^{10}\) est certainement plus approprié.

Figure 2.2-14 : Cas extrêmes pour la segmentation.

\(^{10}\) Une méthode de comptage par surface est proposée par le logiciel que nous avons développé. (cf. section 5.5.1.3)
2.3 CONCLUSION

Face à la diversité des échantillons d’images, il était important de mettre en avant les sources potentielles d’erreurs pour le dénombrement de cellules. Les problèmes principaux rencontrés lors de l’analyse des images sont les suivants :

- Agrégations des cellules.
- Taches de coloration, pores et autres artefacts.
- Variation du contraste, de la luminosité.
- Présence et nature du bruit.
- Taille et morphologie des cellules.

Souvent, ces problèmes apparaissent simultanément sur une seule et même image, le plus délicat étant celui des agrégats. Prendre en considération les facteurs qu’il est possible de standardiser lors de l’acquisition améliorerait sensiblement la qualité des résultats.

Bien que l’approche naïve ne soit pas suffisante, nous sommes conscients qu’il existe une limite à la qualité de l’outil de comptage automatique et qu’aucune méthode de segmentation ne sera parfaite. Pour les cas extrêmes, même l’intervention d’un opérateur humain reste limitée.

Il est dès lors difficile de fixer des paramètres souvent nécessaires pour des algorithmes de traitement d’images. En effet, les caractéristiques d’une image varient considérablement d’une expérience à l’autre et ne sont donc pas utilisables dans un contexte générique. C’est pourquoi nous utiliserons, complémentairement aux outils de traitement d’images, des méthodes d’apprentissage automatique, et plus particulièrement de classification automatique d’images.
3.1 PRÉAMBULE

Cette section présente succinctement les deux grands domaines qui ont permis la réalisation de ce travail. D’une part, nous parlerons de l’apprentissage automatique et plus particulièrement de la classification automatique d’images. D’autre part, nous discuterons du traitement numérique d’images et des outils classiques utilisés dans cette discipline.

3.2 APPRENTISSAGE AUTOMATIQUE

Issu du domaine de l’intelligence artificielle, les méthodes d’apprentissage automatique offrent un champ d’applications aussi large que varié, notamment dans le secteur :

✓ **Industriel**
  - Contrôle de qualité
    - Afin de déterminer si le produit fini répond aux exigences requises, des méthodes de classification automatique peuvent remplacer ou venir en complément d’un système de traitement d’images.

✓ **Médical et bioinformatique**
  - Aide au diagnostic médical
    - Diagnostique des rhumatismes inflammatoires (arthrite rhumatoïde, ...) par classification de fragments de protéine.
    - L’objet même de ce mémoire.
  - Aide à la création de nouveaux médicaments.
  - Analyse et prédiction de données biologiques (génomique, protéomique, ...).

✓ **Financiers**
  - Analyse des marchés boursiers
  - Détection de fraudes
  - Reconnaissance de chiffres

✓ **Autres**
  - Robotique
  - Jeu d’échecs, labyrinthe,...
De nombreux algorithmes ont été développés pour l’apprentissage automatique. Nous nous contenterons, ici, de citer les algorithmes\(^{11}\) les plus fréquemment utilisés, à savoir :

- Les arbres de décision
- La méthode des k plus proches voisins
- Les réseaux de neurones
- Les machines à support vectoriel
- ...

Afin d’obtenir de nouvelles variantes d’apprentissage, ces méthodes peuvent également être combinées. Le but final étant de concevoir un système autonome, capable d’apprendre.

Il existe différents types d’apprentissage, notamment l’apprentissage supervisé, non supervisé, par renforcement, transductif, etc. Néanmoins, nous ne présenterons ici que le type d’apprentissage que nous avons utilisé, c’est-à-dire le mode supervisé. Nous laissons le soin au lecteur d’obtenir de plus amples informations sur ces cas supplémentaires.

### 3.2.1 Apprentissage supervisé

L’approche supervisée cherche à établir un modèle à partir d’une base de données contenant des objets prédéfinis et issus de l’ensemble d’apprentissage. Ces étiquettes déterminent les classes d’appartenance des objets \((C_1, C_2, \ldots, C_m)\) et sont attribuées par un expert du domaine étudié.

Soient :

- \(B\) un ensemble d’apprentissage,
- \(O_i = (A_1, A_2, \ldots, A_n)\) des objets caractérisés par des attributs et constituant \(B\).

\[
\begin{array}{cccc}
\text{Objets} & A_1 & A_2 & \ldots & A_n & \hat{Y} \\
\hline
2.3 & On & 3.4 & C_4 \\
1.2 & Off & 0.3 & C_1 \\
3.9 & Off & 2.8 & C_m \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
\end{array}
\]

\[
\hat{Y} = f(A_1, A_2, \ldots, A_n)
\]

Figure 3.2-1 : Apprentissage supervisé

Mathématiquement, le modèle est une fonction de prédiction \(f\) prenant, en entrée, les attributs des objets pour donner, en sortie, la classe d’appartenance la plus probable de ces objets.

Le modèle permet de prédire la classe à laquelle un nouvel objet appartient sur base de ses attributs. Il permet également de donner des informations sur les relations existantes entre les entrées et la sortie d’un objet.

---

\(^{11}\) Nous renvoyons à la littérature, et plus particulièrement au cours d’apprentissage inductif appliqué de M. Wehenkel, pour l’étude de ces algorithmes. Dans cet ouvrage, seuls les arbres de décisions seront détaillés.
3.3 **CLASSIFICATION AUTOMATIQUE D’IMAGES**

L’importance d’une rigoureuse classification des images est primordiale dans le domaine médical. Cette étape permet, notamment, d’établir un diagnostic médical qui, in fine, se doit d’être aussi précis que possible.

Néanmoins, la classification automatique n’est pas une étape triviale et constitue même un problème épineux car les résultats ne répondent pas toujours aux exigences voulues. Ceci s’explique par le fait que les images biomédicales présentent de nombreuses variations dues :

- Au contexte d’acquisition (manque ou variation de luminosité, matériel et protocole utilisés, sources de bruit,...).
- Aux corps étudiés : les cellules, par exemple, ont une structure non rigide. Il n’est donc pas toujours aisé de les caractériser selon leur forme.

Nous sommes dès lors conscients qu’une classification automatique parfaite relève de l’utopie, néanmoins, selon l’article “BMC Cell Biology” [9], des résultats de classification automatique comparables à ceux d’une classification « manuelle » ont été obtenus en utilisant une méthode reposant sur les arbres de décision. C’est pourquoi nous avons également décidé d’utiliser cette méthode que nous allons présenter dans les sections suivantes.

3.3.1 **LES ARBRES DE DÉCISION**

Comme introduit précédemment, le but d’un algorithme de classification est de créer un modèle qui, appliqué à une série d’images, permet de les classer le plus précisément possible. L’interprétation faite par le modèle donne un sens aux images, ce qui peut améliorer l’étude, le traitement et les résultats issus des expériences menées par les chercheurs.

Dans le cadre d’un apprentissage supervisé, l’algorithme d’apprentissage peut créer des arbres de décision qui définissent alors des modèles facilement interprétables. Les arbres de décision sont des méthodes de classification qui permettent de séparer les objets selon leurs attributs.

Comme le montre l’exemple ci-dessous, le principe est de construire des arbres binaires dont les nœuds intérieurs permettent de tester un attribut et où chaque feuille détermine une prédiction de la classe d’appartenance de l’objet.

---

12 Il est assez intuitif des séparer les objets selon leurs attributs en fonction d’une série de questions dont l’enchaînement dépend de la question précédente. La structure d’arbre apparaît naturellement.

13 Chaque nœud est associé à une question qui n’autorise que deux réponses possibles : « oui » ou « non ». 
La prédiction finale correspond soit à la classe majoritaire des objets de l’ensemble d’apprentissage ayant atteint une même feuille, soit à une distribution de probabilités des classes estimées par la fréquence de ces objets dans chaque classe.
Une étape critique lors de la génération d’un arbre de décision est le choix de l’attribut d’un nœud interne. En effet, le test effectué sur cet attribut doit diviser l’ensemble courant en deux sous-ensembles les plus homogènes possibles. Il s’agit dès lors de trouver le test dont le partitionnement est le plus « pur » de sorte à minimiser la profondeur finale de l’arbre généré.

Pour ce faire, une mesure de score est effectuée pour chaque attribut. L’attribut choisi sera celui ayant obtenu le meilleur score. Sur base de ce dernier, un nœud test sera créé et deviendra racine des deux nouveaux sous-ensembles obtenus lors du partitionnement de l’ensemble courant. L’opération sera réitérée sur ces deux nouveaux sous-ensembles. Le partitionnement d’un nœud de l’arbre se termine :

- Soit quand tous les objets de l’ensemble courant appartiennent à la même classe (feuille pure).
- Soit quand les valeurs des attributs sont constantes et ne permettent plus de diviser l’ensemble courant.
- Soit tout simplement parce qu’il est pertinent de limiter la profondeur de l’arbre.

Pour effectuer la mesure de score, il est courant d’utiliser la mesure d’impureté définie par l’entropie de Shannon :

\[
i(N) = - \sum_{j} P(C_j) \log_2 P(C_j)
\]

où \( P(C_j) \) est la proportion d'objets du nœud N qui appartiennent à la classe \( C_j \). D’après cette définition, l’entropie est nulle quand tous les objets appartiennent à la même classe, sinon elle est positive. L’entropie est maximale quand il existe exactement le même nombre d’objets de chaque classe au sein du même ensemble. Ainsi, le choix d’un test repose sur sa capacité à diminuer le plus possible l’impureté.

3.3.1.1 SUR-APPRENTISSAGE / SOUS-APPRENTISSAGE

Lors de la génération du modèle, nous sommes généralement confrontés à deux types de problèmes :

1. Soient deux arbres de décision \( T \) et \( T' \) et un ensemble d’apprentissage \( LS \). On parle de sur-apprentissage lorsqu’:
   a. \( Error_{LS}(T) < Error_{LS}(T') \).
   b. \( Error_{unseen}(T) < Error_{unseen}(T') \).

Autrement dit, le modèle \( T \) se focalise trop sur l’ensemble d’apprentissage et perd son pouvoir de prédiction sur de nouveaux échantillons. Si on choisit un \( LS \) différent, le modèle induit sera également fort différent du modèle \( T \). On parle aussi d’erreur de variance.

---

14 Au nœud racine, l’ensemble courant correspond à l’entière des objets de l’ensemble d’apprentissage.
15 Le partitionnement de l’ensemble courant est défini par le test effectué en un nœud interne de l’arbre.
16 Afin, par exemple, de définir le taux d’erreur à ne pas dépasser, le nombre d'objets minimum par feuille,…
2. À l'inverse, il existe le problème du **sous-apprentissage**. Dans ce cas, le modèle est dit pauvre et ne reflète pas correctement l'ensemble d'apprentissage. On parle alors de prédiction **biaisée**.

Les méthodes d'apprentissage doivent donc trouver un bon compromis entre variance et biais qui ont des valeurs inversement proportionnelles.

**Figure 3.3-2 : Sur et Sous-apprentissage.** *(Source : Introduction to Machine Learning, P.Geurts [4])*

### 3.3.2 ENSEMBLES D’ARBRES

Il est possible de construire plusieurs modèles (i.e. arbres de décision) à partir d'un seul ensemble d'apprentissage. Afin de déterminer la classe d'appartenance d'un nouvel objet, ce dernier sera propagé dans les différents arbres de l'ensemble. Chacun de ces arbres renvoie un vecteur de probabilités. La classe finale attribuée à l'objet est la classe ayant obtenu le plus de vote (classe majoritaire).

En pratique, il est rare de n'utiliser qu'un seul arbre de décision pour la création du modèle. En comparaison avec d'autres méthodes d'apprentissage, la méthode d'arbre de décision n'est pas aussi précise. C'est pourquoi nous privilégierons d'avantage les méthodes d'ensemble couplées aux méthodes d'arbres offrant ainsi une meilleure précision due à un modèle plus stable.

#### 3.3.2.1 EXTREMELY RANDOMIZED TREES (EXTRA-TREES)

Dans le but de générer différents arbres de décision à partir d'un ensemble d'apprentissage, diverses méthodes ont été élaborées, notamment la méthode d'**Extra-Trees**.

Soient $LS$ un ensemble d'apprentissage de $N$ attributs $\{a_1, \ldots, a_N\}$. Le principe est de générer $K$ partitions $\{s_1, \ldots, s_K\}$ où chaque $s_i$ est un test potentiel, choisi aléatoirement, pour le nouveau nœud à créer. Parmi l'ensemble de ces $s_i$, celui sélectionné pour être le nouveau nœud test sera celui dont la mesure de score est maximale. On le note $s_*$, on a donc :

$$Score(s_*, LS) = Max_{i=1, \ldots, K} Score(s_i, LS)$$
Le nouveau nœud devient racine de deux sous-arbres (à gauche et à droite) dont l’union est équivalente à $LS$.

Là où intervient le facteur aléatoire de l’algorithme se situe au niveau de la génération des différents tests $s_i$. Pour créer une partition aléatoire $s_i$, il convient d’appeler la méthode :

$$s_i = \text{Pick\_a\_random\_Split}(LS, a_i).$$

Celle-ci prend en entrée un ensemble $LS$ et un attribut $a_i$ (avec $i \in [1..K]$). Soient $a_{\text{min}}^{LS}$ et $a_{\text{max}}^{LS}$ respectivement les valeurs minimales et maximales de l’attribut $a_i$ dans l’ensemble $LS$, la méthode choisit aléatoirement un seuil $a_c$ dans l’intervalle $[a_{\text{min}}^{LS}, a_{\text{max}}^{LS}]$. Elle fournit alors en sortie un test aléatoire $[a < a_c]$.

La division d’un nœud se termine lorsque le nombre d’objets dans $LS$ est inférieur à un seuil $n_{\text{min}}$ ou si tous les objets de $LS$ sont de la même classe.

### 3.3.3 PIXIT

Nous allons présenter, de manière concise, les grandes étapes de l’algorithme de classification automatique d’images implémenté dans le logiciel PixIT. La méthode utilisée a été proposée par Raphaël Marée au cours de sa thèse de doctorat. Cette méthode vise à classer automatiquement une série d’images à l’aide d’un ensemble d’arbres de décision et de fenêtres extraites aléatoirement.

La méthode se veut également générique, cela signifie qu’aucune phase de prétraitement spécifique n’est appliquée, contrairement à d’autres méthodes d’apprentissage qui, sans cette phase de prétraitement\(^\text{17}\), pourrait souffrir d’une variance trop importante et de temps de calculs élevés. La « méthode PixIT » permet donc de gérer un très grand nombre de variables d’entrées : les pixels.

---

\(^{17}\) La phase de prétraitement consiste à extraire des caractéristiques pertinentes des objets en vue de réduire la complexité.
Chapitre 3 – Domaines étudiés

3.3.3.1 PHASE D’APPRENTISSAGE

Soit un ensemble d’apprentissage $L_S$ contenant $N$ images. Le principe est d’extraire aléatoirement $N_{sw}$ sous-fenêtres de taille aléatoire avec $N_{sw} >> N$. Chaque sous-fenêtre est redimensionnée en une taille fixe (par défaut 16x16) et étiquetée avec la classe de l’image parente :

![Diagramme](image1)

Figure 3.3-3 : Extraction de sous-fenêtres, redimensionnement et étiquetage.
(Source : Biological Image Classification with RandomSubWindows and Extra-Trees, Raphaël Marée[8])

À partir de ces sous-fenêtres, on crée une base de données où chaque fenêtre est décrite par la valeur des pixels. Nous avons donc, pour chaque fenêtre, 256 attributs si l’image est en niveaux de gris et 768 attributs pour une image couleur. Ensuite, grâce à la méthode d’Extra-Trees, on construit un ensemble de $T$ arbres aléatoires sur base des valeurs des pixels des fenêtres extraites afin d’obtenir le modèle final de classification de fenêtres.

![Diagramme](image2)

Figure 3.3-4 : Construction d’un ensemble de $T$ arbres aléatoires.
(Source : Biological Image Classification with RandomSubWindows and Extra-Trees, Raphaël Marée[8])
3.3.3.2 PHASE DE PRÉDICTION

Pour prédire la classe d’une image inconnue $I_Q$, on applique le modèle aux sous-fenêtres de $I_Q$ extraites aléatoirement. On obtient, pour chaque sous-fenêtre, un vecteur de probabilité d’appartenance de classe. En moyennant l’ensemble de ces vecteurs, on arrive au vecteur final caractérisant l’image $I_Q$. Celle-ci reçoit comme prédiction la classe majoritaire de ce vecteur:

![Diagramme de la phase de prédiction de la méthode «PixIT»](image)

Figure 3.3-5 : Phase de prédiction de la méthode «PixIT».
(Source : Random Subwindows and Randomized Trees, Raphaël Marée[10])
Chapitre 3 – Domaines étudiés

3.4 TRAITEMENT NUMÉRIQUE DES IMAGES

Le traitement numérique des images désigne l’ensemble des théories, méthodes et techniques qui permettent de manipuler les images numériques et dont le but est d’améliorer, étudier, corriger ou interpréter l’information présente dans ces dernières.

