

UNIVERSITÉ DE GENÈVE

Département de géologie et paléontologie

Département d'informatique

FACULTÉ DES SCIENCES

Professeur Eric Davaud

Professeur Christian Pellegrini

---

# **JMicroVision: un logiciel d'analyse d'images pétrographiques polyvalent**

## **THÈSE**

présentée à la Faculté des sciences de l'Université de Genève  
pour obtenir le grade de Docteur ès sciences, mention interdisciplinaire

par

**Nicolas RODUIT**

de

Fully (VS)

Thèse N° 3830

GENÈVE

Atelier de reproduction de la Section de physique

2007

La Faculté des sciences, sur le préavis de Messieurs E. DAVAUD, professeur ordinaire et directeur de thèse (Département de géologie et paléontologie), C. PELLEGRINI, professeur ordinaire et codirecteur de thèse (Département d'informatique), N. KESKES, docteur (Total Exploration & Production, Centre Scientifique Total Jean Ferrer, Pau, France) et B. CALINE, docteur (Total Exploration & Production, Centre Scientifique Total Jean Ferrer, Pau, France) autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 6 février 2007

**Thèse - 3830 -**

*P.Spierer*  
**Le Doyen, Pierre SPIERER**

---

Roduit, N.: JMicrоЩision: un logiciel d'analyse d'images pétrographiques polyvalent.  
Terre & Environnement, vol. 65, ix + 116 pp. (2007)

ISBN 2-940153-64-7

Section des Sciences de la Terre, Université de Genève, 13 rue des Maraîchers, CH-1205 Genève, Suisse  
Téléphone ++41-22-702.61.11 - Fax ++41-22-320.57.32  
<http://www.unige.ch/sciences/terre/>

# **Remerciements**

J'adresse mes sincères remerciements à ERIC DAVAUD, initiateur de ce projet, pour m'avoir fait confiance dans ce travail hors norme, pour sa disponibilité et pour ses précieux conseils. Je tiens également à remercier chaleureusement CHRISTIAN PELLEGRINI pour avoir accepté l'encadrement de la partie informatique.

Je remercie vivement la compagnie Total pour sa contribution au financement de ce projet, en particulier BRUNO CALINE, qui m'a accueilli à plusieurs reprises au centre scientifique de Pau et qui a activement contribué à l'évolution de JMicrоЩision en le confrontant à une réalité industrielle. J'adresse aussi tous mes remerciements à NAAMEN KESKES pour avoir accepté de faire partie du jury et pour avoir partagé sa grande expérience dans le développement logiciel.

Merci à toutes les personnes qui ont contribué de près ou de loin à l'amélioration de JMicrоЩision, notamment à JULIEN FIORE pour m'avoir proposé des idées d'amélioration à de nombreuses reprises, à CLAUDE-ALAIN HASLER, PIERRE LE GUERN, YANNICK FUCHEY, STEPHAN JORRY et BENJAMIN SALLIER pour leurs remarques constructives et pour avoir fourni des exemples d'application, ainsi qu'à MAPATHE NDIAYE pour les discussions passionnées concernant le développement en Java.

Enfin, je tiens à exprimer ma gratitude à mes proches, qui m'ont soutenu tout au long de ce travail, en particulier à VALENTINE pour sa patience et sa compréhension.

# Résumé

Dans le domaine de la pétrographie, l'observation microscopique reste incontournable pour déterminer les roches et comprendre leur genèse. On pourrait s'attendre à ce que les composants pétrographiques puissent être aisément quantifiés et caractérisés de façon informatique. Toutefois, en raison de la complexité et de la grande variété des images, les logiciels du marché ne se montrent pas toujours très performants. Cette thèse tente de pallier cette lacune en fournissant une palette d'outils informatiques plus adaptés à la visualisation et à l'analyse d'images pétrographiques. Ces outils ont été développés en langage Java et regroupés dans le logiciel JMicrоЩision.

JMicrоЩision a été conçu de manière à pouvoir afficher et traiter des images haute résolution dans des conditions optimales. A l'instar d'un microscope, il permet l'observation dynamique d'une lame mince avec la possibilité de combiner différents focus ou modes d'éclairage (lumière polarisée, cathodoluminescence...). De plus, la loupe et l'outil multivue permettent une vision simultanée de plusieurs images, chacune pouvant avoir un coefficient de zoom propre tout en conservant une position commune au centre. Même avec des images de très grandes tailles et un ordinateur doté d'une faible capacité mémoire, les outils de visualisation se révèlent performants grâce à la technique du tuilage et à l'exécution différée.

Cette thèse propose un nouvel algorithme de séparation des grains connexes avec une plus grande capacité à déconnecter les particules allongées ou celles contenant des trous que les algorithmes traditionnels. Afin d'optimiser la gestion de la mémoire lors de certaines opérations, nous avons adapté l'algorithme de croissance de régions pour qu'il soit compatible avec le tuilage et avec l'exécution différée. D'autres outils innovants, notamment l'analyse granulométrique ou le comptage de points informatisé, ont été développés. Ce dernier représente une alternative lorsque les différentes méthodes de segmentation ne permettent pas une estimation satisfaisante de la composition modale d'une roche. Il offre plusieurs avantages par rapport au comptage effectué sous le microscope, comme la limitation à un type de région (après segmentation) afin d'en estimer rapidement les sous-classes.

Dans JMicroVision, les caractéristiques des objets extraits de l'image peuvent être mises en évidence de manière visuelle soit par divers diagrammes soit par une coloration des objets en fonction de leur valeur (selon le descripteur sélectionné). Le rendu visuel d'un projet ou d'une portion de celui-ci peut être exporté sous forme d'images dans plusieurs formats ou sous forme de graphiques vectoriels dans le format SVG. De même, toutes les données sont exportables dans le format CSV.

JMicroVision contient des outils ayant divers degrés d'automatisation afin de pouvoir traiter le caractère complexe et varié des images. L'intuition de l'opérateur reste néanmoins primordiale pour envisager la suite des opérations la plus adaptée et pour le réglage des paramètres dans l'interface utilisateur. Plusieurs marches à suivre pour des analyses pétrographiques sont proposées dans la deuxième partie de cette thèse. Par exemple, afin de caractériser la porosité, une classification supervisée des types de pore a été réalisée à l'aide d'un nouveau paramètre nommé "contexte", décrivant l'environnement proche des pores.

# Abstract

In the field of petrography, microscopic observation remains fundamental for determining rocks and understanding their genesis. One could expect that petrographic components can easily be quantified and characterized by a computer program. However, because of the complexity and the large variety of images, existing software is not always very efficient. This thesis tries to fill this gap by providing a suite of computer tools more adapted to the visualization and the analysis of petrographic images. These tools were developed in the Java programming language and were gathered in the JMicrоЩision software.

JMicrоЩision was designed to display and process high-resolution images under optimal conditions. Like a microscope, it enables a dynamic observation of a thin section with the possibility of combining various focus or modes of lighting (polarized light, cathodoluminescence...). Moreover, the magnifying lens and the multiview tool allow a simultaneous view of several images, each one having its own zoom coefficient while keeping a joint position in the center. Even with very large images and a computer with little memory, the visualization tools remain powerful thanks to tiling and deferred execution.

This thesis presents a new method to separate connected grains, which is more efficient than traditional methods on long particles and on those containing holes. In order to optimize memory management during certain operations, we adapted the algorithm of region growing so that it is compatible with tiling and deferred execution. Other innovating tools, such as granulometric analysis and computerized point counting, were developed. The latter represents an alternative when the various segmentation methods do not allow a satisfying estimate of the modal composition. It offers several advantages compared to the counting under the microscope, like the restriction to a region type (after segmentation) in order to quickly estimate its subclasses.

In JMicrоЩision, characteristics of the objects extracted from the image can be highlighted in a visual way either by various diagrams or by a coloring of objects according to their value (in relation with a selected descriptor). The rendering of the project or a portion of it can be

exported in several image formats or as vector graphics in the SVG format. All data is also exportable in the CSV format.

JMicroVision contains tools having various degrees of automation in order to handle with complex and varied images. The operator's intuition is nevertheless crucial for deciding which is the right way to proceed and for adjusting the parameters in the user's interface. Several procedures for petrographic analyses are proposed in the second part of this thesis. For example, in order to characterize porosity, a supervised classification of the pore types was carried out with a new parameter named “context”, which describes the neighborhood to the pores.

# Table des matières

<b>I. Traitement et analyse d'image: outils et méthodes</b>	<b>1</b>
<b>Introduction.....</b>	<b>1</b>
<b>Chapitre 1 : JMicrоЩision .....</b>	<b>5</b>
1.1 Présentation .....	5
1.1.1 L'interface.....	7
1.1.2 La programmation.....	7
1.2 Architecture et caractéristiques techniques .....	9
1.2.1 Le langage Java.....	9
1.2.2 Lanceur multiplateforme.....	10
1.2.3 Les bibliothèques JAI et Java 2D.....	12
1.2.4 Technique du tuilage.....	14
<b>Chapitre 2 : Acquisition et prétraitement.....</b>	<b>17</b>
2.1 Matériel de numérisation.....	17
2.1.1 Introduction .....	17
2.1.2 Scanner à plat.....	19
2.1.3 Scanner à diapositives .....	19
2.1.4 Microscope équipé d'une caméra .....	20
2.1.5 Microscope électronique à balayage (MEB) .....	21
2.2 Prétraitements .....	22
2.2.1 Introduction .....	22
2.2.2 Opérations de filtrage .....	23
2.2.3 Soustraction d'un fond non uniforme .....	25
2.2.4 Rectification géométrique.....	25
<b>Chapitre 3 : Segmentation.....</b>	<b>27</b>
3.1 Introduction.....	27
3.2 Définitions préliminaires .....	28
3.2.1 Types d'image .....	28
3.2.2 Définition d'une région .....	31
3.2.3 Géométrie discrète et connexité des pixels .....	32
3.2.4 Algorithme de croissance de régions .....	33
3.3 Segmentation par seuillage.....	34
3.4 k-means.....	35
3.5 Ligne de partage des eaux (LPE) .....	37
3.5.1 Segmentation à l'aide de marqueurs .....	39
3.5.2 Segmentation d'images binaires par l'érodé ultime .....	41
<b>Chapitre 4 : Caractérisation d'objets 2D .....</b>	<b>43</b>
4.1 Introduction.....	43
4.2 Objets, trous et fond .....	44
4.3 Représentation vectorielle des objets .....	45
4.3.1 Introduction .....	45
4.3.2 Algorithme d'extraction de contour.....	46
4.3.3 Cas particulier .....	47
4.4 Algorithme de croissance de régions adapté au tuilage .....	49
4.4.1 Extraction des objets.....	50
4.4.2 Opérations binaires .....	51

4.5 Séparation d'objets connexes .....	53
4.5.1 Méthodes de séparation.....	53
4.5.2 Algorithme de séparation par recherche des concavités (SRC).....	54
4.5.3 Discussion .....	60
4.6 Modificateurs de forme.....	61
4.6.1 Introduction .....	61
4.6.2 Lissage par filtrage .....	62
4.6.3 Approximation polygonale.....	63
4.6.4 Enveloppe convexe.....	64
4.7 Descripteurs .....	64
4.7.1 Position.....	65
4.7.2 Descripteurs de base .....	65
4.7.3 Moments.....	67
4.7.4 Ratios .....	69
4.7.5 Descripteurs additionnels.....	71
4.7.6 Discussion .....	74
4.8 Classification .....	76
 <b>II. Exemples d'applications en pétrographie</b>	 <b>79</b>
<b>Introduction.....</b>	<b>79</b>
<b>Chapitre 5 : Mesures et quantification .....</b>	<b>83</b>
5.1 Introduction.....	83
5.2 Comptage de point informatisé .....	84
5.3 Estimation de la composition modale par segmentation .....	87
5.4 Caractérisation d'un faciès par un critère d'orientation .....	88
<b>Chapitre 6 : Granulométrie .....</b>	<b>91</b>
6.1 Introduction.....	91
6.2 Analyse individuelle des grains.....	92
6.3 Profils granulométriques .....	95
<b>Chapitre 7 : Porosité .....</b>	<b>99</b>
7.1 Introduction.....	99
7.2 Estimation de la porosité et de la perméabilité .....	99
7.2.1 La porosité .....	99
7.2.2 La perméabilité.....	100
7.3 Classification des types de porosité.....	102
7.3.1 Introduction .....	102
7.3.2 Proportion des types de pore par analyse individuelle .....	103
7.3.3 Proportion des types de pore par comptage de points .....	104
7.4 Caractérisation du réseau poreux.....	105
7.4.1 Introduction .....	105
7.4.2 Distribution de la taille des pores .....	105
7.4.3 Homogénéité du réseau poreux.....	107
<b>Chapitre 8 : Conclusion .....</b>	<b>109</b>
8.1 Apports de cette thèse .....	109
8.2 Perspectives.....	111
<b>Bibliographie.....</b>	<b>113</b>

# Liste des figures

Figure 1.1 – Interface de JMicroVision.....	6
Figure 1.2 – Architecture du JRE 1.5 .....	10
Figure 1.3 – Fonctionnement du lanceur.....	11
Figure 1.4 – Conception multiplateforme du lanceur .....	12
Figure 1.5 – Tuilage d'une image .....	15
Figure 2.1 – Résolution du système d'acquisition .....	18
Figure 2.2 – Observation d'une lame mince en lumière transmise .....	18
Figure 2.3 – Perception d'une lame mince en lumière transmise .....	19
Figure 2.4 – Scanner à diapositives avec support adapté à l'acquisition en lumière polarisée .....	20
Figure 2.5 – Microscope équipé d'une caméra reliée à une station de travail.....	21
Figure 2.6 – Réduction de l'information chromatique par la compression JPEG.....	23
Figure 2.7 – Soustraction du fond non uniforme .....	25
Figure 2.8 – Correction de la déformation à partir de points de contrôle .....	26
Figure 3.1 – Espaces colorimétriques RVB et TLS .....	31
Figure 3.2 – Relation de voisinage des pixels.....	32
Figure 3.3 – Segmentation par seuillage de l'histogramme .....	34
Figure 3.4 – Segmentation par l'algorithme des k-means.....	36
Figure 3.5 – Principe de la ligne de partage des eaux .....	37
Figure 3.6 – Segmentation par LPE .....	38
Figure 3.7 – LPE construite à partir des marqueurs .....	39
Figure 3.8 – Segmentation par LPE à l'aide de marqueurs .....	39
Figure 3.9 – Segmentation d'une image binaire par l'érodé ultime .....	42
Figure 4.1 – Définition des objets, des trous et du fond.....	44
Figure 4.2 – Directions des huit codes de Freeman .....	45
Figure 4.3 – Digitalisation et extraction de la bordure .....	45
Figure 4.4 – Pincements et filaments d'un objet .....	46
Figure 4.5 – Cas particulier d'extraction de contour .....	47
Figure 4.6 – Tuilage d'une image binaire .....	49
Figure 4.7 – Fusion d'un objet à cheval sur deux tuiles .....	49
Figure 4.8 – Inscription des identificateurs dans l'ossature .....	52
Figure 4.9 – Méthodes classiques de séparation des objets connexes.....	54
Figure 4.10 – Nouvelles méthodes de séparation des objets connexes.....	54
Figure 4.11 – Pixels de liaison .....	55
Figure 4.12 – Masques de détection des zones de liaison .....	55
Figure 4.13 – Différenciation des trous et du fond .....	56
Figure 4.14 – Approximation polygonale des contours .....	57
Figure 4.15 – Recherche des points de coupe dans les concavités du contour polygonal .....	57
Figure 4.16 – Processus de séparation.....	59
Figure 4.17 – Lissage des objets .....	62
Figure 4.18 – Approximation polygonale: algorithme de la corde .....	63
Figure 4.19 – Paramètres de position .....	65
Figure 4.20 – Descripteurs de base .....	67
Figure 4.21 – Mesure d'un diamètre de Feret à 130 degrés .....	71
Figure 4.22 – Principe de la mesure contexte .....	72
Figure 4.23 – Variation de l'aire et du périmètre selon l'orientation .....	74

Figure 4.24 – Variation du périmètre selon l'orientation et la technique utilisée .....	75
Figure 4.25 – Descripteurs de forme.....	75
Figure 4.26 – Classification par les plus proches voisins.....	77
Figure 5.1 – Illusions d'optique.....	83
Figure 5.2 – Chartes visuelles.....	84
Figure 5.3 – Comptage de points sur un sable éolien.....	85
Figure 5.4 – Composition modale d'une roche volcanique .....	87
Figure 5.5 – Caractérisation de deux types dépôts selon l'orientation des grains .....	88
Figure 6.1 – Mesure de la surface des grains dans un faciès éolien .....	93
Figure 6.2 – Histogrammes de la surface des grains.....	94
Figure 6.3 – Processus d'individualisation des grains par LPE .....	95
Figure 6.4 – Profil granulométrique sur un sable .....	96
Figure 7.1 – Compacité de différents types de pore .....	101
Figure 7.2 – Proportion des types de pore par analyse individuelle .....	103
Figure 7.3 – Proportion des types de pore par comptage de points .....	104
Figure 7.4 – Exemples de réseaux microporeux dans des roches carbonatées .....	105
Figure 7.5 – Carte des distances du réseau microporeux .....	106
Figure 7.6 – Caractérisation du réseau microporeux .....	106
Figure 7.7 – Homogénéité du réseau poreux.....	107



# **I. Traitement et analyse d'image: outils et méthodes**

## **Introduction**

En géologie, comme dans d'autres disciplines scientifiques, l'imagerie numérique occupe une place toujours plus grande. Elle comprend notamment les images satellites, les photos aériennes, l'imagerie sismique, les vues de terrains et les images d'échantillons de roche numérisés à différentes échelles par des dispositifs variés. Dans le cadre de ce travail, nous nous sommes plus particulièrement intéressés à ce dernier type d'image, bien que JMicrоЩision, le logiciel issu de cette thèse, soit en mesure de traiter toute sorte d'images.