Plus particulièrement en biologie cellulaire, le traitement d’images permet aujourd’hui de quantifier des phénomènes, d’automatiser des tâches répétitives. De plus, avec la capacité sans cesse croissante des microscopes et des ordinateurs, la qualité du traitement et donc des résultats s’améliore significativement.


3.4.1 PRÉ-REQUIS

On appelle image numérique toute image qui a été acquise, traitée, créé ou sauvegardée sous forme binaire. Ces images numériques sont constituées d’un ensemble de points appelés pixels.

Le pixel est l’unité de surface permettant de définir la base d’une image numérique. Il est généralement codé sur 8 bits pour les images en niveaux de gris et sur 24 bits pour les images couleurs\textsuperscript{18}.

La couleur est la perception que nous avons des différentes longueurs d’onde qui constituent la lumière visible. Cet ensemble de longueurs d’onde, appelé spectre lumineux, s’étend du violet (hautes fréquences) au rouge (basses fréquences). Nous verrons plus loin différentes modélisations des couleurs pour les images numériques.

La résolution d’une image est le nombre de pixels par unité de longueur dans cette image. On parle également de densité de pixels dans l’image. La résolution s’exprime en « pixel par pouce » et ne doit pas être confondue avec la définition d’une image qui correspond simplement au nombre fixe de pixels utilisés pour la représenter.

Pour terminer, introduisons le concept d’histogramme. D’un point de vue général, un histogramme est un outil de statistique permettant de faciliter l’analyse de la répartition d’une

\textsuperscript{18} Il est également fréquent de coder les images couleurs sur 32 bits ; l’octet supplémentaire permet, par exemple, de gérer la transparence.
Chapitre 3 – Domaines étudiés

variable aléatoire continue. Le principe d’un histogramme est de diviser l’axe réel en intervalles (ouverts d’un côté et fermés de l’autre) et de compter le nombre de réalisations de la variable aléatoire qui rentrent dans ces intervalles.

Dans le cas d’une image, le nombre de réalisations est donné par le nombre de pixels de l’image. L’histogramme indique alors la répartition des pixels en fonction de leur luminosité. Pour une image en niveaux de gris (256 valeurs), l’histogramme représente donc les 256 niveaux d’intensité lumineuse en abscisse et le nombre de pixels correspondants dans l’image en ordonnée, comme l’illustre la figure ci-dessous : 

**Figure 3.4-2 : Histogramme d’une image en niveaux de gris.**

L’histogramme s’avère être un outil très intéressant pour le traitement d’images. Modifier l’histogramme d’une image permet par exemple d’augmenter ou de diminuer le contraste (étirement de l’histogramme) ou d’augmenter les nuances dans l’image (égalisation de l’histogramme). Il peut également être utilisé pour déterminer automatiquement le seuil « optimal » d’une image.
3.4.2 ESPACES COLORIMÉTRIQUES

Un espace colorimétrique est un système n-dimensionnel utilisé pour représenter les couleurs. En général, il est représenté en trois dimensions. Une couleur se caractérise alors par ses coordonnées selon les trois axes de l'espace.

3.4.2.1 RGB

Le plus connu et souvent le plus utilisé des espaces colorimétriques, l'espace RGB (Red-Green-Blue) fut définit en 1931 par la CIE et très vite adopté comme standard des espaces RGB. Ces trois couleurs, rouge, verte et bleue sont les couleurs primaires et correspondent aux trois longueurs d'ondes auxquelles sont sensibles les trois types de cônes de l'œil humain.

Par synthèse additive, on peut obtenir un très grand nombre de couleur dont les couleurs secondaires jaune cyan et magenta. L'addition des trois couleurs donne le blanc. Chaque pixel d'une image RGB est caractérisé par 3 octets, un pour chaque couleur, dont les valeurs varient entre 0 et 255.

![Figure 3.4-3 : Espace RGB et synthèse additive.](Source : http://www.astrosurf.com/luxorion/spectro-trichromiesoleil.htm)

Cet espace est malheureusement fort peu intuitif. En effet, il est difficile de décomposer intuitivement une couleur selon ses trois composantes RGB. Il existe une variante que l'on nomme RGB normalisé et qui permet de s'affranchir des variations d'intensité lumineuse (I = R+G+B). Cet espace (r,g,b) normalisé est défini comme suit :

\[
\begin{align*}
    r &= \frac{R}{I} \\
    g &= \frac{G}{I} \\
    b &= \frac{B}{I}
\end{align*}
\]

3.4.2.2 HSI

L'espace HSI, Hue (teinte) – Saturation (saturation) – Intensity (intensité), est également un espace tridimensionnel. Il permet de caractériser une couleur de façon plus intuitive.

De fait, la teinte correspond à la représentation que l'humain se fait d'une couleur (rouge, jaune, brun,...). Plus techniquement, elle représente la longueur d'onde dominante du spectre de couleur, c'est-à-dire la forme pure de la couleur sans adjonction de blanc ou de noir. La teinte est caractérisée par un angle variant de 0° à 360°.

---

19 Les nombreux dispositifs d'acquisition d'images couleurs utilisent directement l'espace RGB pour coder les images.
20 Le système RGB standardisé repose sur les longueurs d'onde définie comme suit : 700nm pour le rouge (R), 546,1nm pour le vert (G) et 435,8nm pour le bleu (B).
Chapitre 3 – Domaines étudiés

La saturation décrit le niveau de pureté de la teinte par rapport à une référence blanche. Par exemple, une couleur entièrement rouge et sans aucun blanc est saturée à 100%. Si nous ajoutons un peu de blanc, la couleur se décale de rouge vers rose. La teinte est toujours rouge mais moins saturée.

Enfin l’intensité décrit la quantité de lumière. Le minimum d’intensité correspond au noir et le maximum au blanc. Ci-dessous, la représentation classique de l’espace HSI :

![Espace HSI](http://en.wikipedia.org/wiki/File:HSV_triangle_and_cone.png)

Selon le problème à résoudre, il est parfois plus intéressant de travailler dans cet espace. En effet, pour la reconnaissance d’objets colorés par exemple, on sait que les composantes \(H\) et \(S\) sont indépendantes de toute variation de luminosité. Plus particulièrement, \(H\) est fort peu sensible aux phénomènes d’ombre.

### 3.4.2.3 XYZ

Pour pallier le fait qu’il faut parfois prendre une intensité négative, pour la composante rouge de l’espace RGB, afin d’obtenir toutes les couleurs pures, la CIE a défini l’espace XYZ où toutes les composantes sont positives.

La conversion\(^{21}\) depuis l’espace RGB est définie comme ceci :

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
2.769 & 1.7518 & 1.13 \\
1 & 4.5907 & 0.0601 \\
0 & 0.0565 & 5.5943
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]

En réalité, la conversion de l’espace RGB vers l’espace XYZ dépend de la valeur du blanc de référence, ce dernier dépendant lui-même de l’illuminant\(^{22}\) de la scène.

A l’instar du RGB, l’espace XYZ définit de la même façon les coordonnées normalisées \((x,y,z)\). On a :

\[
x = \frac{X}{X + Y + Z} ; \quad y = \frac{Y}{X + Y + Z} ; \quad z = \frac{Z}{X + Y + Z}
\]

avec \(x + y + z = 1\)

\(^{21}\) [http://www2.ulg.ac.be/telecom/teaching/notes/totali/elen016.pdf]

Chapitre 3 – Domaines étudiés

La somme des trois composantes étant égale à 1, seules les deux composantes x et y sont nécessaires pour représenter l’information colorimétrique. Ainsi, la CIE a également défini l’espace xyY où Y indique la luminance alors que les valeurs de x et y déterminent la chrominance.

Pour conclure, notons qu’il existe d’autres espaces de couleurs comme les espaces utilisés pour les systèmes télévisuels YCbCr, YIQ, YUV mais aussi les espaces de couleurs complémentaires RG, Yeb, WhBl ou encore les espaces uniformes L’a b* et L’u’v*.

3.4.3 LES FILTRES

A défaut d’entrer dans les théories mathématiques, nous allons présenter les filtres de manière pratique. L’approche se veut donc concrète et le but de cette section est de donner une vue d’ensemble des filtres élémentaires couramment utilisés.

Précisons simplement qu’il existe des filtres linéaires et des filtres non-linéaires. Les **filtres linéaires** consistent à appliquer une matrice M de dimension \(m \times n\) (appelée matrice de convolution) à une image I de dimension \((r_x \times r_y)\) afin d’obtenir l’image filtrée \(I'_{xy}\), résultat du produit de convolution. On a donc :

\[
M_{ij} \quad \text{avec} \quad i < m, \quad j < n
\]

\[
I_{xy} \quad \text{avec} \quad x < r_x, \quad y < r_y
\]

\[
I'_{xy} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} M_{ij} \ast I_{x+i,y+j} \quad \text{avec} \quad (x < r_x - n + 1, y < r_y - m + 1)
\]

Les **filtres non-linéaires**, eux, remplacent chaque pixel par une valeur égale à une loi non-linéaire de ses voisins. Il peut arriver que la valeur d’un pixel soit négative ou supérieure à 255. Dans le premier cas, soit on seuille à la valeur 0, soit on prend la valeur absolue. Dans le second cas, on attribue généralement au pixel la valeur limite (i.e. 255 dans la majorité des cas).

Notons également qu’il est usuel de filtrer non pas l’image originale, mais par exemple son équivalent après **transformation de Fourier**. On parle alors de filtrage spectral qui s’applique dans l’espace de Fourier (domaine fréquentiel). Le **gradient** de l’image présente également des propriétés utiles à certains problèmes de segmentation comme la ligne de partage des eaux.

3.4.3.1 LES FILTRES PASSE-BAS

Le filtre passe-bas, que l’on peut également appeler filtrer « coup-haut », est un filtre qui ne modifie pas ou très peu les basses fréquences dans l’image et qui, au contraire, atténue ou supprime les composantes de haute fréquence (transitions fortes d’intensité lumineuse). On l’utilise principalement pour atténuer le bruit (lissage) et les irrégularités de l’image.

Ce filtre crée généralement un effet de flou car il atténue les transitions rapides d’intensité dans l’image. En pratique, il faut choisir un compromis entre l’atténuation du bruit et la conservation des détails et contours significatifs.

---

23 L’information de couleur.
A titre d’exemple, et parce que nous l’avons utilisé, nous pouvons citer le filtre médian. Ce dernier consiste à remplacer la valeur d’un pixel $p$ par la valeur médiane des valeurs formées par les pixels voisins à $p$. Le voisinage de $p$ est défini par une fenêtre de dimension impaire (par ex. $3 \times 3$, $5 \times 5$, ...).

Le filtre médian ne crée pas de nouveaux niveaux de gris et présente la particularité d’introduire moins de flou dans l’image que les autres filtres de lissage. Il est courant d’utiliser le filtre médian si l’effet recherché correspond à un seuillage de l’image avant sa conversion en une image binaire.

3.4.3.2 LES FILTRES PASSE-HAUT

Au contraire des filtres passe-bas, les filtres passe-haut ne modifient pas ou très peu les hautes fréquences et atténuent ou suppriment les composantes de basse fréquence (transitions faibles).

Ces filtres ont pour but de renforcer le contraste et de mettre en évidence les contours. Un contour correspond à une discontinuité locale d’intensité lumineuse dans l’image. Les techniques de détection de contours se basent principalement sur l’utilisation :


En guise d’illustration, quelques exemples de noyaux connus :

\[
\begin{bmatrix}
0 & 0 & 0 \\
1 & 0 & -1 \\
0 & 0 & 0
\end{bmatrix}
\quad
\begin{bmatrix}
1 & 0 & -1 \\
1 & 0 & -1 \\
1 & 0 & -1
\end{bmatrix}
\quad
\begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1
\end{bmatrix}
\]

Croix Prewitt Sobel

- Des filtres laplacien : les contours correspondent aux 0 de la fonction (dérivée du second ordre). On rencontre généralement 3 noyaux critiques :

\[
\begin{bmatrix}
0 & -1 & 0 \\
-1 & 4 & -1 \\
0 & -1 & 0
\end{bmatrix}
\quad
\begin{bmatrix}
-1 & -1 & -1 \\
-1 & 8 & -1 \\
-1 & -1 & -1
\end{bmatrix}
\quad
\begin{bmatrix}
1 & -2 & 1 \\
-2 & 4 & -2 \\
1 & -2 & 1
\end{bmatrix}
\]

Remarque : Il existe un troisième type de filtre appelé filtre passe-bande. Un filtre passe-bande peut être vu comme le résultat d’un filtrage « passe-bas » suivi d’un filtrage « passe-haut ». Seule une plage de fréquences n’est pas ou peu atténuée par le passage au travers de ce filtre.
Les opérateurs morphologiques

L’étude des opérateurs morphologiques repose sur la théorie des ensembles et a été essentiellement inspirée par des problèmes de traitement d’images. Les algorithmes de morphologie mathématique travaillent essentiellement sur des images binaires.

L’idée de base de la morphologie est de traiter un ensemble à l’aide d’un autre ensemble appelé élément structurant ou noyau. À chaque position de l’élément structurant, on vérifie s’il est inclus dans l’ensemble initial (image binaire). En fonction de la réponse, on construit un ensemble de sortie.

Nous allons illustrer les deux opérateurs morphologiques de base :

- **La dilatation**
  Soient $X$ une image binaire et $K$ un élément structurant. Le dilaté de $X$ par $K$ est l’ensemble des pixels $p$ de l’image tels que $K_p$ (élément structurant $K$ centré sur le pixel $p$) a une intersection non vide avec $X$.

  \[
  \text{dilate}(X, K) = \{p \in X \mid X \cap K_p \neq \emptyset\}
  \]

- **L’érosion**
  Soient $X$ une image binaire et $K$ un élément structurant. L’érodé de $X$ par $K$ est l’ensemble des pixels $p$ de l’image tels que $K_p$ est totalement inclus dans $X$.

  \[
  \text{erode}(X, K) = \{p \in X \mid K_p \subset X\}
  \]

24 [http://www.montefiore.ulg.ac.be/~piater/courses/INFO0903/notes/3-grouping/foil05.xhtml](http://www.montefiore.ulg.ac.be/~piater/courses/INFO0903/notes/3-grouping/foil05.xhtml)
Chapitre 3 – Domaines étudiés

Illustration

Figure 3.4-7 : Erosion morphologique.

Si $K_p$ est totalement inclus dans $X$, le pixel $p$ appartient à l’érodé.

A partir de ces deux opérations, on en définit deux nouvelles à savoir l’ouverture et la fermeture.

L’ouverture consiste d’abord éroder l’image avec un élément structurant $K$, puis à dilater le résultat avec son symétrique. Le but d’une ouverture est de lisser les contours et d’isoler les surfaces présentes dans l’image.

La fermeture est l’opération contraire de l’ouverture : on applique d’abord une dilatation, puis une érosion avec l’élément structurant symétrique. Ici, le but est d’égalemnt lisser les contours mais surtout combler les surfaces proches de manière à fermer les contours disjoints.

Ces opérateurs simples permettent une multitude de tâches pour le traitement d’images comme la squelettisation, le rehaussement et la restauration d’images, l’amélioration du regroupement en composante connexe ou encore la recherche de contours.

Les quatre opérations que nous venons de citer peuvent également s’appliquer à des images en niveaux de gris. Pour de telles images, il existe différentes façons de dilater ou d’éroder. Une opération de dilatation possible (resp. érosion) consiste à remplacer la valeur du pixel $p$ par le maximum (resp. le minimum) des valeurs voisines qui sont définies par l’élément structurant :

Figure 3.4-8 : Dilatation d’une image en niveaux de gris (exemple 1).

25 Si l’élément structurant est symétrique, il sera identique pour l’opération d’érosion et de dilatation.
26 Opérations itératives d’amincissement d’une forme ⇒ obtention du squelette.
27 $\text{Gradient}(\text{Image, } K) = \text{Dilatation}(\text{Image, } K) - \text{Erosion}(\text{Image, } K)$.
28 Et par extension, à des images couleurs.
3.4.4 SEGMENTATION

La segmentation est un processus consistant à partitionner une image \( I \) en plusieurs régions \( R_i \), selon certains critères prédéfinis, et tels que :

- \( \forall i, R_i \neq \emptyset \)
- \( \forall i,j, i \neq j, R_i \cap R_j = \emptyset \)
- \( \varepsilon = \bigcup_i R_i \)

Généralement, le résultat de la segmentation est une image binaire caractérisant les différentes régions, appelées composantes connexes. Les techniques de segmentation sont nombreuses et parfois plus adaptées à certains types de problèmes.