La quantification et la caractérisation du contenu des lames minces sont deux aspects fondamentaux en pétrographie. On pourrait s'attendre à ce que l'informatique facilite ces

opérations, voire permettre de les automatiser, mais la réalité est plus contrastée. Si le pétrographe s'aide effectivement de logiciels pour l'observation et l'archivage des images, la quantification et la caractérisation des éléments de façon informatique restent des tâches difficiles en raison de la diversité et de la complexité des images ainsi que de l'inadéquation des outils logiciels. A défaut de procédures automatisées, la quantité des composants pétrographiques est donc estimée à l'aide de chartes visuelles de façon approximative ou par le comptage de points sous le microscope, qui représente un travail fastidieux. De surcroît, les résultats obtenus par ces deux méthodes peuvent varier d'un opérateur à l'autre selon son appréciation. L'analyse d'image, qui permet une approche quantitative, se profile donc comme une alternative aux méthodes existantes<sup>1</sup>.

Grâce aux performances des ordinateurs actuels et à l'évolution du matériel d'acquisition, l'utilisation d'images haute résolution tend à se généraliser. Cependant, les images de très haute définition nécessitent une architecture logicielle adaptée, qui se retrouve très peu dans les logiciels du marché. L'objectif de ce travail est donc de développer un outil informatique plus adapté aux besoins du pétrographe et capable de traiter des images de grande taille. Par ailleurs, ce logiciel doit permettre à l'utilisateur d'avoir autant que possible un contrôle visuel sur les opérations successives afin qu'il n'ait pas l'impression d'utiliser des outils "boîtes noires". En outre, à l'instar d'un logiciel SIG (Système d'Information Géographique), il doit pouvoir afficher de manière hiérarchisée les couches vectorielles comme les mesures, les annotations ou les objets. Enfin, un effort particulier a été consenti pour rendre l'interface graphique ergonomique et pour donner de la souplesse aux outils tout en conservant leur simplicité et leur efficacité.

Le développement d'un tel programme informatique est un défi ambitieux, car les images pétrographiques se révèlent particulièrement complexes par comparaison avec les images représentant des objets industriels, sur lesquelles les méthodes d'analyse d'image conventionnelles ont déjà démontré leur efficacité. Des outils plus spécifiques à la

---

<sup>1</sup> Parmi les autres approches quantitatives, on trouve notamment les méthodes minéralogiques et géochimiques (comme la microsonde, la fluorescence X, la diffraction des rayons X, etc.), qui requièrent toutefois un matériel sophistiqué.

pétrographie sont donc nécessaires. JMicrоЩision propose des outils innovants comme la séparation des particules connexes, l'analyse granulométrique ou le comptage de points informatisé.

La première partie de cette thèse, qui compte quatre chapitres, présente les outils et les méthodes de traitement et d'analyse d'image du logiciel JMicrоЩision. Mis à part le premier chapitre, qui présente les caractéristiques du logiciel, tous les autres évoquent chronologiquement les cinq grandes étapes de l'analyse d'image<sup>2</sup> (voir Figure 1). Parmi les méthodes élaborées pour JMicrоЩision, seules les méthodes novatrices ou celles présentant un intérêt particulier y sont présentées en détail. La deuxième partie exposera des exemples concrets d'application du logiciel dans le domaine de la pétrographie. En ce qui concerne la prise en main du logiciel, l'information se trouve en annexe dans le guide utilisateur.

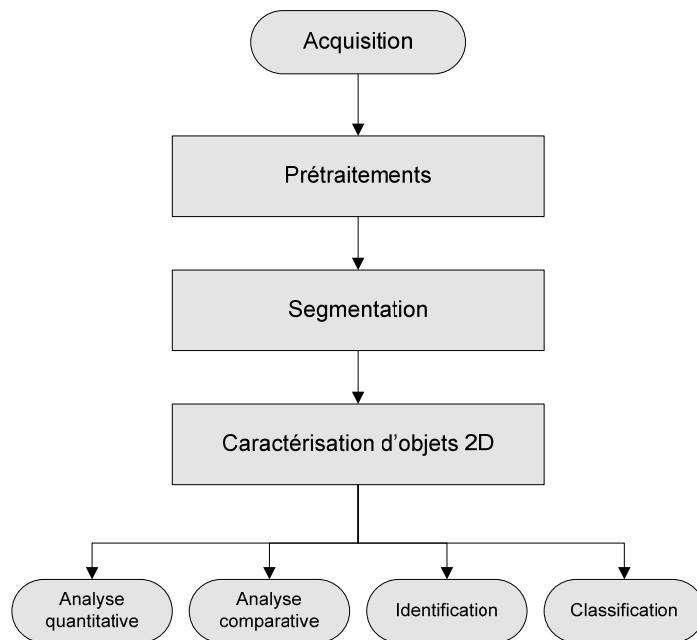


Figure 1 – Déroulement de l'analyse d'image

L'acquisition ne fait pas proprement partie de l'analyse d'image, mais elle fournit les images qui alimentent le processus. Les prétraitements visent à améliorer la qualité des images brutes

<sup>2</sup> Le terme "analyse d'image" est parfois utilisé pour désigner la collecte d'informations sur l'image sans que celle-ci soit segmentée en régions. JMicrоЩision offre des outils permettant d'obtenir diverses mesures statistiques directement sur une portion ou sur l'ensemble de l'image.

lorsqu'elles sont bruitées ou qu'elles contiennent des défauts. Quant à la segmentation, elle est l'étape critique et sans doute la plus difficile. Elle consiste à subdiviser l'image en régions (ou objets) présentant un intérêt du point de vue de l'utilisateur. Dans l'étape suivante, ces objets sont mesurés par différentes techniques. Chaque objet est alors caractérisé par de nombreux descripteurs servant par la suite à la classification, à l'identification ou à la comparaison des objets. Plusieurs outils de JMicrоЩision permettent de mettre en évidence de manière visuelle les caractéristiques des objets en les colorant ou par des diagrammes. Enfin, JMicrоЩision offre la possibilité d'exporter toutes les mesures effectuées sur les objets pour que ces données puissent ensuite être analysées par un logiciel de statistiques.

# **Chapitre 1 : JMicr0Vision**

Au vu de l'évolution incessante du matériel et des technologies informatiques, le choix du langage de développement est déterminant pour la durée de vie d'un logiciel. Plusieurs aspects pratiques dépendront par ailleurs de ce choix, comme l'élaboration, la maintenance, l'évolution et la portabilité du logiciel. Ce chapitre présente le logiciel JMicr0Vision, ses spécificités, son caractère multiplateforme et les librairies qui constituent ses fondations en termes d'architecture logicielle. Le guide utilisateur (User Guide, incorporé au logiciel) fournit plus d'informations concernant JMicr0Vision et son utilisation.

## **1.1 Présentation**

JMicr0Vision est un programme informatique regroupant différents outils et techniques servant à analyser les images pétrographiques. Plusieurs versions stables sont disponibles sur Internet<sup>3</sup> et peuvent être installées sur les systèmes Windows, Linux, Solaris et Mac OS X. Une distribution cent pour cent Java peut également être installée sur d'autres plateformes supportant l'environnement d'exécution Java. Toutefois, les opérations de traitement d'image ne bénéficieront pas d'une accélération native sur ces systèmes (voir Section 1.2.3).

JMicr0Vision intègre les principales fonctions de traitement d'image, un module de rectification, un module de calibration, différentes méthodes de segmentation, un module de comptage de points, des outils de mesure selon une ou deux dimensions, un outil pour placer des annotations sur l'image et une aide contextuelle. Il permet de plus le développement de

---

<sup>3</sup> Sur le site officiel à l'adresse <http://www.jmicr0vision.com> ou sur différents sites de téléchargement comme Download.com, Softpedia, Tucows, MacUpdate ou VersionTracker.

nouvelles opérations de traitement ou d'analyse d'image, qui peuvent être ajoutées dans l'interface par le biais d'une structure dédiée aux plugins.

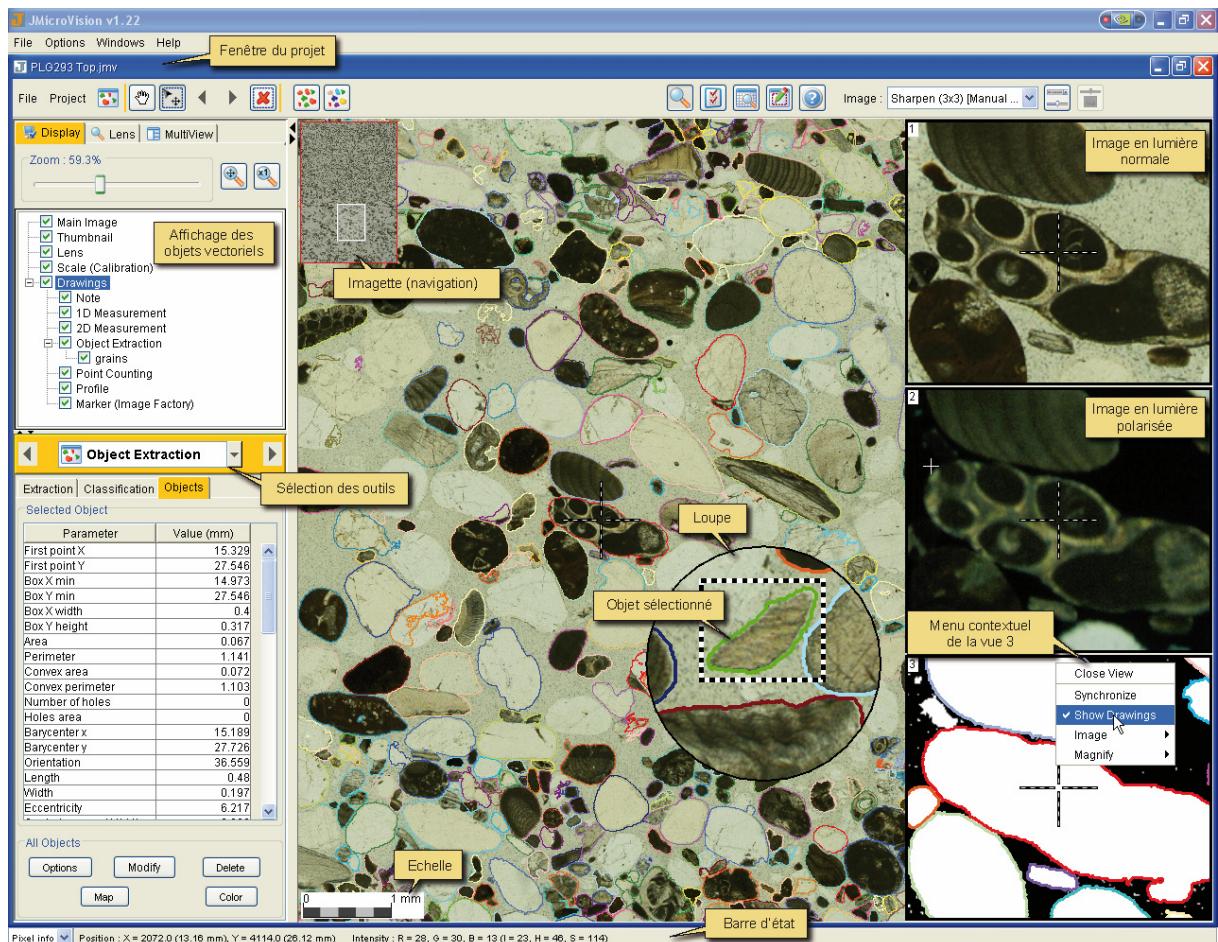


Figure 1.1 – Interface de JMicrоЩision

JMicrоЩision lit les formats d'image TIFF, BMP, FlashPiX, GIF, JPEG, PNG et PNM, et peut exporter un projet (image + les couches vectorielles) ou une image dans un format TIFF ou PNG. Un projet peut aussi être exporté dans le format vectoriel SVG.<sup>4</sup> JMV est le format de fichier spécifique à JMicrоЩision qui regroupe le contenu d'un projet, c'est-à-dire les couches vectorielles (les annotations, les mesures selon une ou deux dimensions, les objets résultant de l'extraction automatique, les profils et le comptage de points), la calibration spatiale, les zones de travail, toutes les opérations effectuées sur les images, ainsi que la configuration des outils

<sup>4</sup> Scalable Vector Graphics (SVG) est un format de fichier permettant de décrire des ensembles de graphiques vectoriels. SVG est une spécification du World Wide Web Consortium (W3C).

et de certains éléments de l'interface. Afin de limiter la taille du fichier projet, seuls les chemins des images liées à un fichier et l'information nécessaire pour créer les images résultant d'opérations sont inclus.

### 1.1.1 L'interface

La fenêtre d'un projet (voir Figure 1.1) contient une ou plusieurs images issues d'une scène, toutes les opérations effectuées sur ces images, les différentes couches vectorielles et tous les outils de visualisation et d'analyse. Par cette configuration d'interface, qui regroupe à la fois les données et les outils autour d'une suite d'images représentant la même scène, JMicrоЩision se distingue nettement des autres logiciels d'analyse d'image. Lorsque plusieurs projets sont ouverts en même temps, la plupart des opérations peuvent être exécutées en parallèle et arrêtées à tout moment en cliquant sur leur barre de progression.

Le système de visualisation, qui comprend la partie droite de la fenêtre principale, la loupe et l'outil multivue (multiview), permet d'afficher les images de façon optimale grâce à la technique du tuilage et à l'exécution différée (voir Section 1.2.4). Avec l'outil multivue, plusieurs images selon une position donnée peuvent être affichées en même temps avec un zoom propre à chacune. Cet outil s'avère particulièrement utile pour comparer plusieurs opérations sur une image ou pour visualiser simultanément différentes images provenant d'une même scène.

L'outil "Data Viewer" permet d'afficher sous forme de tableaux de données les caractéristiques des différents objets vectoriels. En cliquant sur une donnée du tableau, l'objet est sélectionné sur l'image et inversement. L'outil "Area Editor" donne la possibilité de définir des zones de forme rectangulaire, elliptique ou polygonale, qui serviront par la suite à limiter le traitement d'image ou l'analyse de données à une certaine surface.

### 1.1.2 La programmation

JMicrоЩision a été réalisé à l'aide de l'environnement de développement intégré Borland JBuilder (avec les versions 9, 10, 2005 et 2006), qui offre de nombreux outils pour faciliter le développement et accroître le rendement. Le recours à la librairie commerciale Jsuite, contenant une suite de composants graphiques, a permis d'éviter autant que possible la

programmation de diagrammes. Cette dernière se révèle en effet difficile, surtout lorsque les diagrammes doivent répondre à une certaine souplesse d'affichage. Dans quelques cas, des diagrammes ont tout de même dû être développés car les objets préconstruits ne correspondaient pas aux attentes.

Le tableau ci-dessous donne un aperçu de certaines propriétés et de la longueur du code source de la version 1.2.2 de JMicr0Vision.

Nombre de lignes	91'229
Nombre de lignes de code	71'884
Nombre de commentaires (ceux-ci peuvent figurer sur plusieurs lignes)	3'244
Nombre de caractères	3'265'007
Nombre de points-virgules	42'042
Nombre de packages	29
Nombre de classes	557
Abstraction (rapport entre classes abstraites + interfaces et classes)	7.3 %
Nombre de propriétés	4'206
Nombre de méthodes	4'806
Nombre d'opérandes	193'263
Nombre d'opérateurs	89'577
Nombre d'opérandes uniques	11'398
Nombre d'opérateurs uniques	53
Moyenne du nombre de constructeurs par classe	1.56
Moyenne du nombre de propriétés par classe	6.34
Moyenne du nombre de méthodes par classe	8.62
Moyenne du nombre de paramètres par méthode	0.82
Moyenne du nombre de lignes de code par méthode	11.35
Moyenne des imbrications de blocs (structures délimitées par des accolades) par méthode	1.06
Moyenne de la complexité cyclomatique par méthode (donne le nombre de chemins distincts des instructions de branchemennt conditionnel)	1.97

Tableau 1.1 – Statistiques du code source

Le code a été protégé contre la décompilation au moyen du logiciel libre Proguard. En plus de brouiller le code, ce logiciel l'optimise en supprimant les parties non utilisées et en remplaçant tous les identificateurs par des lettres de l'alphabet. Grâce à cette optimisation, la taille de l'ensemble des classes a été réduite de plus de moitié.

## 1.2 Architecture et caractéristiques techniques

Comme nous l'avons dit, le langage de programmation est un facteur déterminant pour l'élaboration d'un logiciel. Pour notre part, nous avons opté pour le langage Java en raison de son caractère orienté objet et multitâche, de sa portabilité sur les différentes plateformes et du nombre important de librairies qu'il offre. JMicr0Vision a été développé avec les spécifications de Java 1.5 et principalement à l'aide des librairies Java 2D et Java Advanced Imaging (JAI).

### 1.2.1 Le langage Java

Java est un langage de programmation orienté objet, développé par Sun Microsystems et présenté officiellement en 1995, dont l'évolution a été rapide en raison de son grand succès. Il permet une gestion appropriée des erreurs et possède un ramasse-miettes, chargé de recycler la mémoire en libérant les objets obsolètes. Le risque d'erreur est réduit car Java n'utilise pas de pointeurs et oblige à initialiser les variables.

Java est aussi un langage semi-interprété, lié à un environnement d'exécution nommé Java Runtime Environnement (JRE). Le code source est compilé en un langage intermédiaire appelé byte-code Java, qui est ensuite interprété par n'importe quelle plateforme possédant un JRE. L'utilisation d'un environnement d'exécution intermédiaire entre l'application et la plateforme constitue un avantage considérable, car il rend le développement logiciel indépendant des caractéristiques du système d'exploitation et du type de matériel. Ainsi, la même application peut être exécutée sur une machine équipée d'un processeur 32 bits avec le système Windows ou sur une machine munie d'un processeur 64 bits avec le système Linux. Les différentes architectures matérielles et le type de système d'exploitation sont pris en charge par une version du JRE adaptée à ces caractéristiques. Seul bémol, il est parfois difficile de lancer une application Java pour différentes raisons exposées à la Section 1.2.2. Pour aider l'utilisateur lors du démarrage de l'application, un lanceur a donc été développé, dont la caractéristique est d'être multiplateforme, c'est-à-dire capable de s'exécuter sur la plupart des machines (architectures des processeurs Intel x86, Powerpc, Sparc etc.) et des systèmes d'exploitation (Windows, Linux, Mac OS, Unix etc.).