On parle de segmentation de bas niveau quand on ne tient compte que de l’information contenue dans l'image, au contraire de la segmentation haut niveau qui tente d’accorder à chaque région un objet sémantique.

L’approche peut être locale, c’est-à-dire que, contrairement à l’approche globale où on considère la totalité de l'image, seulement le voisinage direct de chaque pixel est pris en compte.

Pour terminer, signalons au lecteur qu’il est fréquent de combiner les deux méthodes de segmentation, illustrées ci-dessous, selon les caractéristiques du problème à résoudre.

3.4.4.1 SEGMENTATION PAR SEUILLAGE

Les techniques de segmentation par seuillage consistent à créer une image binaire \( I_b \) à partir d'une image couleur ou en niveaux de gris \( I_c \) sur base d'un seuil \( T_h \). L’hypothèse défendue est que les objets se différencient de l’arrière plan.

Par exemple, on peut décider que pour chaque valeur de pixel de l'image \( I_c \) inférieure ou égale à \( T_h \), on attribue la valeur 0 (background) au pixel correspondant dans l'image \( I_b \). Par contre, si cette valeur est supérieure au seuil \( T_h \), le pixel correspondant prend la valeur 255 (arrière plan).

Pour une même application, si l’ensemble des images présentent des caractéristiques semblables, un seuil de valeur fixé empiriquement peut parfaitement convenir. Ce cas limite toutefois l’application à un seul type d’images. C'est pour cette raison que des techniques de seuillage automatique ont été élaborées, les plus connues étant probablement celles d’Otsu\(^{29}\) et de Ridler\(^{30}\).\[29\] http://en.wikipedia.org/wiki/Otsu%27s_method\[30\] Le choix du seuil initial doit évidemment se situer dans l’intervalle défini par l’histogramme. Généralement, il s’agit de l’intervalle [0,255] couramment utilisé pour définir l’histogramme d’une image.
Chapitre 3 – Domaines étudiés

\[ S_{k+1} = \frac{\text{Mean}_{\text{inf}} + \text{Mean}_{\text{sup}}}{2} \]

\( S_k \) devient \( S_{k+1} \) et on réitère cette opération tant que la valeur du seuil ne converge pas, i.e. tant que \( S_k \neq S_{k+1} \). Cette valeur de convergence définit le seuil final. L'algorithme de Ridler présente l'avantage d'être extrêmement simple et rapide.

Figure 3.4-9 : Résultat du seuillage de Ridler.

3.4.4.2 SEGMENTATION PAR DÉTECTION DE RÉGIONS

Une méthode souvent utilisée pour ce type de segmentation est la méthode par croissance de région. Elle consiste à faire croître une région, selon un critère prédéfini, par incorporation des pixels jusqu'à ce que toute l'image soit couverte. Le premier ensemble de régions choisi pour débuter la croissance peut par exemple correspondre à l'ensemble des minimas de l'image.

L'algorithme de « ligne de partage des eaux » (LPE) appartient à ce type de méthode. En pratique, on considère la surface du gradient de l'image comme un relief topographique où les bassins versants sont séparés par des crêtes.

Figure 3.4-10 : Relief du gradient de l'image (bassins versants, minima et LPE).
(Source : http://www2.ulg.ac.be/telecom/teaching/notes/totali/elen016/node140_tf.html)

Un point de vue possible est d'imaginer l'immersion de ce relief dans l'eau, en précisant que l'eau ne peut pénétrer dans les vallées que par les minima. Elle monte ensuite progressivement jusqu'à ce que deux bassins distincts se rejoignent déterminant ainsi une ligne.

31 Une image gradient est une image obtenue lorsqu'on applique un opérateur de dérivée spatiale à l'image originale.
Chapitre 3 – Domaines étudiés

de rencontre, appelée ligne de partage des eaux. Le processus est répété jusqu’à ce que le plus
haut point du relief soit inondé.

Conceptuellement, le principe est simple à comprendre. En pratique, il faut faire appel aux
notions de distance géodésique, chemin géodésique et zones d’influence.

**Définition :** Soit un ensemble $X$ composé d’objets disjoints. A chaque objet $X_i$, on peut
associer une **zone d’influence** $Y_i$ telle que chaque point $y$ de $Y_i$ est plus proche de $X_i$ que de tout
autre objet $X_j$ pour tout $j \neq i$.

$$Y_i = \{ y \mid \forall j \neq i, \text{dist}(y, X_i) < \text{dist}(y, X_j) \}$$

Dans notre cas, les distances en question sont des distances géodésiques. Notons également que les bassins versants correspondent aux zones d’influences des minima locaux de l’image traitée.

**Définition :** La distance géodésique entre les points $x$ et $y$ est la longueur du plus court
chemin géodésique reliant $x$ à $y$ ; elle est infinie si un tel chemin n’existe pas.

**Définition :** Un chemin géodésique de longueur $l$ entre deux points $x$ et $y$ est une famille de
$l + 1$ pixels $z_0 = x, z_1, ..., z_l = y$ telle que :

$$\forall i \in [0, l], z_i \in \text{Image et } \forall i \in [0, l], z_{i-1}, z_i \text{ sont voisins}$$

Introduisons maintenant l’algorithme LPE. Soient :

- $f$, la fonction étudiée
- $h_{min}$ et $h_{max}$, respectivement les valeurs minimales et maximale de $f$ sur son domaine
de définition
- $M_i$ les minima et $C(M_i)$ les bassins correspondants avec :

$$
\begin{align*}
  T_h(f) &= \{ x \in \text{dom} f, f(x) \leq h \}, \text{ h étant un seuil} \\
  C_h(M_i) &= \{ x \in C(M_i), f(x) \leq h \} = C(M_i) \cap T_h(f) \\
  C(M) &= \bigcup_i C(M_i) = l’\text{union de tous les bassins versants}
\end{align*}
$$

$C_h(M_i)$ est défini comme la partie du bassin $M_i$ rempli au temps algorithmique $h$. De même, $C_h(M)$ est défini comme la partie de l’union des bassins $M_i$ remplis au temps $h$.
L’inconnue du problème est $C(M_i)$ qu’il faut construire progressivement.

Initialement, on considère que tous les minima sont des sources d’où l’eau peut pénétrer,
ce qui se traduit mathématiquement par :

$$C_{h_{min}}(M) = T_{h_{min}}(f)$$
Ensuite, la construction des différentes régions se déroule comme suit :

\[ \forall h \in [h_{\text{min}} + 1, h_{\text{max}}] : C_h(M) = ZI_h \cup \text{Min}_h \]

où

- \( ZI_h \) est la zone d'influence (de domaine \( T_h(f) \)) composée des centres contenu dans \( C_{h-1}(M) \).
- \( \text{Min}_h \) est l'ensemble des points de \( T_h(f) \) qui, après le processus de recherche des zones d'influences, n'ont toujours pas de voisin. Il s'agit de nouveaux minima, centres de nouvelles zones d'influence.

**Problème** : La segmentation par ligne de partage des eaux est très sensible au bruit qui engendre de faux minima, d'où de nouveaux bassins versants pour la LPE qui sont, en réalité, des bassins indésirables. L'image résultante est alors sur-segmentée. Pour pallier à ce problème, l'application d'un filtre de lissage s'avère généralement insuffisant.

C'est pour cette raison que la technique du marquage a été développée. Elle consiste à déterminer le nombre de minima locaux et donc le nombre de zone que l'on souhaite mettre en évidence grâce à la LPE. Pour ce faire, on dispose a priori d'un critère permettant de séparer les « vrais » minima des « faux ». Lors de la montée des eaux, seuls les bassins qui ont été marqués seront inondés.
CHAPITRE 4

ETAT DE L’ART

4.1 INTRODUCTION

Avant de nous lancer dans le développement de l’outil de comptage, il nous semblait intéressant de nous documenter sur les méthodes et/ou algorithmes déjà existants dans ce domaine. Comme on peut s’en douter, le comptage des cellules n’est pas un problème nouveau. Or, on le sait, un travail répétitif et ennuyeux a toujours poussé l’homme à automatiser la tâche à accomplir.

Le père du comptage automatique de cellules se nomme Wallace H. Coulter\textsuperscript{32}. Il a fabriqué et breveté, en 1953, un appareil connu sous le nom de « Coulter Counter ». Cet appareil fonctionne comme suit : La solution étudiée est aspirée avec une pompe à vide vers un tube chargé électriquement et au bout duquel se trouve un trou. En passant par le trou, chaque particule de la solution bloque le champ électrique pendant un certain moment. Au final, il est donc possible de compter le nombre de cellules présentes dans la solution.

4.2 RECHERCHE BIBLIOGRAPHIQUE

Evidemment, cette introduction n’est qu’anecdotique dans notre cas puisqu’il nous a été demandé d’établir un outil de comptage à partir d’images. En ce sens, nous avons recherché, dans un premier temps, des logiciels, commercialisés ou non, capables de répondre à notre problématique. Nous n’avons cependant pas trouvé une solution susceptible de convenir. De fait, plusieurs sociétés telles que Sysmatec\textsuperscript{33}, Nexcelom\textsuperscript{34}, vendent des produits de type « compteur de colonies automatique » mais aucune d’entre elles n’a accepté de nous fournir quelques explications sur les algorithmes de traitement d’images utilisés.

C’est pourquoi nous nous sommes tournés vers les articles scientifiques. La littérature offre une panoplie d’articles intéressants. Malheureusement, une grande partie de ces articles ne sont pas libres d’accès.

Parmi ceux que nous avons analysés\textsuperscript{19}|\textsuperscript{20}|\textsuperscript{21}|\textsuperscript{22}, seule la thèse d’Estelle Glory[1] a particulièrement influencé notre travail. Les autres articles, bien qu’intéressants, proposent des méthodes trop peu détaillées mais surtout des méthodes focalisées sur un type d’images bien

\textsuperscript{32} http://www.beckmancoulter.com/hr/ourcompany/oc_WHcoulter_bio.asp
\textsuperscript{33} http://www.sysmatec.ch/French/fB1_ComptageColonie.html
\textsuperscript{34} http://www.nexcelom.com/Products/CellCounter/Features.html
précis. Nous ne pouvons dès lors pas les utiliser et nous remarquons là toute la difficulté de notre travail !

Néanmoins, comme nous venons de le dire, la thèse d'E. Glory, est particulièrement riche en termes d'informations pertinentes pouvant être utilisées dans le présent travail. C'est pourquoi nous allons consacrer les sections suivantes à une analyse des techniques et méthodes mises en œuvre dans cette thèse.

4.3 THÈSE D’E. GLORY

Dans cette section, nous analysons les trois grandes étapes développées dans l'outil de comptage automatique réalisé par E. Glory, savoir :

1. La segmentation d'images cytologiques en couleur, (section 4.3.2)
2. La classification des composantes connexes, (section 4.3.3)
3. La segmentation des agrégats. (section 4.3.4)

Le lecteur remarquera, au chapitre suivant, que nous avons également adopté cette stratégie à 3 temps. Néanmoins, les algorithmes utilisés pour les phases 2 et 3 de notre logiciel diffèrent de ceux utilisés par E. Glory.

Avant de parler de ces trois grandes étapes, nous allons étudier un critère développé par E. Glory et qui permet d'évaluer automatiquement la qualité d'une segmentation pour des images similaires à celles qui nous ont été fournies.

4.3.1 ÉVALUATION DE LA QUALITÉ D'UNE SEGMENTATION D'IMAGES COULEUR

En général, une segmentation précise ne peut s'obtenir qu'à partir de connaissances à priori sur le domaine étudié. Néanmoins, diverses méthodes ont été développées afin d'évaluer automatiquement la qualité d'une segmentation sans aucune connaissance a priori de l'image. Pour notre application, les critères d'une bonne segmentation sont :

- La séparation des couleurs
- La distinction des noyaux formant un agrégat
- L’absence de régions trop petites

La plupart des techniques d'évaluation automatique s'appliquent à une segmentation dite « de bas niveau » souvent employée comme première étape d'une stratégie ascendante.

Nous allons maintenant détailler et analyser le critère d'évaluation automatique d'E. Glory appelé critère C et proposé par E. Glory. Ce critère repose sur deux critères de base, à savoir le critère de Liu[14] et le critère de Borsotti[15].

4.3.1.1 CRITÈRE DE LIU

J. Liu est le premier à définir un critère capable d'évaluer automatiquement une la qualité d'une segmentation réalisée sur une image en couleur. Le critère de Liu prend en compte deux paramètres, à savoir le nombre de régions et l'homogénéité de la couleur de chaque région. Nous avons donc le critère suivant :

\[(4.1)\]
Chapitre 4 – Etat de l’art

\[ L(I) = \frac{1}{1000A} \sqrt{R} \sum_{i=1}^{R} \frac{e_i^2}{\sqrt{A_i}} \]

où \( I \) est l’image segmentée, \( R \) le nombre de régions après segmentation, \( A_i \) l’aire de la \( i^{ème} \) région et \( e_i \) l’hétérogénéité des couleurs de la \( i^{ème} \) région. \( L \) est normalisé par une constante \( A \), la taille de l’image.

Le terme \( e_i \) représentant la dispersion des couleurs de la région \( i \) est mesuré, dans l’espace RGB, comme suit :

\[ e_i = \sqrt{\sum_{p_i} [(p_{iR} - \mu_{iR})^2 + (p_{iG} - \mu_{iG})^2 + (p_{iB} - \mu_{iB})^2]} \]

Cette expression mesure donc la somme des distances euclidiennes entre les vecteurs de couleur des pixels \( p \) de la région \( i \) et le vecteur de la couleur moyenne \( \mu_i \) de cette région.

Pour en revenir à l’expression 4.1, plus la valeur de \( L(I) \) est petite et plus la segmentation est considérée comme satisfaisante. On remarque donc, via le facteur global \( \sqrt{R} \), que le critère de Liu sera d’autant meilleur que le nombre de régions est limité. D’un autre côté, la mesure locale \( \frac{e_i^2}{\sqrt{A_i}} \) pénalise les petites régions ou celles qui ont une grande hétérogénéité de couleur.

Ces deux paramètres jouent un rôle de balance afin de trouver un compromis entre la préservation des détails et l’élimination du bruit. Le problème majeur du critère de Liu est qu’il sera favorable à une image présentant de nombreuses petites régions. En effet, pour chaque région, le facteur \( \frac{e_i^2}{\sqrt{A_i}} \) tend vers 0. Il n’y a alors plus que le terme global \( \sqrt{R} \) pour augmenter la valeur de \( L(I) \) mais il ne suffit pas toujours.

4.3.1.2 CRITÈRE DE BORSOTTI

Pour remédier à la faiblesse du critère de Liu, Borsotti propose d’introduire un nouveau terme :

\[ \sum_{i=1}^{R} \left( \frac{N(A_i)}{A_i} \right)^2 \]

Le critère de Borsotti est alors défini comme suit :

\[ B(I) = \frac{1}{1000A} \sqrt{R} \sum_{i=1}^{R} \left[ \frac{e_i^2}{1 + \log A_i} + \left( \frac{N(A_i)}{A_i} \right)^2 \right] \]

où \( N(A_i) \) représente le nombre de régions ayant une aire égale à \( A_i \). Le premier terme\(^{35}\) de la somme sera élevé lorsque la couleur des différentes régions n’est pas homogène (caractéristique des grandes régions), alors que le second terme sera élevé pour les régions dont

\(^{35}\) Le dénominateur de ce premier terme permet de pénaliser encore plus les régions de couleur non-homogène.
Chapitre 4 – Etat de l’art

l’aire $A_i$ est égale à l’aire de nombreuses autres régions de l’image (caractéristique des petites régions). Autrement dit, Borsotti a introduit l’expression 4.2 en espérant que sa valeur soit égale à 1 pour les grandes régions et supérieures à 1 pour les petites régions. Ainsi, l’avantage du critère de Borsotti par rapport à celui de Liu est de tenir compte des pixels isolés.

Nous allons voir comment E. Glory a utilisé ces deux critères pour en définir un nouveau, le critère $C$, encore plus adapté au genre d’images que nous traîtons.