En ce qui concerne la performance, l'interprétation du bytecode par le JRE a pour effet de ralentir l'exécution d'un programme Java. Cet inconvénient est en grande partie compensé par la technologie "Just in time" (Deutsch & Schiffman, 1984), intégrée dans la machine virtuelle Java, qui permet d'accélérer considérablement l'exécution des programmes en compilant à la volée les parties de byte-code les plus importantes en un code natif. Actuellement, cette technologie permet d'exécuter des programmes avec des performances comparables à celles obtenues par des applications développées en langage C ou C++.

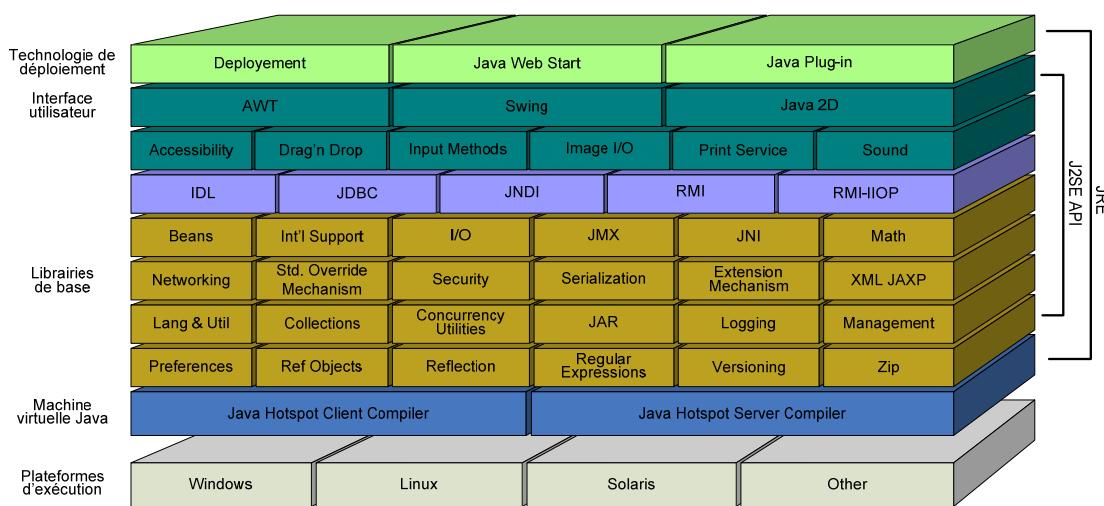


Figure 1.2 – Architecture du JRE 1.5

### 1.2.2 Lanceur multiplateforme

En raison d'intérêts divergents entre les géants de l'informatique, lancer une application Java sur certains systèmes d'exploitation n'est pas toujours une tâche facile. Ces difficultés apparaissent lorsque le JRE est inexistant, mal configuré ou que le système en possède plusieurs. C'est pour ces raisons que certaines distributions de JMicroVision regroupent à la fois l'application et le JRE. Cette association garantit que le programme soit exécuté par une version adaptée du JRE. En contrepartie, la taille du logiciel augmente fortement. Dans le cas des distributions sans JRE, un lanceur aide à détecter et à configurer l'emplacement du JRE au travers d'une interface graphique.

Le lanceur est incorporé dans toutes les distributions, excepté celle cent pour cent Java et celle pour Mac OS X, et ses fonctions principales sont: transmettre des arguments et démarrer l'application. Les arguments transmis sont lus à partir du fichier JMicroVision.ini et

définissent des paramètres comme la mémoire requise par l'application ou l'utilisation de pilotes graphiques OpenGL.

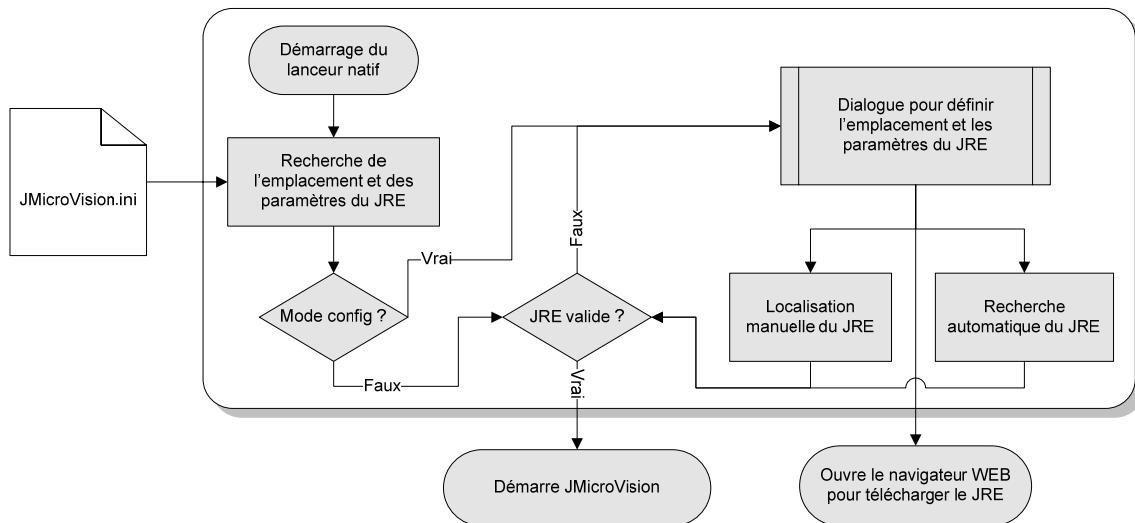


Figure 1.3 – Fonctionnement du lanceur

Le lanceur doit s'exécuter directement sur le système d'exploitation puisque son premier objectif est de découvrir l'emplacement du JRE. Par conséquent, il doit être compilé dans un code natif, propre au système et au type de matériel. Pour rester dans une conception de programmation indépendante de la plateforme, le lanceur a été conçu en C++ (librairies standards) avec des spécifications pour Windows, Linux et Solaris, et l'interface graphique a été construite à l'aide de la librairie multiplateforme wxWidgets<sup>5</sup>. Cette approche permet d'écrire un seul code source, qui doit cependant être compilé pour chaque système d'exploitation et architecture de processeur (voir Figure 1.4).

En pratique, le lanceur ne fonctionne pas sur tous les systèmes d'exploitation, bien qu'il emploie exclusivement les bibliothèques standards C++ et que les parties requises de la librairie wxWidgets y soient incorporées. Des tests effectués sur plusieurs systèmes d'exploitation ont mis en évidence que le lanceur fonctionne correctement sous les différents Windows (excepté Windows 3.1). En revanche, avec les nombreuses distributions et versions de Linux, le lanceur ne peut parfois pas démarrer sans une mise à jour de certaines librairies

<sup>5</sup> Bibliothèque multiplateforme de composants graphiques, voir le site officiel "<http://www.wxwidgets.org>".

du système. Cette difficulté apparaît aussi avec le système Solaris. Une procédure d'identification des librairies manquantes ou obsolètes est proposée dans la documentation d'installation de JMicr0Vision afin de faciliter la mise à jour du système d'exploitation. En cas de problème persistant, il est toujours possible de lancer JMicr0Vision en ligne de commande en définissant correctement l'emplacement du JRE et ses paramètres. En ce qui concerne le système Mac OS X, les problèmes liés au JRE ne se posent pas, car celui-ci est déjà intégré au système et il existe une méthode pour lancer et configurer une application Java par le biais d'un fichier XML.

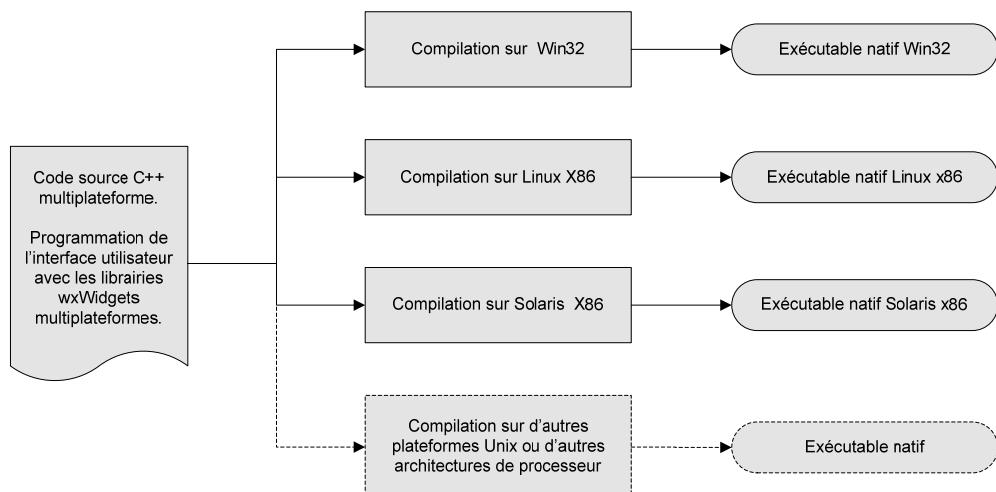


Figure 1.4 – Conception multiplateforme du lanceur

### 1.2.3 Les librairies JAI et Java 2D

JAI (Java Advanced Imaging) est une librairie additionnelle dédiée aux traitements d'image avancés et liée à la librairie Java 2D (voir Figure 1.2), avec laquelle elle est compatible. Java 2D est un ensemble de classes destinées à l'affichage en deux dimensions; son API (Application Programmer's Interface) offre, en autres, une série de fonctions permettant de manipuler les dessins vectoriels. Le code source de la librairie JAI a récemment été ouvert sous une licence non-commerciale JRL<sup>6</sup> et sous une licence commerciale JDL<sup>7</sup>. L'élaboration du logiciel JMicr0Vision a permis de révéler plusieurs bogues dans JAI et grâce à l'ouverture du code source, nous avons pu contribuer à leur résolution.

<sup>6</sup> JRL (Java Research License) peut être consultée à l'adresse <http://jai.dev.java.net/jrl.html>

<sup>7</sup> JDL (Java Distribution License) peut être consultée à l'adresse <http://jai.dev.java.net/jdl-jai.pdf>

L'API de JAI offre une suite de fonctions de traitement d'image sophistiquées et donne la possibilité de créer ses propres fonctions. Par ailleurs, JAI se distingue des librairies du même type par le fait qu'elle intègre des techniques comme le tuilage et l'exécution différée (voir Section 1.2.4). Pour réduire le temps de calcul, la plupart des opérateurs de JAI sont dupliqués dans un code natif spécifique aux systèmes Windows, Linux ou Solaris. De plus, des accélérations matérielles pour l'architecture des processeurs UltraSparc et pour la technologie MMX, développée par Intel, sont prises en charge. D'autres accélérations natives devraient être ajoutées à JAI dans le futur.

L'API de JAI contient des classes abstraites à partir desquelles de nouvelles opérations peuvent être développées. Voici les principales:

- PointOpImage: regroupe toutes les opérations de point à point, comme les opérations arithmétiques et logiques, qui sont caractérisées par le fait que la valeur du pixel de l'image résultante est calculée à partir du même pixel d'une ou de plusieurs images sources.
- AreaOpImage: permet de réaliser des opérations de filtrage ou de morphologie mathématique, où la valeur du pixel de destination est déterminée à partir d'une région rectangulaire (correspondant à la taille du filtre ou de l'élément structurant) autour du pixel de l'image source. Un mécanisme regroupant l'information nécessaire permet de traiter les extrémités des tuiles de l'image correctement.
- StatisticsOpImage: permet d'effectuer des calculs statistiques sur une région de l'image en définissant un taux d'échantillonnage.
- UntiledOpImage: permet d'accomplir des opérations où la valeur du pixel de destination est calculée à partir de pixels distribués dans une portion de l'image source, voire dans l'image entière. L'inconvénient de cette classe est qu'elle n'utilise pas la technique du tuilage et, par conséquent, qu'elle doit garder entièrement en mémoire l'image source et l'image résultante.
- WarpOpImage: regroupe toutes les opérations de transformation géométrique.
- OpImage: est une superclasse abstraite qui regroupe toutes les opérations effectuées sur une image. Elle permet de définir la structure de donnée de l'image résultante et son modèle de couleur.

#### 1.2.4 Technique du tuilage

La combinaison du tuilage et de l'exécution différée optimise considérablement le temps de calcul des opérations et la gestion de la mémoire. Lorsqu'une image de l'ordre du giga-octet doit être visualisée, traitée ou analysée, il devient primordial de se servir aussi efficacement que possible des ressources mémoire (RAM) pour ne pas devoir employer la mémoire swap<sup>8</sup>, ce qui risquerait de paralyser l'application. La technique du tuilage n'est en général pas intégrée dans les logiciels d'analyse d'image, car elle ne s'applique pas à toutes les opérations et parce que l'utilisation d'images très haute résolution et les moyens de stockage de grande capacité sont tout de même assez récents.

La technique du tuilage consiste à découper l'image en carreaux de taille identique. Dans JMicrоЩision, ce sous-ensemble de l'image est composé de tuiles d'une dimension de 512 par 512 pixels, soit la taille décrite par la documentation JAI comme étant la plus efficace par expérience. Le tuilage existe à différents niveaux dans l'architecture du logiciel. Tout d'abord, il peut se trouver dans la structure même du fichier de l'image (mais uniquement dans certains formats<sup>9</sup> comme Tiled TIFF, Flashpix et JPEG 2000) et dans les images résultant d'opérations, qui ne sont pas directement liées à un fichier. Le tuilage existe aussi au niveau de l'affichage des images et des différentes couches vectorielles superposées. Ainsi, pour la visualisation d'une image dont le fichier est tuilé et de grande taille, seules les tuiles requises par la zone d'affichage seront chargées en mémoire, et non toute l'image. La plupart des algorithmes de traitement et d'analyse d'image implémentés dans JMicrоЩision utilisent la structure en tuiles. Toutefois, dans certains cas où l'information nécessaire pour déterminer la valeur du pixel de destination est diffuse dans toute l'image source, le résultat sera exact pour chaque tuile considérée indépendamment, mais erroné pour l'image entière. Différentes astuces ont été mises en œuvre pour essayer d'adapter des opérations a priori non compatibles avec la technique du tuilage. C'est le cas pour la segmentation par les k-means (voir Section 3.4) et pour la croissance de régions (voir Section 4.4) telle qu'effectuée dans l'extraction des

---

<sup>8</sup> La mémoire swap ou virtuelle est une extension de la mémoire vive (RAM). Elle réside sur le disque dur, où le temps d'accès est en moyenne 5000 fois plus lent que celui de la RAM.

<sup>9</sup> Toutefois certains formats non compressés, comme TIFF ou BMP, devraient à l'avenir pouvoir être lus directement en mode tuile par les codecs de JAI.

objets et dans les opérations binaires de reconstruction, de remplissage des trous et de suppression des objets touchant les bords. Pour la ligne de partage des eaux (voir Section 3.5), les tentatives d'adaptation n'ont pas abouti car il est difficile de concilier l'indépendance de chaque tuile et le fait que l'algorithme doive progresser simultanément à partir des minima répartis dans toute l'image.

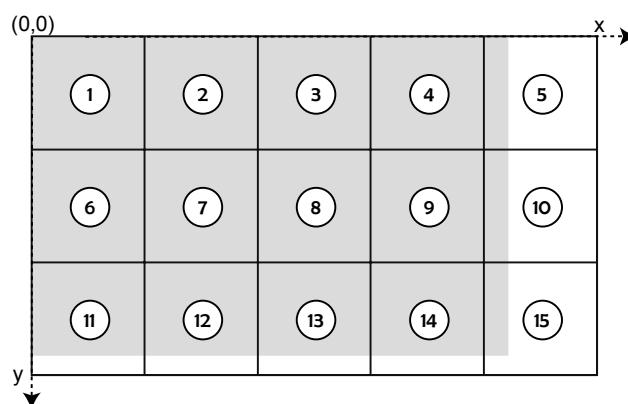


Figure 1.5 – Tuilage d'une image

L'exécution différée consiste à déclencher une opération ou une suite d'opérations uniquement au moment où l'image résultante devient indispensable, par exemple quand elle doit être affichée ou quand des statistiques sur l'image sont requises. Par cette technique, il est donc possible de conserver seulement le schéma des opérations successives avec leurs paramètres et de fabriquer les images uniquement à la demande. Lorsqu'une opération est lancée, chaque construction de tuile peut être incorporée dans une tâche<sup>10</sup> et traitée simultanément<sup>11</sup>. Les morceaux de l'image sont donc traités en parallèle par le processeur, qui n'a pas besoin d'attendre le chargement en mémoire de l'image entière pour commencer ses traitements. De plus, JAI offre un mécanisme de mémoire cache et de recyclage des tuiles afin d'assurer un équilibre optimal entre la construction des tuiles et l'utilisation de la mémoire. Les tuiles sont conservées autant que possible dans le cache pour être accessibles rapidement lorsqu'elles

<sup>10</sup> Processus indépendant géré par le système d'exploitation.

<sup>11</sup> La simultanéité est réelle pour un ordinateur multiprocesseur et apparente pour un monoprocesseur. Cependant, même la simultanéité apparente optimise le temps de calcul, car le processeur passe moins de temps à attendre le chargement des données à traiter.

sont demandées régulièrement. Dans JMicr0Vision, la capacité du cache à stocker les tuiles peut être ajustée dans les options.

# **Chapitre 2 : Acquisition et prétraitement**

L'acquisition et les prétraitements sont deux étapes qui précèdent la segmentation dans le schéma classique d'analyse d'image et qui peuvent avoir une influence cruciale sur les résultats des étapes suivantes. Ce chapitre présente les principales méthodes de numérisation de lames minces et d'échantillons de roche. Il y est question aussi de prétraitements qui ont pour objectif de corriger les défauts des images brutes. Cependant, tous les prétraitements visant à rendre l'image plus agréable à l'œil humain, comme les filtrages de renforcement, l'étalement des niveaux, la correction gamma et les réglages du contraste et de la luminosité, ne sont pas abordés ici car ils ont tendance à entraîner des pertes d'information et ne sont généralement daucune aide dans les processus d'analyse d'image.

## **2.1 Matériel de numérisation**

### **2.1.1 Introduction**

La numérisation d'échantillons s'effectue au moyen de systèmes d'acquisition très divers, dont le coût représente l'investissement principal dans le processus global d'analyse d'image. La première partie de ce chapitre n'a pas la prétention de couvrir l'ensemble des moyens d'acquisition, mais donne plutôt un aperçu des dispositifs les plus fréquemment utilisés en pétrographie, en relevant leurs avantages et leurs inconvénients du point de vue de l'utilisateur.