4.3.1.3 CRITÈRE $C$

Afin d’établir son propre critère, E. Glory a tout d’abord appliqué les critères de Liu et Borsotti à des images caractérisées par un fond clair et relativement uniforme et des objets biologiques de couleurs relativement homogènes. Voici l’exemple qu’elle a analysé :

![Image originale](image.png)

<table>
<thead>
<tr>
<th>Référence</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Liu $L_{(RGB)}$</th>
<th>d</th>
<th>e</th>
<th>c</th>
<th>b</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10,3</td>
<td>25,2</td>
<td>40,9</td>
<td>47,3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Borsotti $B_{(RGB)}$</th>
<th>e</th>
<th>c</th>
<th>b</th>
<th>a</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,7</td>
<td>16,0</td>
<td>20,9</td>
<td>25,3</td>
<td>1 407 763 881</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.3-1 : Evaluation des critères de Liu et Borsotti.
(Source, E.Glory[1])

Les segmentations $a$, $b$ sont obtenues en seuillant respectivement la composante R et B de l’espace RGB. La segmentation $c$ a été obtenue en délimitant manuellement les régions de couleur homogène et les segmentations $d$ et $e$ sont des segmentations erronées introduites.
volontairement. La première pour illustrer le cas où chaque pixel est une région (sur-segmentation maximale), la deuxième pour illustrer une sous-segmentation maximale (l'image entière constitue la seule région segmentée).

Le classement Référence a été réalisé par un biologiste et sert de modèle. On remarque que le critère de Liu positionne la segmentation d en première position (faiblesses de ce critère). En revanche, le critère de Borsotti place cette segmentation en dernière position mais place également le cas de non-segmentation en première position. Ceci provient du terme $\sqrt{R}$ qui pénalise les segmentations présentant plusieurs régions. Enfin les deux critères placent les segmentations $a$, $b$ et $c$ dans le même ordre, qui ne correspond pas à celui du biologiste.

Pour résumer, les deux critères ont tendance à favoriser les images comprenant peu de régions. Pour pallier à ce problème, la première solution proposée par E. Glory consiste à supprimer le terme $\sqrt{R}$ du critère de Borsotti, vu qu’a priori nous n’avons pas d’informations sur le nombre de régions à trouver.

Dans ce cas, le classement obtenu est le suivant :

$$C_1 = \frac{1}{1000A} \sum_{i=1}^{R} \left[ \frac{e_i^2}{1 + \log A_i} + \left( \frac{N(A_i)}{A_i} \right)^2 \right]$$

(On a la somme sur $R$)

(Source, E.Glory[1])

Une deuxième modification apportée par E.Glory au critère $C_1$ est de calculer le terme $e_i$, représentant l’hétérogénéité des couleurs, dans les espaces uniformes L’$a^*b^*$ et L’$u^*v^*$. Dans ces espaces, définis par la CIE, les distances calculées entre couleurs correspondent aux différences perçues par l’œil humain. Grâce à cette seconde modification, le classement donné par le nouveau critère $C_2$ correspond au classement de référence.

$$C_2 = \frac{1}{1000A} \sum_{i=1}^{R} \left[ \frac{e_i^2}{1 + \log A_i} + \left( \frac{N(A_i)}{A_i} \right)^2 \right]$$

Glory propose finalement le critère $C$ où elle normalise les deux termes de la somme. Le premier terme est normalisé selon la dispersion des couleurs de l’image entière :

$$d_i = \frac{e_i^2}{1 + \log A_i}$$

$$d_{image} = \frac{e^2}{1 + \log A}$$
Chapitre 4 – Etat de l’art

\[ h = \frac{\sum_{i=1}^{R} d_i}{1 + d_{image}} \]

En ce qui concerne la normalisation du deuxième terme, E. Glory propose de l’exprimer sous la forme suivante :

\[ \sum_{i=1}^{R} \left( \frac{N(A_i)}{A_i} \right)^2 = \sum_{A_k = A_{\text{min}}}^{A_{\text{max}}} \frac{N(A_k)}{A_k^2} \]

où \( A_{\text{min}} \) et \( A_{\text{max}} \) sont respectivement les aires de la plus petite et de la plus grande région segmentée et \( N(A_k) \) le nombre de régions d’aire \( A_k \). Maintenant que la dispersion des couleurs est calculée dans un espace de couleur uniforme, E. Glory a déterminé empiriquement que la formulation la plus adéquate pour le second terme était :

\[ \sum_{A_k = A_{\text{min}}}^{A_{\text{max}}} \frac{N(A_k)}{A_k^2} \]

Enfin pour être comparable avec le terme \( h \), le terme relatif à la présence de régions de petite taille, noté \( n \), est normalisé par le nombre de régions segmentées. On a donc :

\[ n = \frac{\sum_{A_k = A_{\text{min}}}^{A_{\text{max}}} \frac{N(A_k)}{A_k^2}}{R} \]

Au final, le critère \( C \) proposé par E. Glory est le suivant :

\[ C = h + n. \]

4.3.1.3.1 APPLICATIONS

Si nous avons pris le temps d’analyser en détails, la « construction » du critère \( C \), c’est parce que les applications que nous pouvons en faire sont extrêmement importantes. En effet, à partir de ce critère, il est possible, par exemple, de déterminer automatiquement l’espace de couleur qui convient le mieux.
Figure 4.3-2 : Critère C : Choix automatique de l’espace de couleur. 
(Source, E.Glory[1])

Un raisonnement identique peut également permettre de déterminer l’algorithme de segmentation à utiliser, où même de déterminer le meilleur compromis entre espace de couleur/algorithme de segmentation.

Enfin, si nous prenons le cas d’un algorithme de segmentation par seuillage, la valeur de seuil optimal peut également être déterminée via le critère $C$.

Figure 4.3-3 : Critère C : Choix automatique du seuil de segmentation. 
(Source, E.Glory[1])
4.3.3 LA SEGMENTATION D’IMAGES CYTOLOGIQUES EN COULEUR

Bien qu’ayant développé le critère $C$, E. Glory ne l’a pas exploité dans le logiciel qu’elle a développé, faute de temps. La segmentation qu’elle utilise est la méthode de seuillage de Ridler\textsuperscript{36}.

Néanmoins, le critère $C$ a quand même permis de déterminer que la composante verte de l’espace RGB normalisé était la meilleure composante à utiliser puisqu’elle obtient le meilleur score selon ce critère.

4.3.4 LA CLASSIFICATION DES COMPOSANTES CONNEXES

Avant de proposer une méthode de classification automatique, E. Glory émet deux hypothèses de travail :

1. « Sur les images de culture cellulaire à traiter, les composantes connexes correspondant aux noyaux isolés sont plus nombreuses que les composantes connexes correspondant aux agrégats. »

2. « Les noyaux isolés d’une même culture cellulaire ont une morphologie similaire, contrairement aux agrégats qui ont une forme différente les uns des autres. »

Dans notre cas, la deuxième hypothèse n’est pas toujours vérifiée car, contrairement à E. Glory, les images que nous avons reçues proviennent d’expériences non standardisées. Au sein d’une même expérience, il arrive que le focus et/ou le zoom varient. Il serait donc dangereux d’utiliser la méthode que nous allons décrire ci-dessous puisque, globalement, les images que nous avons à notre disposition présentent des caractéristiques plus diversifiées.

4.3.3.1 MÉTHODE UTILISÉE

E. Glory propose dans un premier temps de classer les composantes connexes sur base de leur surface respective. Une aire de référence $A_{\text{ref}}$, considérée comme l’aire d’un noyau isolé, est déterminée et la classification est réalisée de cette façon :

- Les composantes connexes dont l’aire est inférieure à la moitié de $A_{\text{ref}}$ sont considérées comme des artefacts et sont ignorées.
- Les composantes connexes dont l’aire est supérieure au double de $A_{\text{ref}}$ sont considérées comme des agrégats.
- Les composantes connexes dont l’aire est comprise entre la moitié et le double de $A_{\text{ref}}$ sont considérées comme des noyaux isolés.

Malheureusement, cette classification n’est pas adéquate car le taux de faux négatifs\textsuperscript{37} est trop important. En d’autres termes, la proportion d’agrégats qui ne sera pas soumise à l’algorithme de séparation est trop importante.

\textsuperscript{36} Le lecteur se rendra compte au chapitre suivant que nous avons également utilisé l’algorithme de Ridler comme algorithme initiale de segmentation. La justification est faite à la section 5.3.2

\textsuperscript{37} Cf. section 5.4
Finalement, E. Glory s’est tournée vers une méthode de classification supervisée non paramétrique avec un apprentissage automatique qui s’adapte aux caractéristiques des noyaux. Cette méthode repose sur le principe des méthodes à noyau, et plus particulièrement la fenêtre de Parzen, en calculant la densité de probabilité des prototypes.

Un prototype est une composante connexe définie par un vecteur contenant les informations suivantes :

- L’aire,
- Le périmètre,
- La longueur du grand axe et du petit axe de l’ellipse modélisant la composante connexe,
- L’elongation,
- La circularité.

La fenêtre de Parzen est une méthode d’apprentissage par voisinage, proche de la méthode des k plus proches voisins. Elle permet de réaliser une prédiction sur un objet $X$ en prenant en compte les objets dont la proximité avec $X$ sera jugée suffisante.

La différence entre la fenêtre de Parzen et la méthode des k plus proches voisins réside dans la notion de voisinage. Il est constant dans la méthode des k plus proches voisins (valeur de k) alors qu’il est défini par un noyau dans le cas de la fenêtre de Parzen. L’utilisation d’une fenêtre de Parzen est décrite par l’expression suivante :

$$
\hat{P}(y|X) = \frac{\sum_{i=1}^{n} K(X, X_i) \; \; y_i = y}{\sum_{i=1}^{n} K(X, X_i)}
$$

où $n$ représente le nombre d’objets disponibles dans la base de données (apprentissage), $i$ représente l’index d’un des objets de cette base de données, $X$ représente une donnée pour laquelle on souhaite faire une prédiction. $K(X, X_i)$ représente le calcul de la fonction noyau (K) entre l’objet $X$ et l’objet $X_i$. Enfin, $\hat{P}(y|X)$ représente la prédiction de la classe $y$ pour l’objet $X$. 
### 4.3.3.1.1 PHASE D’APPRENTISSAGE

La phase d’apprentissage permet de modéliser les paramètres des noyaux isolés. Au cours de cette étape, certaines composantes connexes vont être sélectionnées pour représenter les noyaux isolés, elles sont appelées « noyaux isolés de référence ».

Cette sélection est réalisée en fonction de valeurs de la densité de probabilité des composantes connexes calculées au moyen d’une fenêtre de Parzen, en ne conservant que les composantes dont la densité de probabilité est supérieure à 20% de la densité maximale.

### 4.3.3.1.2 PHASE DE PRÉDICTION

L’étape de prédiction consiste à classer chaque nouvelle composante connexe dans une des deux classes ($C_0$ ou $C_1$). Pour ce faire, la densité de probabilité de chaque composante est calculée en fonction des noyaux isolés de référence grâce à une fenêtre de Parzen.

Si la densité est non-nulle, la composante connexe est considérée comme un noyau isolé appartenant à la classe $C_0$. En revanche, si la densité est nulle, la composante sera considérée comme un agrégat appartenant donc à la classe $C_1$ car ses caractéristiques sont éloignées des caractéristiques des noyaux isolés de référence.

Avec la méthode qu’elle a développée, E. Glory est parvenu à avoir un taux de faux négatifs inférieur à 5%.
4.3.5 LA SEGMENTATION DES AGRÉGATS

E. Glory débute le chapitre de segmentation des agrégats en informant directement le lecteur que pour les images à forte densité cellulaire, il est impossible de distinguer les frontières des noyaux, même pour un biologiste. Par conséquent, ce type d’images ne doit pas être traité par sa méthode.

Naturellement, seules les composantes appartenant à la classe $C_1$ sont soumises à l’algorithme de segmentation des agrégats. Le schéma général de l’algorithme est le suivant :

![Schéma général de l'algorithme de segmentation des agrégats.](Source, E.Glory[1])

La première étape consiste à vérifier si la composante n’est pas, en réalité, un gros noyau isolé. Pour ce faire, sa morphologie est comparée à l’ellipse représentant au mieux la composante connexe. Ensuite, une proportion $d_{ellipse}$ est calculée représentant la différence de surface entre la composante connexe et l’ellipse. E.Glory a déterminé empiriquement que si cette valeur était inférieure à 15%, la composante est considérée comme un gros noyau.
La deuxième étape coïncide avec l'utilisation d’un algorithme de « séparation-fusion ».
Dans un premier temps, un algorithme de séparation segmente les agrégats en noyaux isolés. Cette étape implique une sur-segmentation du nombre de noyaux agrégés. C'est la raison pour laquelle, successivement à la séparation, un algorithme de fusion est appliqué.

L’étape de séparation utilise un algorithme récursif qui applique à la composante agrégée la ligne de partage des eaux. Ensuite, parmi les différentes régions obtenues, celle qui ressemble le plus à un noyau isolé de référence est soustraire de la composante et le processus est réappliqué récursivement à la composante restante jusqu’à ce que l’algorithme LPE ne trouve plus de régions à segmenter.

L'ensemble des parties issues de la séparation d’une composante connexe est alors soumis à un algorithme de fusion afin de trouver la partition la plus vraisemblable de noyaux. La fusion repose sur la construction d’un graphe d’adjacence afin de construire l’ensemble des parties possibles de la partition et est déterminée par la minimisation de la fonction suivante :

$$f_{fusion} = \frac{\sum_{i=1}^{n}|A_i - A_{ref}|}{n} + \sigma_A$$

où $A_i$ représente l’aire de la région $i$, $A_{ref}$ l’aire du noyau individuel de référence, $n$ le nombre de région après la fusion et $\sigma_A$ l’écart-type des aires de toutes les régions $A_i$. Le gros désavantage de cette méthode est qu’elle est extrêmement gourmande en temps de calcul puisqu’il faut générer l’ensemble des parties et l’ensemble des fusions possibles ! Voici un exemple du travail à effectuer :

---

38 La ressemblance avec un noyau de référence est calculée grâce à la densité de probabilité estimée par la fenêtre de Parzen.
Enfin, après l’application de l’algorithme « séparation-fusion », un seuillage local est réalisé afin de prendre en compte le cas où l’agrégat est composé d’une partie cytoplasmique et d’une partie nucléaire. Cette approche permet de retrouver les pixels appartenant aux noyaux.
4.3.6 CONCLUSIONS E.GLORY

Une des principales contributions du travail d’E.Glory a été la proposition d’un critère normalisé qui évalue comparativement la qualité d’une segmentation d’images en couleur. Ce critère a permis de déterminer la meilleure composante à seuiller dans le cas d’images cytologiques, en l’occurrence la composante verte de l’espace RGB normalisé. Nous avons également constaté, dans notre cas, que cette composante était la plus pertinente.

Vient ensuite la méthode de classification. Comme nous l’avons déjà évoqué, E. Glory utilise une classification supervisée reposant sur le principe des méthodes à noyau. La densité de probabilité est calculée dans l’espace des caractéristiques (telles que l’aire, le périmètre,...). Dans notre cas, cette approche est un peu plus dangereuse vu que les caractéristiques des cellules d’une expérience $Y$ ne sont pas forcément semblables aux caractéristiques des cellules de l’expérience $X$.

Enfin, sa méthode de segmentation des agrégats (séparation-fusion) s’avère très robuste mais excessivement coûteuse en temps de calcul. En effet, il s’agit, pour chaque composante connexe, de déterminer et tester l’ensemble des partitions possibles à partir d’une composante-agréga­t. Pour notre application, nous utiliserons une méthode beaucoup plus rapide basée sur la recherche des marqueurs d’un algorithme de ligne de partage des eaux.

Les résultats obtenus via la méthode d’E.Glory sont excellents puisqu’il faut compter, en général, un taux d’erreur proche de 1% (même si les images qu’elle analyse sont moins variées que les nôtres). Nous n’avons pas la prétention d’obtenir d’aussi bons résultats que cette thèse de doctorat, mais nous verrons au chapitre 6 que la méthode que nous avons développée tient tout à fait la route.
CHAPITRE 5

DÉVELOPPEMENT DE L’OUTIL DE COMPTAGE AUTOMATIQUE

5.1 INTRODUCTION

Ce chapitre est consacré à l’étude et à la justification des techniques mises en œuvre lors du développement de l’outil de comptage automatique. Nous rappelons au lecteur que le logiciel se veut aussi générique que possible. Autrement dit, la précision des résultats de comptage doit, autant que faire se peut, être indépendante du type d’images traitées.

D’une manière générale, nous avions deux possibilités :

1. Soit arriver à identifier les propriétés générales et répétibles des cellules, quelles que soient les conditions d’acquisition, et encoder ces caractéristiques dans le logiciel à développer.

2. Soit imaginer un logiciel adaptatif qui apprend les caractéristiques des cellules et propose une méthode de comptage basée sur ces modèles. Le programme conserverait ces paramètres pour permettre de les appliquer ultérieurement sur plusieurs autres images.

C’est cette deuxième approche que nous avons retenue car le risque de la première est qu’un futur chercheur propose des images où les conditions d’acquisition et le matériel employé soient fort différents des propriétés déjà définies.

5.1.1 CHOIX D’IMPLÉMENTATION

Le programme a été développé en Java sous la forme d’un plugin ImageJ. Ce choix a été motivé par le fait que le logiciel ImageJ offre une vaste bibliothèque d’outils de traitement d’images et, surtout, qu’il est « open-source », c’est-à-dire que le code est en accès libre et qu’il peut, au besoin, être modifié.

De plus, l’implémentation en Java rend le logiciel utilisable sur différents systèmes d’exploitation. Nous le verrons par la suite, le choix du langage Java a aussi facilité l’interfaçage du plugin avec le code source du classificateur automatique PixIT, également écrit en Java.
5.2 STRATÉGIE GÉNÉRALE

Nous rappelons ici au lecteur la chaîne de traitement générale pour l’acquisition et le dénombrement des images.

A présent, intéressons-nous à la chaîne de traitement de l’outil de comptage automatique illustrée par le schéma suivant :

Figure 5.2-1 : Chaîne de traitement des images.

Figure 5.2-2 : Stratégie générale pour le dénombrement de cellules.
La figure précédente présente la stratégie adoptée. La première étape consiste à distinguer les objets en couleurs, idéalement les cellules, de l’arrière plan. Comme nous l’avons évoqué au chapitre 2, lorsque l’image traitée entre dans la catégorie des « cas favorables », cette segmentation est suffisante pour dénombrer les noyaux.

En revanche, lorsque la densité des cellules est importante ou que l’image présente de nombreux artefacts (pores, taches, …), une méthode de classification automatique est nécessaire afin de trier les composantes détectées en trois catégories :

1. **Noyau isolé**
2. **Agrégats**
3. **Autres**

La classe *Autres* comprend toutes les composantes qui ne sont considérées ni comme *Noyau seul*, ni comme *Agrégats*. Il s’agit principalement des pores, taches et autres parasites.

**Remarque** : Bien que les expériences ne soient pas standardisées, nous avons décidé de rejeter toutes les composantes dont l’aire est inférieure à 100 pixels. Au vu des images traitées, cette limite inférieure convient parfaitement.

Après classification, une méthode de segmentation est appliquée uniquement sur les images de la classe *Agrégats* dans le but de distinguer les différents noyaux présents dans ces amas de cellules. Une fois cette opération terminée, nous sommes en mesure d’évaluer le nombre de cellules présentes sur l’image.

Pour terminer, nous offrons la possibilité à l’utilisateur de pouvoir éditer les résultats. Cette étape de correction, illustrée à la section 5.6, permet non seulement d’améliorer la qualité du dénombrement mais, surtout, d’exporter les composantes corrigées, ce qui permettra, au fil des expériences menées par les chercheurs, de renforcer le modèle du classificateur automatique. En effet, certaines composantes pourraient être correctement classées mais avec une faible certitude, les inclure dans l’ensemble d’apprentissage devrait rendre les modèles plus confiants.

Cette dernière étape est donc nécessaire et ne constitue certainement pas un « abandon » de l’idée de départ qui, pour rappel, était d’automatiser le plus possible le dénombrement.

### 5.3 SEGMENTATION INITIALE

L’étape de segmentation initiale a pour but de binariser l’image originale afin d’identifier les différentes régions d’intérêts (ROI). Ces régions sont considérées comme régions pertinentes de l’image et sont les seules à être traitées. Par convention, les pixels de chaque ROI binaire prendront la valeur 0 (noir) alors que les autres pixels auront la valeur 255 (blanc).
Nous avons testé et évalué quatre méthodes de segmentation différentes. La première utilise un seuillage manuel tandis que les trois autres appliquent l'algorithme de Ridler sur différentes composantes de couleurs. L'utilisation de cet algorithme est justifiée au point 5.3.2.

Le choix des techniques de seuillage comme étape de segmentation est guidé par l'utilisation d’algorithmes de faible complexité calculatoire. L’objectif étant de pouvoir traiter rapidement plusieurs dizaines, voire centaines d’images.

5.3.1 SEUILLAGE MANUEL

La première méthode testée est celle décrite dans le chapitre « Analyse des images », section 2.1.1. Pour rappel, le seuil est fixé « manuellement » en fonction du colorant utilisé lors des expériences. Evidemment cette méthode n’est pas automatique et ne s’adapte pas à d’autres images.

5.3.2 SEUILLAGE AUTOMATIQUE

Afin d’utiliser l’algorithme de seuillage le plus adapté à notre application, nous avons testé différentes techniques de seuillage automatique grâce au logiciel\(^{39}\) développé par Sankur et Sezgin [16].

Ce test a été réalisé sur base de 10 images représentatives des échantillons d’images mis à notre disposition. Pour chaque image, un ensemble \(E\) de valeurs de seuils \(t_m\), produisant des segmentations satisfaisantes, est déterminé grâce à une méthode de *seuillage multiple* développée dans ledit logiciel. Ensuite la distance \(d\) entre les valeurs de \(E\) et le seuil \(t_a\) trouvé automatiquement est calculée comme suit :

\[
d = \min_{t_m \in E} (|t_m - t_a|)
\]

L’ensemble de ces distances est rapporté dans le tableau ci-dessous. Comme nous le constatons, l’algorithme de Ridler, décrit dans la section 3.4.4.1, minimise la somme et l’écart-type des distances entre les seuils manuels et les seuils trouvés automatiquement. De plus, il s’agit d’un des algorithmes de segmentation automatique les plus simples et les plus rapides, c’est pourquoi nous avons décidé d’utiliser pour l’étape initiale de segmentation.

\(^{39}\) Ce logiciel est téléchargeable à l’adresse suivante :
http://www.busim.ee.boun.edu.tr/~sankur/SankurFolder/OTIMEC_INg.zip
5.3.3 CHOIX D’UNE COMPOSANTE DE COULEUR

Comme nous l’avons déjà évoqué, l'algorithme de Ridler utilise l'histogramme d'une image pour déterminer un seuil. Deux images présentant le même histogramme auront la même valeur de seuillage. Pour une image en couleur, nous avons jugé intéressant de déterminer quelle composante donnera l'histogramme le plus pertinent à seuiller.

5.3.3.1 PROJECTION DE L’ESPACE RGB EN NIVEAUX DE GRIS

L'idée consiste à projeter les valeurs des pixels de l'espace RGB (3 dimensions) sur un espace à une dimension. Cette opération consiste en une combinaison linéaire des valeurs des trois composantes RGB.

La projection la plus simple revient à calculer la moyenne de chaque composante. Soient $P$ la valeur projetée et $R, G, B$ les valeurs de la composante rouge, verte et bleue respectivement de l'espace RGB, on a :

$$ P = \frac{R + G + B}{3} $$

Cependant, cette heuristique ne prend pas en compte la perception des couleurs par l'œil humain. En effet, nous distinguons trois types de cônes selon le pigment qu’ils contiennent. Ces cônes ont donc une sensibilité à des ondes lumineuses de longueurs différentes, comme en témoigne la figure ci-dessous :

---

40 Cellules de l’œil sensibles à la couleur.
41 L’erythropsine (sensibles au rouge), de la chloropsine (vert), de la cyanopsine (bleu).
Chapitre 5 – Développement de l’outil de comptage automatique

C'est pourquoi la C.I.E propose de caractériser la valeur de luminance d'un pixel (niveau de gris) comme suit :

- Selon la recommandation 709,
  \[ P = 0.2125 R + 0.7154 V + 0.0721 B \]

- Selon la recommandation 601,
  \[ P = 0.299 R + 0.587 G + 0.114 B \]

Dans les deux cas, la somme des coefficients est égale à 1. Pour une image numérique, il importe de savoir si les valeurs R, G et B des pixels sont linéaires par rapport aux intensités lumineuses mesurées (recommandation 709), ou si, au contraire, ces valeurs ont subi une correction « gamma »42 (recommandation 601), ce qui est généralement le cas pour des images acquises par microscope électronique.

Ainsi, nous avons adopté la recommandation 601 comme méthode de projection de l'image couleur en niveaux de gris. Remarquons qu’ici, nous ne prenons en considération aucune information de couleur, la technique est totalement générique.

### 5.3.3.2 COMPOSANTE VERTE DE L’ESPACE RGB NORMALISÉ

Comme nous l’avons vu au chapitre précédent, la composante verte de l’espace RGB normalisé (cf. section 3.4.2.1) présente l’histogramme le plus pertinent à seuiller pour des images de cellules colorées au « Giemsa ». En effet, cette composante minimise la valeur moyenne du critère \( C \).

Pour rappel, la valeur \( g \) de la composante verte de l'espace Nrgb est définie, à partir des valeurs R, G et B de l'espace RGB, comme suit :

\[ g = \frac{G}{R + G + B} \]

42 Plus de détails sur la correction gamma : http://www.w3.org/TR/PNG-GammaAppendix.html
Nous le verrons dans le chapitre suivant, l’utilisation de cette composante, appelons la \textit{Nrgb2}, donne la segmentation la plus représentative et la plus précise des cellules. Malheureusement, nous faisons ici l’hypothèse que le colorant utilisé est du « Giemsa ». Si tel n’était pas le cas, l’utilisation de \textit{Nrgb2} ne serait plus justifiée.

Cependant, d’un point de vue pratique pour les chercheurs, spécialement ceux du GIGA, et parce que le « Giemsa » est couramment utilisé comme colorant au sein de multiples expériences, nous avons jugé opportun de proposer l’usage de cette composante dans le processus de segmentation.

5.3.3.3 \textbf{ANALYSE EN COMPOSANTES PRINCIPALES}

Nous l’avons constaté, il n’est pas trivial de choisir de façon optimale une composante de couleur pertinente. Nous sommes confrontés à un dilemme entre précision de la segmentation et indépendance de la méthode face au colorant utilisé. C’est pourquoi, suivant les conseils de Sébastien Piérard, nous nous sommes tournés vers une technique d’\textit{analyse en composantes principales}.

D’une manière générale, l’analyse en composantes principales est une méthode mathématique d’analyse des données utilisée comme technique de réduction de dimensionnalité.

Soit un nuage de points de \( N \) réalisations chacune définie par \( P \) variables. La représentation matricielle est la suivante :

\[
M = \begin{bmatrix}
X_{1,1} & \cdots & X_{1,P} \\
\vdots & \ddots & \vdots \\
X_{N,1} & \cdots & X_{N,P}
\end{bmatrix}
\]

Admettons que ces \( P \) variables soient linéairement indépendantes, nous avons alors besoin de chacune d’entre elles pour prendre en compte toute la variabilité du nuage de points. L’objectif de l’ACP est de décrire un maximum de cette variabilité avec \( Q < P \) variables. Ces \( Q \) variables sont des combinaisons linéaires des variables originales et portent le nom de composantes principales. L’ACP tente donc de minimiser la perte d’information due à la réduction du nombre de variables.

5.3.3.3.1 \textbf{ALGORITHME UTILISÉ}

Nous avons implémenté la technique de transformation de Karhunen-Loève[23]. L’idée de cette transformation est de réaliser un changement de base afin de trouver un axe \( u \), issu d’une combinaison linéaire des \( X_p \), tel que la variance du nuage autour de cet axe soit maximale. Plus la variance est importante et plus nous sommes aptes à séparer les données.

Dans notre cas, \( P = 3 \) et \( Q = 1 \). En effet, nous devons projeter les valeurs d’un espace à 3 dimensions (RGB) dans un espace à 1 dimension (niveaux de gris). Le nombre de réalisations \( N \)
correspond au nombre de pixels dans l'image\textsuperscript{43}. L’image traitée est donc vue comme un nuage de $N$ pixels dans l’espace RGB.

La première étape de l’algorithme consiste à organiser l’image de dimension $(L \times C)$ sous forme matricielle de dimension $(N \times 3)$, où $N = L \times C$. Chaque ligne de la matrice $M$ représente la décomposition d’un pixel de l’image selon ses composantes R, G et B.

\[
M = \begin{bmatrix}
    X_{1,R} & X_{1,G} & X_{1,B} \\
    \vdots & \vdots & \vdots \\
    X_{N,R} & X_{N,G} & X_{N,B}
\end{bmatrix}
\]

Une fois la matrice $M$ créée, la moyenne de chaque colonne est calculée, définissant ainsi la matrice $Mean$ de dimension $(1 \times 3)$ :

\[
Mean = [mean(M(:,1)) \ mean(M(:,2)) \ mean(M(:,3))]
\]

Afin de centrer la matrice $M$, on soustrait chaque élément $X_{i,j}$ par sa moyenne respective. On obtient ainsi la matrice $B$, de dimension $(N \times 3)$, telle que :

\[
B = M - Mean \times h
\]

où $h$ est un vecteur de dimension $(1 \times N)$ dont tous les éléments ont la valeur 1. Il est important de centrer la matrice $M$ dans le but de trouver une base qui minimise l’erreur quadratique moyenne\textsuperscript{44} des données.

\textsuperscript{43} Pour avoir un ordre de grandeur, les images traitées ont une définition de 1360 x 1024 pixels, ce qui donne un total de $N = 1392640$ pixels.

\textsuperscript{44} Cf. Chapitre 6 : « Tests et évaluations ». 
Nous calculons ensuite la matrice de covariance $C$, de dimension $(3 \times 3)$ de la manière suivante :

\[ C = \frac{1}{N} \ast (B^T \ast B) \]

Notons qu’en statistiques, la covariance permet de mesurer le degré d’indépendance entre 2 variables et qu’elle est définie comme suit :

\[ COV(X, Y) \equiv E[(X - E[X]) \ast (Y - E[Y])], \]

ce qui correspond tout à fait à la définition de la matrice $C$ ($E$ étant l’espérance mathématique).

La covariance est positive pour chaque couple de valeurs qui diffèrent de leur moyenne dans le « même sens » et négative pour chaque couple de valeurs qui diffèrent de leur moyenne dans le « sens opposé ». Une matrice de covariance peut être considérée comme la généralisation, dans un espace multidimensionnel, du concept de variance d’une variable aléatoire.

L’étape suivante consiste à calculer les valeurs et vecteurs propres de la matrice $C$. Remarquons que la matrice $C$ est symétrique$^{45}$ et à valeurs réelles. Nous pouvons donc écrire

\[ C = V D V^T \quad (1) \]

où $V$ est une matrice orthogonale de dimension $(3 \times 3)$ dont les colonnes correspondent aux vecteurs propres de $C$ et où $D$ est une matrice diagonale (donc diagonalisable$^{46}$) dont les éléments diagonaux sont les valeurs propres de $C$. L’équation (1) peut aussi s’écrire

\[ D = V^{-1}CV \]

vu que $V$ est orthogonale$^{47}$. Les colonnes de $D$ et les colonnes respectives de $V$ sont triées dans l’ordre décroissant des valeurs singulières.

Finalement, les valeurs projetées sont définies par la matrice $Y$, de même dimension que la matrice $M$ de départ, telle que

\[ Y = (V^T \ast B^T)^T \]

La première colonne de $Y$ correspond en réalité à l’axe $u$ défini au début de cette sous-section et pour lequel la variance des valeurs est maximale. La deuxième colonne correspond à l’axe définissant le maximum de variabilité restante, etc.

Pour notre application$^{48}$, nous utilisons donc le vecteur $Y_1$, défini par les valeurs de la première colonne de la matrice $Y$, comme composante de couleur.

---

45 Une matrice $A$ est dite symétrique lorsqu’elle est égale à sa transposée, i.e. $A = A^T$. Une telle matrice est donc obligatoirement carrée.

46 Une matrice carrée $A$ (de dimension $n \in \mathbb{N}^*$) est diagonalisable si il existe une matrice inversible $B$ et une matrice diagonale $D$ telles que $A = BDB^{-1}$.

47 Une matrice carrée $A$ est orthogonale si $A^T \ast A = I$, où $I$ est la matrice identité.

48 Toutes les opérations matricielles ont été implémentées grâce à la librairie JAMA. (Java Matrix, http://math.nist.gov/javanumerics/jama/)
5.3.4 BINARISATION DE L’IMAGE

Pour résumer, nous avons détaillé quatre techniques de segmentation possibles :

1. Seuillage manuel
2. Seuillage automatique
   a. Recommandation 601 de la C.I.E
   b. Composante \( Nrgb2 \)
   c. Composante \( Y_1 \)

Pour la méthode 1, la binarisation de l’image est simple. Si les valeurs des composantes R, G et B sont comprises dans leurs intervalles respectifs, le pixel prend la valeur 0, sinon la valeur 255.