La qualité d'une image est déterminée par les propriétés optiques du matériel de numérisation et surtout par sa résolution, c'est-à-dire le degré de détail avec lequel l'appareil capture une zone donnée. La résolution se compte généralement en points par pouce (dpi - dots per inch) et correspond au nombre de points échantillonnés horizontalement ou verticalement par le dispositif d'acquisition sur 2,54 cm, soit 1 pouce. Ainsi, pour une même surface digitalisée,

plus la résolution d'acquisition est élevée, plus la dimension de l'image numérisée sera grande (voir Figure 2.1).

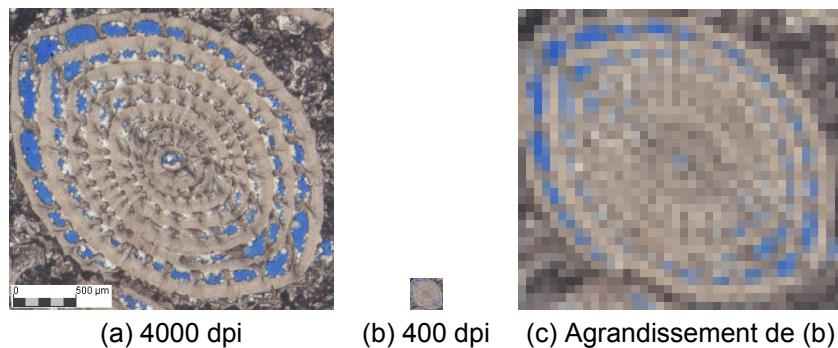


Figure 2.1 – Résolution du système d'acquisition

L'observation microscopique des lames minces est la base de la pétrographie. La plupart du temps, les lames minces sont observées en lumière transmise (voir Figure 2.2). La préparation des lamelles de roche est importante, car si ces dernières ont une épaisseur non constante, une partie de l'image capturée sera toujours floue, et si elles présentent des rayures, l'image contiendra des artefacts.

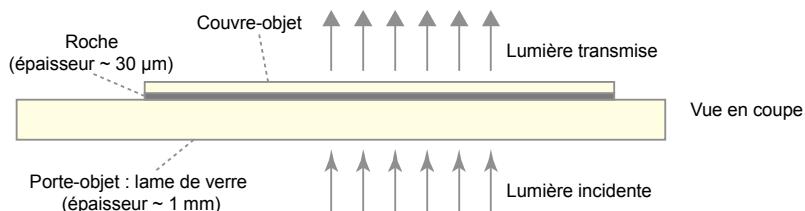


Figure 2.2 – Observation d'une lame mince en lumière transmise

Tous les dispositifs d'acquisition utilisant la lumière transmise fournissent des images qui ne représentent pas dans l'absolu de vraies coupes à deux dimensions. En réalité, la valeur d'un pixel est l'intensité du faisceau lumineux après avoir traversé l'épaisseur de l'échantillon, soit approximativement 30 microns pour une lame mince. Par conséquent, les images haute résolution provenant de certains types de roche ayant des cristaux en biseau (comme illustré dans la Figure 2.3) ou des éléments suffisamment petits pour pouvoir être empilés dans le sens de l'épaisseur de la lamelle de roche peuvent apparaître floues.

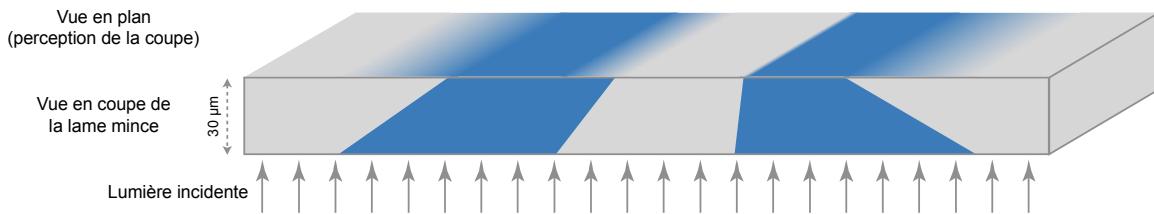


Figure 2.3 – Perception d'une lame mince en lumière transmise

### 2.1.2 Scanner à plat

Le scanner à plat sert communément à numériser des documents et des images argentiques, voire des diapositives. Il peut cependant aussi numériser, en lumière réfléchie, des échantillons de roche sciés. La surface plane peut être humidifiée afin d'obtenir un meilleur contraste de l'image. Pour ne pas endommager le scanner, il est important de le protéger en mettant un verre ou un plastique transparent entre la vitre et l'échantillon. Il est aussi possible de scanner du sédiment meuble placé dans une boîte transparente. La réflexion de lumière provoquée par des minéraux comme les micas sature les capteurs CCD et produit un effet similaire à la surexposition. Ce problème peut être contourné en intercalant un film polarisant entre l'échantillon et l'appareil. Certains scanners à plat avec dos pour transparents permettent la numérisation à haute résolution<sup>12</sup> de lames minces en lumière transmise. Toutefois, la qualité des images numérisées est inférieure à celles obtenue avec un scanner à diapositives du fait que le faisceau doit retraverser l'échantillon pour être collecté.

Les avantages de cette méthode sont sa simplicité de mise en œuvre et la possibilité qu'elle offre d'obtenir des images avec une vue globale de l'échantillon. En contrepartie, la qualité d'image est souvent moyenne.

### 2.1.3 Scanner à diapositives

Ce type de dispositif permet la numérisation en lumière transmise de lames minces. Ces dernières doivent pouvoir être insérées convenablement dans le support de l'appareil. La plupart des scanners offrent différents supports, dont au moins un dans lequel une lame mince peut être placée. L'utilisation du scanner à diapositives comporte plusieurs avantages. D'une

<sup>12</sup> Une résolution optique d'au moins 1200 dpi, sans interpolation, est nécessaire.

part, il permet une acquisition de toute la surface de la lame mince avec une luminosité constante (1999). D'autre part, sa résolution élevée (d'environ 4000 dpi) offre un niveau de détail généralement suffisant pour visualiser et analyser les composants pétrographiques. La contrepartie est la taille importante des images, qui peuvent dépasser les 100 MO (non compressées) pour une résolution de 4500 dpi. Cependant, de telles images correspondent aux capacités de visualisation et de traitement de JMicrоЩision.

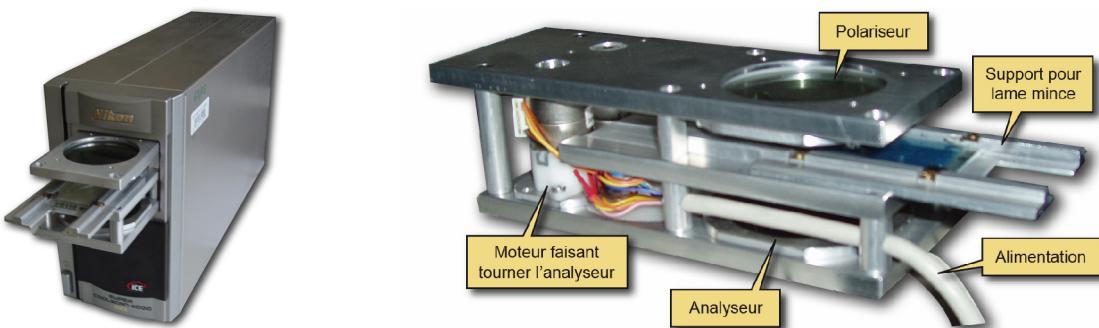


Figure 2.4 – Scanner à diapositives avec support adapté à l'acquisition en lumière polarisée

La modification du support à diapositives illustrée à la Figure 2.4 a été réalisée par Gilbert Overney<sup>13</sup> afin de permettre l'acquisition d'images en lumière polarisée. Dans ce dispositif, c'est l'ordinateur qui fait tourner le polariseur et l'analyseur. A défaut d'un tel support, la lumière peut être polarisée simplement en plaçant un filtre polarisant juste au-dessus et au-dessous de la lame mince.

#### 2.1.4 Microscope équipé d'une caméra

Une caméra montée sur un microscope ou une loupe binoculaire est un autre moyen de numériser des lames minces en lumière transmise ou en lumière réfléchie. Un appareil de photo numérique peut également remplacer la caméra à condition qu'il puisse être fixé correctement sur le microscope. L'acquisition est aussi possible sur des microscopes ayant d'autres types de source lumineuse, comme la fluorescence ou la cathodoluminescence. Toutefois, ces dispositifs nécessitent en principe des caméras plus sensibles, adaptées à une faible luminosité. Quant à la loupe binoculaire, elle permet de digitaliser différentes vues

<sup>13</sup> Technicien au sein de la Section des Sciences de la Terre de l'Université de Genève.

d'échantillons de roche ou de fossiles en lumière réfléchie ainsi que des lames minces en lumière transmise à des grossissements toutefois inférieurs à ceux du microscope traditionnel.