Pour les méthodes de seuillage automatique, nous appliquons préalablement un filtre médian à l’image en niveau de gris afin d’éliminer un maximum de pores. Le filtre médian utilisé est un disque dont nous avons fixé le rayon à 5 pixels. Chaque pixel de l’image est donc remplacé par la valeur médiane des pixels définis par l’élément structurant suivant :

![Figure 5.3-5 : Elément structurant du filtre médian. (Rayon = 5 pixels)](image)

Sur base du seuil retourné par la méthode de Ridler, l’étape de binarisation est alors appliquée à cette image filtrée. Les pixels inférieurs au seuil prennent la valeur 0 et ceux supérieurs prennent la valeur 255.

Pour les méthodes 2a et 2b, ce seuil à une valeur \( v \in \mathbb{N} \) comprise entre [0 et 255] puisque les composantes de couleur utilisées utilisent des histogrammes définis dans cet intervalle.

En revanche, pour la méthode 2c, les valeurs de \( Y_1 \) appartiennent à \( \mathbb{R} \) et ne sont pas bornées dans l’intervalle [0,255]. Les normaliser dans cet intervalle n’est pas une solution car cela modifie la variance et ne conserve dès lors pas la propriété du résultat de la ACP. C’est pourquoi, pour la méthode 2c, nous avons été amenés à construire un histogramme sur mesure à partir des valeurs de \( Y_1 \).

---

49 Les mêmes intervalles que ceux définis à la section 2.1.1.
50 Différents tests ont été effectués afin de définir la taille du rayon. Il en résulte qu’un rayon de 5 pixels est adéquat pour les images que nous traitons.
5.3.4.1 CONSTRUCTION D’UN HISTOGRAMME

La première opération à réaliser est de déterminer le nombre d’intervalles, aussi appelés classes, de l’histogramme. Le nombre de classes dépend du nombre total de réalisations, c’est-à-dire du nombre de pixels de l’image, à savoir $N$. Une heuristique courante est de considérer que le nombre de classes $S$ est égal à :

$$S = \sqrt{N}$$

Ensuite, il faut définir la largeur $L$ de chaque classe. Pour ce faire, il faut considérer l’amplitude $K$ de l’histogramme. On a :

$$K = valeur\ max - valeur\ min$$

$$L = \frac{K}{S}$$

Chacune des classes correspond à un intervalle fermé à gauche et ouvert à droite. La $valeur\ max$ est donnée par la valeur maximale observée dans $Y_1$ (arrondie à l’entier supérieur) tandis que la $valeur\ min$ est donnée par la valeur minimale observée dans $Y_1$ (arrondie à l’entier inférieur).

Enfin, sur base des valeurs de $Y_1$, il suffit de comptabiliser le nombre de réalisations observées pour chacune des classes.

Figure 5.3-6 : A gauche l’image originale. A droite l’histogramme construit sur base des valeurs de $Y_1$ relatives à l’image originale.
5.4 CLASSIFICATION AUTOMATIQUE DES COMPOSANTES CONNEXES

Pour l’étape de classification nous avons utilisé la librairie « iclass.jar » développée par Raphaël Marée et sur laquelle se base la méthode de classification du logiciel PixIT (cf. section 3.3.3).

Afin de pouvoir utiliser cette librairie, il faut tout d’abord concevoir une base de données, représentant l’ensemble d’apprentissage. Ce dernier consiste en un classement des composantes connexes que nous avons du trier nous-mêmes en trois catégories, à savoir :

1. La classe des noyaux isolés
2. La classe des artefacts
3. La classe des agrégats

A l’heure actuelle, l’ensemble d’apprentissage contient 2915 composantes réparties comme suit :

- 1145 composantes pour la classe « Noyau isolé »
- 930 composantes pour la classe « Agréat »
- 840 composantes pour la classe « Artefacts »

Une fois ce classement effectué, il est alors possible de passer à la phase d’apprentissage qui permettra de créer un modèle sur base duquel les composantes seront automatiquement classées lors de la phase de prédiction. Nous allons maintenant présenter au lecteur les paramètres que nous avons fixés pour ces deux phases.

5.4.1 PHASE D’APPRENTISSAGE

Sur base de ce qui a été dit à la section 3.3.3.1, la phase d’apprentissage extrait aléatoirement $N_{sw} = 100$ sous-fenêtres, pour chaque image. Chacune de ces $N_{sw}$ sous-fenêtres a une taille représentant entre 50 et 100% de la taille originale de l’image et est ensuite redimensionnée en une image de 16x16 pixels.

Nous pouvons donc représenter chaque sous-fenêtre par un vecteur de 768 attributs (16x16x3) dans la base de données. En effet, chaque pixel est défini par ses trois composantes de couleur, dans l’espace colorimétrique HSI. Nous avons choisi l’espace HSI car, comme nous le verrons à la section suivante, le test en validation croisée offre de meilleurs résultats en HSI qu’en niveaux de gris. Bien sur, nous perdons un peu de généricité mais cela est nécessaire pour obtenir de bons résultats lors du dénombrement.

Enfin, en utilisant la méthode d’Extra-Trees, nous construisons un ensemble de $T = 10$ arbres aléatoires qui serviront de modèle pour la phase de prédiction. Pour chacun de ces arbres, lors de leur construction, et plus particulièrement lors de la détermination d’un nouveau nœud test, $N_{bpart} = 96$ partitions sont générées aléatoirement. La valeur de ce paramètre est relativement arbitraire mais elle a tout de même un sens: ni trop petite (pour filtrer les variables non pertinentes), ni trop grande (pour éviter des temps de construction de modèle trop longs).

---

51 Par « image », nous entendons évidemment l’image représentant la composante et non l’image sur laquelle nous devons faire le dénombrement.
52 Le redimensionnement est réalisé par interpolation bilinéaire, grâce aux librairies Java standards.
5.4.1.1 VALIDATION CROISÉE

Afin de valider la valeur des paramètres présentés ci-dessus, nous avons utilisé la méthode de validation croisée. Cette méthode consiste à choisir un certain nombre d'objets de l'ensemble d'apprentissage (ou une proportion par rapport au nombre d'objets) dans le but de créer un modèle. Ensuite, il suffit de propager le reste des objets dans ce modèle.

Dans notre cas, nous avons effectué 10 tests en validation croisée. Chaque test utilise 80% des images de chaque classe comme ensemble d'apprentissage et les 20% restants sont testés. Le taux d'erreur final, $T_{CV}$, est obtenu en moyennant le taux d'erreur obtenu pour chacun de ces 10 tests. Pour notre application, le taux d'erreur sur l'ensemble d'apprentissage est :

$$T_{CV1} = 5,76\% \text{ dans l'espace HSI}$$
$$T_{CV2} = 9,86\% \text{ en niveaux de gris}$$

Au vu de ces deux résultats, nous avons décidé, pour l'étape de classification, de travailler uniquement dans l'espace de couleur HSI. Afin d'analyser plus précisément cette valeur de $T_{CV1}$, il est judicieux d'analyser la matrice de confusion finale. Cette matrice permet de se faire une idée plus précise de la qualité de classification des composantes lors du test en validation croisée. Dans notre cas, nous avons observé :

<table>
<thead>
<tr>
<th>Class</th>
<th>Model Output</th>
<th>Agrégat</th>
<th>Artefact</th>
<th>Noyau</th>
<th>TOTAL</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agrégat</td>
<td>1696</td>
<td>24</td>
<td>140</td>
<td>1696/1660</td>
<td>91,18 %</td>
<td></td>
</tr>
<tr>
<td>Artefact</td>
<td>84</td>
<td>1593</td>
<td>13</td>
<td>1583/1680</td>
<td>94,23 %</td>
<td></td>
</tr>
<tr>
<td>Noyau</td>
<td>87</td>
<td>17</td>
<td>2186</td>
<td>2186 / 2299</td>
<td>95,45 %</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>1867</td>
<td>1624</td>
<td>2339</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.4-1 : Matrice de confusion après validation croisée.
L’interprétation de cette matrice est simple. Elle peut se faire par ligne ou par colonne. Par exemple, pour la ligne « Agrégat », il faut lire :


Pour la colonne « Agrégat », il faut plutôt lire :

« 1867 composantes de l’ensemble d’apprentissage ont été classées comme ‘Agrégat’, alors que 1696 sont effectivement des agrégats ».

On constate donc d’une manière générale que les artefacts qui sont mal classés sont plutôt considérés comme des agrégats plutôt que des cellules, ce qui est positif puisque lorsque cette composante sera traitée par l’algorithme de séparation des agrégats, nous pouvons raisonnablement espérer qu’il ne sera pas comptabilisé.

On remarque également qu’un noyau qui est mal classé est souvent considéré comme un agrégat, ce qui n’est pas contraignant puisqu’il sera également soumis à l’algorithme de séparation des agrégats.

Enfin, on s’aperçoit que les agrégats dont le classement est également erroné sont équitablement répartis entre les classes « Noyaux » et « Artefacts ». Un mauvais classement d’une composante « agrégat » est plus embarrassant dans la mesure où cet agrégat sera, dans le meilleur des cas, comptabilisé comme un noyau. Dès lors, il est temps de définir les notions de faux positifs et faux négatifs.

5.4.2 PHASE DE PRÉD I C T I O N

Afin d’évaluer la qualité de la classification automatique, les taux de faux positifs et faux négatifs sont calculés par rapport à un étiquetage manuel. L’étiquetage manuel classe les composantes en 2 classes :

1. $C_0$ = la classe des noyaux isolés,
2. $C_1$ = la classe des agrégats.

Les taux de faux positifs $F_p$ et négatifs $F_n$ sont définis comme ceci :

\[
F_p = \frac{\text{nombre de noyaux isolés retenus}}{\text{nombre total de noyaux isolés}} \times 100
\]

\[
F_n = \frac{\text{nombre d’agrégats rejetés}}{\text{nombre total d’agrégats}} \times 100
\]

$F_p$ reflète donc le nombre de noyaux isolés appartenant à la classe $C_1$, tandis que $F_n$ reflète le nombre d’agrégats appartenant à la classe $C_0$. Pour bien faire, il faudrait que $F_n$ soit aussi proche que possible de la valeur zéro afin d’éviter que des agrégats soient classés dans la classe des noyaux isolés.

En revanche, la contrainte sur la valeur $F_p$ peut être moins exigeante puisque, normalement, un noyau isolé considéré comme un agrégat sera comptabilisé comme « 1 » par la méthode de comptage des agrégats.
Nous avons mesuré ces deux taux sur base de neuf images représentatives de l'ensemble des images mises à notre disposition. Les résultats sont les suivants :

<table>
<thead>
<tr>
<th>Images</th>
<th>Faux positifs (%)</th>
<th>Faux négatifs (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.jpg</td>
<td>7,97</td>
<td>2,32</td>
</tr>
<tr>
<td>108.jpg</td>
<td>34,46</td>
<td>4,64</td>
</tr>
<tr>
<td>B52.3_B2.2.2.jpg</td>
<td>14,03</td>
<td>3,03</td>
</tr>
<tr>
<td>BB94_10_5M.jpg</td>
<td>6,79</td>
<td>4,69</td>
</tr>
<tr>
<td>IK9-10-5M(21).jpg</td>
<td>1,33</td>
<td>4,65</td>
</tr>
<tr>
<td>si3_bFGF_B6.jpg</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>svl2-10-8M(5).jpg</td>
<td>1,45</td>
<td>3,19</td>
</tr>
<tr>
<td>Tv6.tif.jpg</td>
<td>18,02</td>
<td>6,92</td>
</tr>
<tr>
<td>VEGF_h_25_A(7).jpg</td>
<td>22,49</td>
<td>9,93</td>
</tr>
<tr>
<td><strong>Moyenne</strong></td>
<td><strong>11,83</strong></td>
<td><strong>4,04</strong></td>
</tr>
</tbody>
</table>

D'après le tableau, nous remarquons que le classificateur, lorsqu'il se trompe, aura plutôt tendance à considérer les noyaux comme des agrégats plutôt que l'inverse. C'est évidemment ce que nous cherchons afin de minimiser le taux de faux négatifs. Nous remarquons ici toute l'importance et la difficulté de créer un ensemble d'apprentissage de départ qui soit pertinent !

Nous pouvons considérer que les résultats de classification sont encourageants pour la suite des opérations. De plus la phase finale d'édition/correction permettra de renforcer le modèle du classificateur et donc de diminuer petit à petit les valeurs des deux taux calculés.

5.5 SEGMENTATION DES AGRÉGATS

Nous venons de le voir, grâce à l’étape de classification automatique, les composantes sont triées en trois catégories. Nous sommes alors en mesure de traiter chacune de ces trois classes comme il se doit.

Pour la classe « Artefacts », aucun traitement n’est réalisé, les composantes sont simplement ignorées. Pour la classe « Noyaux seul », chaque composante est considérée comme un noyau isolé, comptant pour « un » lors du dénombrement. Enfin, comme on peut s'y attendre, un traitement particulier est réservé aux composantes de la classe « Agrégats ».

A travers le terme « Agrégats », nous voulons évidemment exprimer l'idée d'un regroupement de noyaux isolés mais, comme nous l'avons remarqué lors de la création de l’ensemble d’apprentissage, il est difficile de caractériser leur forme ou leur taille moyenne tant ils diffèrent d'une expérience à l'autre. Pour preuve, ces quelques images :

Figure 5.5-1 : Forme et taille diverses des agrégats.
Chapitre 5 – Développement de l’outil de comptage automatique

5.5.1 TECHNIQUE DÉVELOPPÉE

5.5.1.1 INTRODUCTION

En biologie, la séparation desagrégats cellulaires est un problème récurrent. En effet, il arrive fréquemment que les objets étudiés se touchent. Généralement, les solutions développées reposent sur des connaissances a priori de la taille et la forme de ces objets.

Pour les raisons invoquées précédemment, ces approches [24][25] basées sur la morphologie des objets n’ont pas été retenues. De même, l’étude des contours n’est pas totalement appropriée vu que les composantes présentent des contours parfois imprécis et pas toujours bien définis, dus à un mauvais focus utilisé ou simplement dus à un faible grossissement. Enfin, certaines études [26] mettent en avant les différentes intensités de couleur des objets, ce qui permet, par exemple, d’appliquer ultérieurement un algorithme de ligne de partage des eaux.

5.5.1.2 MISE EN PRATIQUE

La technique que nous avons mise en œuvre rejoint cette approche « intensité de couleur + ligne de partage des eaux ». Le choix de l’intensité des pixels repose sur le fait que les noyaux font généralement apparaître des pixels plus foncés.

En réalité, le point de départ de notre réflexion résidait dans l’utilisation d’un algorithme LPE pour segmenter les agrégats. La littérature abondante dans ce domaine préconise d’utiliser la LPE sur le gradient de l’image et non l’image originale. Nous avons testé l’algorithme LPE développé par Vincent et Soile [13]. L’image est sur-segmentée comme en atteste la figure suivante :

![Figure 5.5-2 : Sur-segmentation de la LPE classique.](image)

Le résultat précédent nous a poussés à utiliser l’algorithme LPE contraints par marqueurs vu la sur-segmentation importante engendrée par la LPE classique.

La technique des marqueurs consiste à définir préalablement le nombre de régions à mettre en évidence, c’est-à-dire le nombre de minima locaux et pertinents de l’image. La difficulté de cette approche est donc de déterminer ces marqueurs qui permettent de localiser les minima.

La stratégie développée sera illustrée via un exemple typique sur lequel on peut raisonnablement compter 10 noyaux isolés :
Chapitre 5 – Développement de l’outil de comptage automatique

Premièrement, l’image que nous traitons est « nettoyée », ce qui signifie que seuls les pixels inclus dans la composante sont pris en compte et non pas tous les pixels du rectangle englobant la composante :

Ensuite, nous appliquons un filtre maximum sur l’image nettoyée. Ce filtre utilise un élément structurant équivalent à celui de la figure 5.3-4, c’est-à-dire un disque d’un rayon de 5 pixels. L’opération consiste alors en une dilatation morphologique de l’image en niveaux de gris où chaque pixel est remplacé par la valeur maximale des pixels définis par l’élément structurant. Pour notre exemple, cela donne :

La dilatation a pour conséquence de réduire la surface de la composante, ce qui est logique puisque les pixels blancs qui entourent la composante ont une valeur supérieure aux pixels de cette dernière. Remarquons, ici, que la taille de l’élément structurant a été définie empiriquement de façon à convenir à toute taille d’agrégats.

53 Cf. section 5.3.3.1, recommandation 601.
54 Plus le rayon de l’élément structurant est grand et plus la composante sera « amincie ». Autrement dit, les agrégats de petite taille sont ignorés. À l’inverse, un rayon trop petit ne permet pas de mettre correctement en évidence les minima locaux.
Chapitre 5 – Développement de l’outil de comptage automatique

A ce stade, nous sommes en mesure de sélectionner les marqueurs pour la LPE. Pour ce faire, nous recherchons les minima locaux de l’image filtrée. L’algorithme utilisé pour la recherche des minima locaux a été développée dans ImageJ par Michael Schmid. Il consiste à parcourir la totalité de l’image où chaque pixel sera noté comme « minimum local » s’il respecte la condition suivante :

- Sa valeur est inférieure aux valeurs de ses pixels directs (en 8-connexité).

Un pixel sera noté comme « minimum local pur » si, en plus d’être un minimum local, il respecte le critère de tolérance au bruit.

**Définition** : Soient un seuil \( t_B \in \mathbb{N} \) appelé niveau de tolérance au bruit et \( P \) un minimum local. On dit que \( P \) respecte le critère de tolérance au bruit si aucun pixel de son enveloppe maximale, \( E_P (t_B) \), ne contient une valeur supérieure à la valeur du pixel \( P \).

**Définition** : Soient \( P \) un pixel quelconque de l’image et \( t_B \in \mathbb{N} \). L’enveloppe maximale \( E_P (t_B) \) du pixel \( P \) correspond à l’ensemble des pixels entourant \( P \) et dont les valeurs \( V_{env} \) sont telles que :

\[
V_{env} < P + t_B
\]

Pour notre application, les marqueurs utilisés pour la LPE correspondent en réalité aux « minima locaux purs » avec un seuil \( t_B \) fixé empiriquement \(^{55}\) à 10, ce qui donne, pour notre exemple, les marqueurs suivants :

---

\(^{55}\) Cette valeur de \( t_B \) (10) reflète bien le niveau de bruit que nous pouvons tolérer pour les images mises à notre disposition.
Chapitre 5 – Développement de l’outil de comptage automatique

Figure 5.5-7 : Marqueurs utilisés pour la LPE.

L’algorithme de ligne de partage des eaux utilisé dans notre application est simple. Il fait appel aux marqueurs préalablement trouvés et à la carte des distances euclidiennes, CDE, de l’image.

Les marqueurs sont les minima potentiels représentant les centres des composantes qui seront séparés par la segmentation. La CDE d’une image est calculée sur base de l’image binaire. Chaque pixel de l’avant plan (noir) est remplacé par une valeur en niveau de gris correspondant à sa distance euclidienne le séparant du plus proche pixel de l’arrière plan (blanc). Pour notre exemple, cela donne :

Figure 5.5-8 : A droite : Carte des Distances Euclidiennes + Marqueurs.

L’opération de la LPE consiste à dilater itérativement les régions autour de chaque marqueur jusqu’à ce que ces régions atteignent un pixel blanc, défini par la CDE, ou le bord d’une autre région. Pour résumer, nous obtenons le résultat suivant, avec 10 composantes trouvées :

Figure 5.5-9 : Chaîne de traitement du comptage des agrégats.

Remarque : Le nombre d’agréagts est calculé sur base de l’image binaire obtenue après l’application de la LPE. Les régions dont la surface est inférieure à 15 pixels sont ignorées car considérées comme non-significatives.
5.5.1.3 Évaluation de la méthode

Afin d'évaluer la méthode, nous proposons d'établir le taux d'erreur entre un comptage d'agrégats manuel et le comptage automatique proposé ci-dessus. De plus, nous comparerons également notre méthode avec un comptage par surface.

Le comptage par surface que nous avons implémenté est simple : afin de trouver le nombre de noyaux dans un agrégat, nous divisons la surface de cet agrégat par la valeur médiane de l'ensemble des surfaces détectées56.

Ceci implique inévitablement l'hypothèse que plus de la moitié des composantes détectées sont des noyaux isolés. Cette hypothèse est raisonnable car même lorsque la densité cellulaire est importante, la majorité des composantes connexes restent isolés. La proportion d'agrégat ne dépasse qu'exceptionnellement la barre des 50%.

Les tests effectués ont été réalisés sur base de neuf images représentatives de l'ensemble des images mises à notre disposition. Nous rapportons ici les résultats globaux pour chaque image et renvoyons aux l'annexe fournie sur le CD pour le détail complet du comptage.

<table>
<thead>
<tr>
<th>Images</th>
<th>Manuel</th>
<th>Notre Méthode</th>
<th>Taux d'erreur (%)</th>
<th>Surface</th>
<th>Taux d'erreur (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.jpg</td>
<td>21</td>
<td>19</td>
<td>9,52</td>
<td>21</td>
<td>0</td>
</tr>
<tr>
<td>108.jpg</td>
<td>534</td>
<td>523</td>
<td>2,06</td>
<td>625</td>
<td>17,04</td>
</tr>
<tr>
<td>B52.3_B2.2.2.jpg</td>
<td>82</td>
<td>87</td>
<td>6,10</td>
<td>191</td>
<td>132,93</td>
</tr>
<tr>
<td>BB94_10_5M.jpg</td>
<td>24</td>
<td>24</td>
<td>0</td>
<td>29</td>
<td>20,83</td>
</tr>
<tr>
<td>IK9-10-5M(21).jpg</td>
<td>87</td>
<td>93</td>
<td>6,9</td>
<td>129</td>
<td>48,28</td>
</tr>
<tr>
<td>si3_bFGF_B6.jpg</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>svl2-10-8M(5).jpg</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>10</td>
<td>42,86</td>
</tr>
<tr>
<td>Tv6.tif.jpg</td>
<td>53</td>
<td>55</td>
<td>3,77</td>
<td>62</td>
<td>16,98</td>
</tr>
<tr>
<td>VEGF_h_25_A(7).jpg</td>
<td>101</td>
<td>101</td>
<td>0</td>
<td>163</td>
<td>61,39</td>
</tr>
<tr>
<td>TOTAL</td>
<td>913</td>
<td>913</td>
<td>0</td>
<td>1234</td>
<td>35,16</td>
</tr>
<tr>
<td>MOYENNE</td>
<td>3,15</td>
<td></td>
<td>37,83</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Nous remarquons que notre méthode s'avère plus précise qu'un comptage par surface. De plus, dans l'absolu, elle offre des résultats très satisfaisants. Un comptage par surface surévalue presque toujours le nombre réel de noyaux dans l'agrégat, notamment à cause du cytoplasme.

En effet, lors de la segmentation initiale, il arrive fréquemment que les composantes détectées incluent du cytoplasme. Ce dernier augmente inévitablement la surface totale de la composante, et le comptage devient alors erroné. Pour preuve, cet exemple d'une composante de l'image « B52.3_B2.2.2.jpg »57 :

56 La valeur du nombre d'agrégats est arrondie selon la règle standard de l'arrondi, en utilisant l'entier supérieur ou inférieur le plus proche selon la valeur partie décimale.
57 Voir « DétailsComptageAgrégats.pdf » sur le CD fourni ; Annexe I, B.52.3_B2.2.2.jpg, crop 70.
En ce qui concerne la méthode que nous avons développée, nous avons constaté un risque de

- **Sous-évaluation** : lorsque les différents noyaux formant l’agrégat sont très proches et que la couleur est uniformément homogène. L’explication réside dans la difficulté à trouver les minima locaux purs pertinents alors que les intensités des noyaux de l’agrégat sont homogènes.

- **Sur-évaluation** : lorsque la composante, classée comme « Agrégat » comporte également un ou plusieurs artefacts (en l’occurrence des pores). Il s’agit évidemment d’un problème de classification automatique. Néanmoins, à ce niveau, notre méthode ne fait pas de distinction parmi les différentes particules constituant l’agrégat. Illustration :

Les phénomènes de sous-évaluation et sur-évaluation se compensent en partie mais, globalement, la méthode a une très légère tendance à surévaluer le nombre exact de noyaux constituant tous les agrégats de l’image.
5.6 PHASE D’ÉDITION - CORRECTIONS

Avant de considérer les résultats comme définitifs, nous avons jugé pertinent de proposer à l’utilisateur un outil d’édition/correction. Cet outil permet de modifier la classe à laquelle appartiennent les différentes composantes connexes. Une composante classée comme « Agrégat » peut donc être modifiée en une composante classée « Noyau isolé » ou « Artefact » et vice-versa.

De plus, pour la classe « Agrégat », il est possible d’éditer le nombre de noyaux trouvés dans un seul agrégat, ceci uniquement afin d’améliorer les résultats de comptage.

En guise d’illustration, voici l’interface graphique utilisée pour l’édition/correction. Un simple clic sur une composante permet de modifier sa classe. Nous invitons le lecteur à lire le manuel d’utilisation fourni avec le CD complémentaire afin de mieux visualiser les possibilités offertes par notre logiciel pour cette phase d’édition.

Figure 5.6-1 : Interface graphique de l’outil d’édition/correction
Bien entendu, au même moment, l’image originale est également visible afin de mener à bien les corrections.

Comme nous l’avons déjà mentionné, il ne s’agit pas d’abandonner l’idée de totalement automatiser le comptage, mais plutôt d’offrir une perspective de renforcement des modèles grâce à l’exportation des composantes corrigées dans l’ensemble d’apprentissage du classificateur.

Ne perdons pas non plus de vue que cette étape « manuelle » est relativement rapide, voire inutile dans le cas des images « favorables » pour lesquelles le comptage automatique, précédemment réalisé, est suffisant.
CHAPITRE 6

TESTS ET ÉVALUATIONS

6.1 INTRODUCTION

Nous avons déjà évalué les étapes de classification automatique et de segmentation des agrégats individuellement. Malheureusement, la première étape de segmentation initiale n’a pas été évaluée car, pour ce faire, il aurait fallu que des experts valident eux-mêmes la qualité des quatre méthodes de segmentation développées. Néanmoins, cette première étape influence les résultats finaux. Nous remarquerons donc, qu’en fonction de la segmentation appliquée, le dénombrement est plus ou moins précis.

Afin d’évaluer la méthode que nous avons développé, nous allons mesurer la capacité du logiciel à dénombrer correctement le nombre de noyaux présents sur une image. Pour ce faire, nous comparons le nombre de noyaux détectés automatiquement au nombre de noyaux comptés manuellement par les biologistes.

6.2 MESURES

Afin d’évaluer la qualité des résultats, nous avons réalisé différentes mesures dont :

1. **Le taux d’erreur moyen pondéré**
   Soit l’expérience $Y$ comptant $T$ images. Soient aussi $C_m$, le nombre de cellules comptées manuellement sur ces $T$ images et $C_a$ le nombre de cellules comptées automatiquement sur ces mêmes $T$ images.

   En général, il est habituel de calculer le taux d’erreur de l’expérience $Y$ de la manière suivante :

   $$E_c = \frac{|C_a - C_m|}{C_m}.$$

   Comme vous le constaterez à la section 6.3 (expérience NT24HA), le taux d’erreur classique ne donne pas toujours un aperçu correct de la qualité de l’outil de comptage. En effet, les sous-évaluations commises dans une image peuvent être compensées par les sur-évaluations commises dans une autre image de la même expérience.

   Pour cette raison, nous avons décidé de calculer un taux d’erreur moyen pondéré en fonction du nombre de cellules présentes dans chaque image.
Le poids de chaque image $\alpha_i$ est défini par :

$$\alpha_i = \frac{C_{m_i}}{\max_i (C_m)}$$

où $C_{m_i}$ est le nombre de cellules comptées manuellement dans l’image $i$ et où $\max_i (C_m)$ est le maximum des $C_{m_i}$ de l’expérience.

Le taux d’erreur moyen pondéré $E_p$ de l’expérience $Y$ se calcule donc comme suit :

$$E_p = \frac{\sum_{i=1}^{T} \left( \frac{C_{a_i} - C_{m_i}}{C_{m_i}} \times 100 \right) \times \alpha_i}{\sum_{i=1}^{T} \alpha_i}$$

2. L’erreur quadratique moyenne (MSE)
L’erreur quadratique moyenne est une mesure permettant d’évaluer la qualité d’un estimateur. Dans notre cas, les estimateurs sont l’ensemble des $C_{a_i}$, c’est-à-dire l’ensemble des valeurs données par le comptage automatique.

La MSE mesure donc la « proximité » de chaque $C_{a_i}$ avec leurs $C_{m_i}$ respectifs. Cette proximité est maximale lorsque la valeur estimée est égale à la valeur exacte. En particulier, la MSE permet de pénaliser plus fortement les cas où la différence entre valeur estimée et valeur exacte est importante.

La formule utilisée pour définir la MSE d’une expérience de $T$ images est :

$$MSE = \frac{\sum_{i=1}^{T} (C_{a_i} - C_{m_i})^2}{T}$$

Dans notre cas, on considérera que plus la valeur de la MSE est petite et plus le dénombrement est précis.

3. L’erreur moyenne absolue (MAE)
L’idée est la même que la MSE, si ce n’est qu’au lieu de calculer le carré de la différence entre la valeur exacte et la valeur estimée, on calcule la valeur absolue de la différence. La formule est donc la suivante :

$$MAE = \frac{\sum_{i=1}^{T} |C_{a_i} - C_{m_i}|}{T}$$

Ici aussi, une faible valeur pour la MAE signifiera qu’il y a peu de différence entre la valeur estimée et la valeur exacte. Le dénombrement est donc plus précis.
4. **Coefficient de corrélation de Pearson**

Un outil efficace pour mesurer l’intensité de la liaison qui existe entre deux variables est le coefficient de corrélation de Pearson. Ce coefficient est égal au rapport de leur covariance et du produit non nul de leurs écarts types. Sa valeur appartient à $\mathbb{R}$ et varie dans l’intervalle $[-1,1]$.

Soit

$$ PEARSON = \frac{\sum_{i=1}^{T} (C_{mi} - C_{m}) \cdot (C_{ai} - C_{a})}{\sqrt{\sum_{i=1}^{T} (C_{mi} - C_{m})^2 \cdot \sum_{i=1}^{T} (C_{ai} - C_{a})^2}} $$

où $T$ est le nombre d’images, $C_{mi}$ (resp. $C_{ai}$) le nombre de cellules comptées manuellement (resp. automatiquement) dans l’image $i$ et $C_{m}$ (resp. $C_{a}$) la moyenne du nombre de cellules trouvées manuellement (resp. automatiquement) dans l’ensemble des $T$ images.

Plus ce coefficient est proche de 1 (ou -1) et plus la corrélation entre les variables est forte. Une corrélation égale à 0 signifie que les variables sont linéairement indépendantes. En revanche, un coefficient de 1 (resp. -1) signifie que l’une des variables est fonction affine$^{58}$ croissante (resp. décroissante) de l’autre. Dans ce cas, les variables sont linéairement dépendantes.

Graphiquement, le coefficient de Pearson permet de visualiser la dispersion du nuage de points autour de la droite $y = x$. Sur le schéma suivant, nous avons en abscisse, les valeurs estimées et en ordonnée les valeurs réelles issues du comptage manuel.

Figure 6.2-1 : Coefficient de Pearson (Expérience IK9-8M, cf. section 6.3).

---

$^{58}$ Une fonction affine est une fonction dont la représentation graphique est une droite. $f : x \rightarrow ax + b$
Nous allons maintenant présenter et interpréter les mesures que nous avons obtenues lors de la phase de tests effectuée sur 78 images provenant de 5 expériences différentes et pour lesquelles nous avions les résultats de comptage manuel réalisé par les chercheurs du GIGA\textsuperscript{59}.

Chaque expérience est illustrée par une image type et une image résultat. L’image résultat est composée de 3 couleurs :

- Le vert pour les composantes de la classe « Noyaux isolés »
- Le rose pour les composantes de la classe « Agrégats »
- Le bleu pour les composantes de la classe « Artefact »

\textsuperscript{59} Le lecteur trouvera, sur le CD fourni en complément, le fichier « Tests-Experiences.xls » où nous avons analysé 205 images provenant de 15 expériences différentes.
Chapitre 6 – Tests et Evaluations

1. EXPÉRIENCE IK9-8M

Image type de l’expérience

Figure 6.3-1 : Expérience IK9-8M.

Résultats

<table>
<thead>
<tr>
<th>Expérience IK9-8M</th>
<th>Mesures</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 images</td>
<td>Manuel</td>
</tr>
<tr>
<td>Dénombrément total</td>
<td>3466</td>
</tr>
<tr>
<td>Taux d’erreur % (classique)</td>
<td>2,07</td>
</tr>
<tr>
<td>Taux d’erreur % (pondéré)</td>
<td>2,36</td>
</tr>
<tr>
<td>MSE</td>
<td>41,55</td>
</tr>
<tr>
<td>MAE</td>
<td>4,55</td>
</tr>
<tr>
<td>Pearson</td>
<td>0,999559</td>
</tr>
</tbody>
</table>

Interprétation

Les images de cette expérience présentent peu d’artefacts et peu d’agrégats. Le contraste de couleur est bien prononcé. Il s’agit d’un cas favorable. Les différentes mesures prouvent que l’erreur est minime surtout pour la segmentation NRGB (ce qui sera le cas pour toutes les expériences.
2. EXPÉRIENCE NT24HA

Image type de l’expérience

Figure 6.3-2 : Expérience NT24HA.

Résultats

<table>
<thead>
<tr>
<th>Expérience NT24HA</th>
<th>Mesures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>6 images</strong></td>
<td>Manuel</td>
</tr>
<tr>
<td>Dénombrement total</td>
<td>1082</td>
</tr>
<tr>
<td>Taux d’erreur % (classique)</td>
<td>4,99</td>
</tr>
<tr>
<td>Taux d’erreur % (pondéré)</td>
<td>19,22</td>
</tr>
<tr>
<td>MSE</td>
<td>1292,33</td>
</tr>
<tr>
<td>MAE</td>
<td>34,66</td>
</tr>
<tr>
<td>Pearson</td>
<td>0,741503</td>
</tr>
</tbody>
</table>

Interprétation

Ici, le taux d’erreur classique est trompeur. En effet, on pourrait croire que les résultats sont bons mais ce n’est pas vraiment le cas comme en témoignent les 4 mesures que nous avons définies. On remarque ici un autre avantage de se fier au taux d’erreur pondéré : en cas de mauvais comptage, il reflète mieux la réalité que le taux d’erreur classique. De plus, les mesures de MSE et MAE sont plus élevées que pour les autres expériences et les coefficients de corrélation de Pearson traduisent bien une corrélation moyenne entre les valeurs estimées et les valeurs exactes.

Ce cas est défavorable car les images sont sombres et certaines cellules se confondent avec le cytoplasme. La phase de segmentation initiale ne prend donc pas en compte ces cellules pour lesquelles le contraste de couleur est trop peu marqué !
Chapitre 6 – Tests et Evaluations

3. EXPÉRIENCE SVL2-8M

Image type de l’expérience

![Image 1]

![Image 2]

Figure 6.3-3 : Expérience svl2-8M.

Résultats

<table>
<thead>
<tr>
<th>Expérience svl2-8M</th>
<th>Mesures</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 images</td>
<td>Manuel</td>
</tr>
<tr>
<td><strong>Dénombrement total</strong></td>
<td>4496</td>
</tr>
<tr>
<td><strong>Taux d’erreur % (classique)</strong></td>
<td>3,04</td>
</tr>
<tr>
<td><strong>Taux d’erreur % (pondéré)</strong></td>
<td>3,04</td>
</tr>
<tr>
<td><strong>MSE</strong></td>
<td>37,14</td>
</tr>
<tr>
<td><strong>MAE</strong></td>
<td>5,07</td>
</tr>
<tr>
<td><strong>Pearson</strong></td>
<td>0,999850</td>
</tr>
</tbody>
</table>

Interprétation

Ce type d’image est tout à fait approprié à l’outil que nous avons développé. Quelque soit le type de segmentation utilisé, le taux d’erreur pondéré ne dépasse pas les 6%, avec encore une très bonne approche de la méthode Nrgb.
4. EXPÉRIENCE IK9-6M

Image type de l’expérience

Figure 6.3-4 : Expérience IK9-6M.

Résultats

<table>
<thead>
<tr>
<th>Expérience IK9-6M</th>
<th>Mesures</th>
</tr>
</thead>
<tbody>
<tr>
<td>21 images</td>
<td>Manuel</td>
</tr>
<tr>
<td>Dénombrément total</td>
<td>3181</td>
</tr>
<tr>
<td>Taux d’erreur % (classique)</td>
<td>15,54</td>
</tr>
<tr>
<td>Taux d’erreur % (pondéré)</td>
<td>5,43</td>
</tr>
<tr>
<td>MSE</td>
<td>112,42</td>
</tr>
<tr>
<td>MAE</td>
<td>8,23</td>
</tr>
<tr>
<td>Pearson</td>
<td>0,998361</td>
</tr>
</tbody>
</table>

Interprétation

Ici, le taux d’erreur pondéré ainsi que les valeurs de la MSE et MAE prouvent que le comptage de l’image n’est pas un cas favorable. L’erreur commise sur ce type d’image est très difficile à corriger. En effet, nous remarquons globalement une surévaluation du nombre exact de cellules. Cela s’explique par le fait que les images contiennent énormément de taches de coloration (artefacts) que le classificateur assimile à des noyaux.

Nous sommes nous-mêmes parfois étonnés des résultats du comptage manuel tant la confusion entre taches et noyaux est possible. Afin de corriger ce problème, nous avons introduit ces taches dans la classe « Artefacts » de l’ensemble d’apprentissage. Malheureusement nous avons vite abandonné cette idée car des tests effectués sur d’autres types d’images donnaient une classification incorrecte où trop de noyaux étaient considérés comme artefacts !
Chapitre 6 – Tests et Evaluations

5. EXPÉRIENCE « AGRÉGATS »

L’expérience « Agrégats » consiste à réunir une série d’images pour lesquelles la densité cellulaire est importante. Dans les tableaux de résultats, nous ajoutons la colonne « Méthode de base » qui fait référence à l'algorithme naïf développé à la section 2.2.1. Ainsi nous pourrons constater les améliorations apportées par notre méthode.

Image type de l’expérience

![Image de l’expérience](image.png)

Figure 6.3-5 : Expérience "Agrégats".

Résultats

<table>
<thead>
<tr>
<th>Expérience « Agrégats »</th>
<th>Mesures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>6 images</strong></td>
<td>Manuel</td>
</tr>
<tr>
<td>Dénombrement total</td>
<td>2964</td>
</tr>
<tr>
<td>Taux d'erreur % (classique)</td>
<td>20,14</td>
</tr>
<tr>
<td>Taux d'erreur % (pondéré)</td>
<td>20,14</td>
</tr>
<tr>
<td>MSE</td>
<td>17553,16</td>
</tr>
<tr>
<td>MAE</td>
<td>99,5</td>
</tr>
<tr>
<td>Pearson</td>
<td>0,640629</td>
</tr>
</tbody>
</table>
Interprétation

Nous remarquons logiquement une sous-évaluation forte du nombre de cellules dans le cas de la méthode de base. C’est évidemment son principal défaut puisque les composantes ne sont pas classées et que dès lors les agrégats sont comptés comme « 1 » lors d’un dénombrement.

Il est aussi intéressant de comparer la méthode de base avec la segmentation RGB. En effet, sur base de la même image binaire, nous constatons toute l’importance des étapes de classification et segmentation des agrégats.

Enfin, nous remarquons encore une fois que la segmentation Nrgb est préférable. Afin d’illustrer la raison pour laquelle cette segmentation donne les meilleurs résultats, nous consacrons l’expérience suivante à illustrer les images binaires issues des quatre méthodes de segmentation développées. Le lecteur pourra se rendre compte de la grande qualité obtenue via la segmentation Nrgb (ce qui est logique vu les conclusions tirées à la section 4.3.3).
Chapitre 6 – Tests et Evaluations

6. EXPÉRIENCE « COLORATION »

Voici une série d’images miniatures présentant des colorations variées mais toujours dans les tons mauve/rose.

Figure 6.3-6 : Exemple de différentes colorations des images.

En zoomant sur certaines parties de ces images, nous allons mettre en évidence les différentes forces et faiblesses des quatre méthodes de segmentation.

Exemple 1

Constatations

La segmentation RGB ignore certaines composantes que nous avons entourées en orange. Cela est du à la surface de ces composantes qui est inférieure à 100 pixels. Les composantes sont donc ignorées.

La segmentation Nrgb, elle, est quasi parfaite tandis que les segmentations GRAY et PCA incluent une partie de cytoplasme dans l’image binaire. Il en résulte une seule composante connexe au lieu de deux comme l’indiquent les flèches orange. De plus, la GRAY inclut aussi un pore.
Exemple 2

Constatations

On constate de nouveau que la segmentation RGB ignore quelques composantes entourées en orange. Cette fois-ci, cela est du à la circularité de cette composante qui est inférieure à 0.10. D'une manière générale les méthodes RGB, GRAY et PCA incluent une nouvelle fois une bonne partie du cytoplasme (cet effet est renforcé avec des images sombres comme celle-ci). L’avantage de la segmentation Nrgb est de pouvoir s’affranchir des variations d'intensité lumineuse.

Ce nouvel exemple illustre encore une fois toute l’importance d’une segmentation initiale de qualité.

Exemple 3

Constatations

Nous remarquons qu’hormis la segmentation Nrgb, les trois autres prennent une partie des portes en considération. De plus, certaines composantes sont assimilées à des agrégats alors que la segmentation Nrgb définit correctement les frontières de chaque noyau.
Chapitre 6 – Tests et Evaluations

Exemple 4 : Cas rares (PCA)

Sur plus de 200 images analysées, nous avons aperçu trois cas où la segmentation PCA était complètement erronée. Cela arrive lorsque l'image présente très peu de cellules et de nombreux pores. En effet la couleur « mauve » n’étant pas suffisamment présente sur l'image, la PCA segmente l’image en incluant les pores. Grossièrement, on peut dire que la couleur des pores l’emporte sur la couleur des noyaux.

![Figure 6.3-7 : Erreur PCA : Image originale.](image1)

![Figure 6.3-8 : Erreur PCA : Image binaire.](image2)

Conclusions

Si nous devions établir un classement selon la qualité de segmentation, la première place revient évidemment à la segmentation Nrgb qui peut être jugée comme « très bonne ».

Viennent ensuite les segmentations GRAY et PCA qui se valent. Chacune de deux a tendance à inclure certains pores et/ou cytoplasme. On vient de le voir, la segmentation PCA peut aussi avoir des effets inattendus.

La segmentation la moins efficace est la segmentation RGB. Cela s’explique par le fait que trop de pixels sont « autorisés », c’est-à-dire considérés comme composante connexe. Les régions peuvent alors être très grandes ou très petites, ce qui engendre des composantes ignorées à cause de leur petite taille, ou des composantes pas assez circulaires.
Chapitre 6 – Tests et Evaluations

6.4 TEMPS D’EXÉCUTION ET COMPLEXITÉ

L’étape de segmentation initiale utilise un algorithme par seuillage. Cet algorithme a une complexité de l’ordre de $O(n)$, où $n$ est le nombre de pixels de l’image.

L’algorithme de segmentation des agrégats a une complexité équivalente à la complexité de recherche des minima locaux purs, soit $O(n_{a} \times N_{Env})$ où $n_{a}$ est le nombre de pixels de l’image-agrégat et où $N_{Env}$ correspond au nombre de pixels formant l’enveloppe maximale d’un minimum local.

Le temps d’exécution de ces 2 étapes est négligeable par rapport au temps d’exécution de l’étape de classification. En effet, cette dernière met entre 20 secondes et 2 minutes par image, dépendant du nombre de composantes connexes à classer. On pourrait dès lors mettre en cause la complexité de l’algorithme de classification qui est

$$O(N_{Tests} \times T \times \log N_{LS}),$$

où $N_{sw}$ est le nombre de fenêtres extraites dans l’image de départ (pour rappel, nous avons fixé sa valeur à 100), $T$ est le nombre d’arbres du modèle (10) et $\log N_{LS}$ est la profondeur moyenne de chaque arbre, avec $N_{LS}$ étant le nombre de fenêtre de l’ensemble d’apprentissage. La complexité de l’étape de classification est donc tout à fait acceptable !

En réalité, si le temps de classification est parfois long, c’est parce qu’actuellement, chaque composante est d’abord sauvegardée sur le disque, puis classée et enfin supprimée. Cette opération est répétée autant de fois qu’il y a de composantes détectées. C’est donc cette succession de sauvegardes/suppressions sur disque qui est coûteuse en temps.

Ne perdons donc pas de vue que le logiciel développé est un prototype et qu’une des perspectives du travail sera de fournir directement les composantes au classificateur sans passer préalablement par une sauvegarde sur disque.
CHAPITRE 7

CONCLUSION

7.1 RÉSUMÉ

Ce projet de fin d'études s'intéresse à la possibilité d'automatiser le plus possible le processus de comptage cellulaire pour des images acquises au microscope et issues d'expériences en chambre de Boyden.

Les principaux axes de recherche sont la recherche d'une composante de couleur adéquate pour la segmentation initiale, la classification des composantes connexes par une méthode d'apprentissage supervisé et enfin la mise au point d'une technique efficace de séparation des agrégats.

Le résultat final consiste en un logiciel et plus exactement un « plug-in » ImageJ que nous mettrons prochainement à la disposition des biologistes afin d'atteindre entièrement le but initial que nous nous étions fixés, à savoir proposer une solution concrète à cheval entre les besoins des chercheurs et nos connaissances en sciences informatiques.

Une part importante de ce travail a consisté en une recherche de méthodes déjà existantes dans le domaine. La littérature nous a dirigés vers la thèse très intéressante d'E. Glory, source indéniable d'idées pertinentes et applicables, dans une certaine mesure, à notre cas (notamment le choix de la segmentation Nrgb).

Nous avons ensuite implémenté puis testé quatre techniques de segmentation d'images. En particulier, la technique PCA n'a malheureusement pas été à la hauteur de nos attentes. En effet, nous cherchions une technique de segmentation aussi précise que générique.

Pour l'étape de classification, nous avons utilisé le classificateur PixIT, fruit de la thèse de doctorat de Raphaël Marée. Enfin, une technique de segmentation des agrégats basée sur l'utilisation d’un algorithme de partage des eaux contraint par des marqueurs a été développée.

D'un point de vue général, nous jugeons les résultats très encourageants. Les différentes mesures effectuées (taux d’erreur moyen pondéré, MSE, MAE, Person) le prouvent. Notre méthode est certes imparfaite mais donne un niveau de précision très satisfaisant. De plus, nous le verrons dans les perspectives d'amélioration, ces résultats ne tendent qu’à s’améliorer.

Personnellement, ce projet a été quelques fois éprouvant de par sa durée dans le temps et la rigueur que je me suis imposée. Jamais, auparavant, je ne m’étais autant impliqué dans un travail. J’espère avoir mis ma persévérance au service des biologistes pour qui la corvée du comptage manuel est réellement problématique. Puise mon travail les aider dans cette voie, cela constituerait pour moi une grande réussite et une grande fierté.
7.2 PERSPECTIVES / AMÉLIORATIONS

Nous l’avons vu, une des principales difficultés rencontrées est la caractérisation des agrégats, et plus globalement, la détermination d’un ensemble d’apprentissage robuste à plusieurs types d’images. En ce sens, une des perspectives du travail serait de définir un ensemble d’apprentissage par expérience ou par types d’images similaires. Cela permettrait également d’annihiler un autre problème de classification : les taches de coloration qui sont parfois considérés comme des noyaux isolés.

Pour ce faire, nous invitons les biologistes à standardiser le processus d’acquisition des images. Une piste à suivre est définie dans la thèse d’E. Glory à partie de la section 3.2. En résumé, il faudrait :

– Définir un protocole bien précis de préparation des échantillons (fixation, coloration,...)
– Utiliser un matériel d’acquisition performant et un environnement d’acquisition (luminosité, focus, zoom) répétables pour plusieurs expériences.
– Automatiser l’acquisition des images grâce à un logiciel tel que Lucía (Laboratory Universal Computer Image Analysis) qui contrôle l’autofocus ou encore la vitesse d’acquisition.

Une autre perspective de travail, ô combien intéressante, serait d’implémenter le critère $C$ dont nous avons parlé à la section 4.3.1.3. Nous serions alors en mesure de choisir automatiquement le seuil de segmentation optimal pour chaque image, mais également de choisir l’espace de couleur adéquat ou encore mieux de définir notre propre espace de couleur, celui minimisant au plus ledit critère.

Une perspective que nous offrons aux biologistes sans pouvoir encore mesurer son efficacité est la phase d’édition/correction. Les composantes connexes exportées lors de cette étape permettront, on l’espère, de renforcer les nouveaux modèles. Les taux d’erreur du classificateur devraient diminuer au fil des expériences.

Enfin, d’un point de vue plus technique, le temps de classification des composantes connexes diminuera significativement dès lors qu’on envoie directement au classificateur les composantes connexes sans passer par des phases d’écriture/suppression sur disque. Pour l’instant, le classificateur impose cette contrainte.
Bibliographie


[26] Zhang et al. (2004). "Extraction of karyocytes and their components from microscopic bone marrow images based on regional color features", Pattern Recognition