Ces dispositifs présentent deux avantages majeurs: a) ils permettent de numériser les vues variées pouvant être obtenues avec un microscope (en lumière normale, en lumière polarisée, à différents grossissements, etc.); b) l'image numérisée bénéficie de la qualité des propriétés optiques. Cependant, la capture se limite généralement à un champ restreint et quand l'échantillon peut être numérisé entièrement, l'image obtenue a une résolution moyenne, voire faible selon les caméras. Pour palier ce problème, des captures successives avec un certain chevauchement peuvent être transformées en une seule et même image par la technique du mosaïquage<sup>14</sup>. Le résultat de la reconstruction n'est toutefois jamais parfait, en raison notamment d'une illumination non uniforme. Par ailleurs, il est conseillé d'avoir un microscope équipé d'une platine motorisée pour effectuer les déplacements successifs avec une certaine précision.

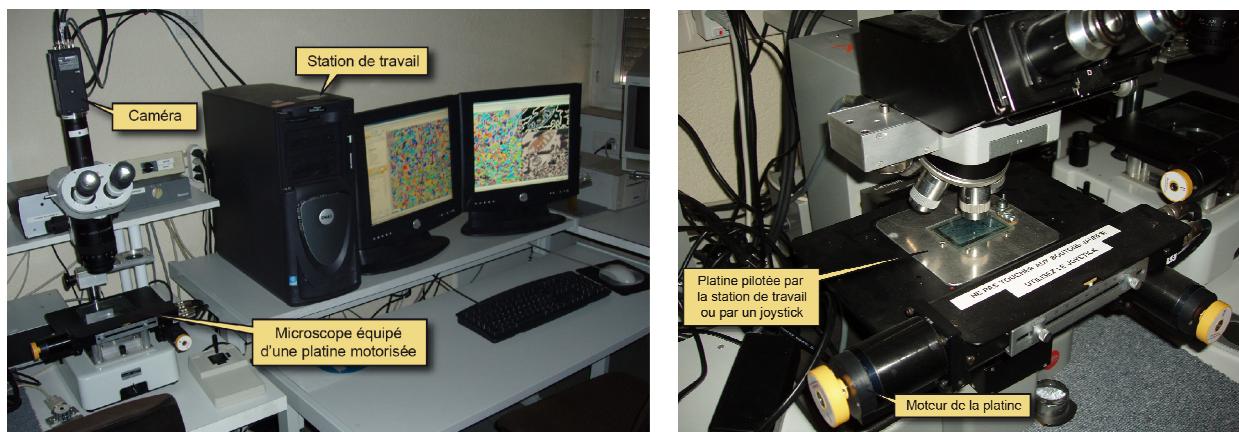


Figure 2.5 – Microscope équipé d'une caméra reliée à une station de travail

### 2.1.5 Microscope électronique à balayage (MEB)

L'imagerie en électrons rétrodiffusés du MEB permet de numériser la surface d'échantillons sciés ou de lames minces sans couvre-objet. Les électrons rétrodiffusés sont des électrons issus du faisceau primaire réémis avec une faible perte d'énergie. Les atomes les plus lourds de l'échantillon réémettent plus d'électrons que les atomes plus légers. Cette méthode rend

<sup>14</sup> Cette fonctionnalité n'est pas disponible dans JMicrоЩision.

donc compte de la composition chimique de la surface d'un échantillon en attribuant aux pixels de l'image des valeurs proportionnelles aux nombres atomiques. Elle permet un grossissement de 4000 fois au maximum et représente l'information de surface, du fait que le faisceau pénètre seulement de 0,1 à 0,2 microns dans l'échantillon. La préparation des échantillons, comme le polissage, est un facteur déterminant pour obtenir des images de qualité (Reed, 1996).

## 2.2 Prétraitements

### 2.2.1 Introduction

Les prétraitements exposés ici visent à améliorer la qualité des images en vue d'opérations de segmentation ou d'analyse d'image. Ils s'avèrent utiles lorsque les images brutes sont bruitées, qu'elles présentent des déformations géométriques ou qu'elles sont illuminées de façon non uniforme. Ils doivent néanmoins être employés avec précaution, car ils peuvent aussi altérer l'image. Dans JMicrоЩision, les opérations de traitement d'image ne modifient jamais directement l'image source. Elles sont construites à la demande (voir Section 1.2.4), ainsi plusieurs prétraitements peuvent être conservés (sans réellement créer les images) et être utilisés par la suite dans un processus afin de comparer leur efficacité.

Lors de la numérisation, l'opérateur peut sauvegarder l'image dans un format compressé sans perte (ZIP, LZW, RLE, etc.) ou avec perte (compression JPEG, fractale ou par ondelette). Il est cependant conseillé d'opter pour une compression sans perte afin de conserver autant que possible l'information pouvant être utile par la suite. En effet, la compression destructive consiste à niveler ou à écarter l'information la moins significative du point de vue de la perception visuelle humaine. Ainsi la transformation JPEG procède par sous-échantillonnage de la chrominance<sup>15</sup>. La conséquence de cette perte d'information colorimétrique peut affecter les traitements ultérieurs, comme la segmentation par seuillage (voir Section 3.3). La Figure 2.6 illustre clairement qu'un traitement identique sur les images (a) et (d), à première vue semblables, n'aboutit pas au même résultat.

---

<sup>15</sup> Représentation des valeurs relatives à la couleur d'une image.

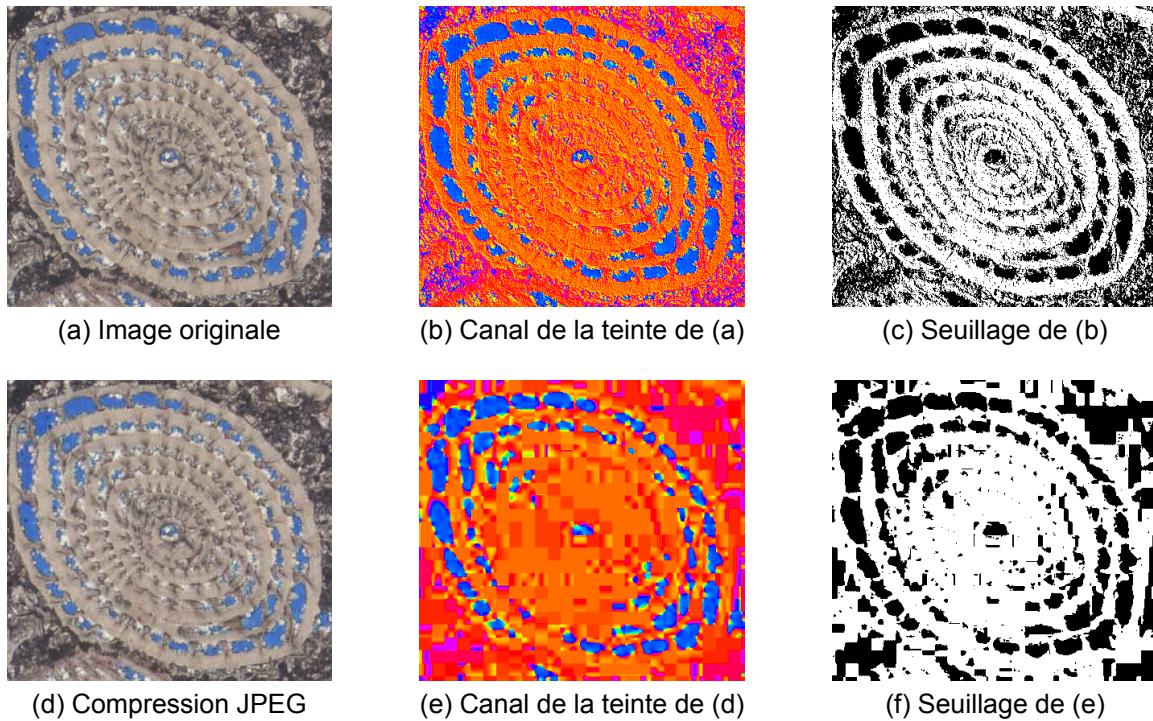


Figure 2.6 – Réduction de l'information chromatique par la compression JPEG

## 2.2.2 Opérations de filtrage

Dans le contexte des prétraitements, les opérations de filtrage servent principalement à l'homogénéisation des régions et à l'atténuation du bruit dans l'image. Le filtrage est un traitement d'image où le pixel de destination est calculé par une opération spécifique à partir d'un kernel, qui représente la structure et les valeurs propres au filtre (Russ, 1992). Il existe deux catégories de filtres spatiaux: d'une part les filtres linaires, les plus nombreux, qui englobent notamment les filtres moyenneurs et les filtres gaussiens; d'autre part les filtres non-linaires, comme les filtres médians, le filtre de Kuwahara et les filtres de morphologie mathématique (ou éléments structurants). Ces derniers, grâce notamment à leurs opérateurs d'érosion, de dilatation et de tophat, peuvent être utilisées pour corriger des défauts spécifiques comme les rayures (Russ, 1992).

L'homogénéisation des régions consiste à uniformiser localement les valeurs des pixels en vue de faciliter des opérations comme la segmentation (voir Chapitre 3). Les filtres moyenneurs et les filtres gaussiens ont certes cette propriété, mais ils ont des conséquences sur les bords des objets, puisqu'ils les déplacent et les rendent flous en lissant les zones de fort contraste. Dans

le cas d'un filtre moyenneur, un pixel est remplacé par la moyenne de ses voisins et de lui-même, le voisinage étant défini par le kernel. Avec un filtre gaussien, les valeurs sont pondérées de manière exponentielle, de la périphérie vers le centre du kernel, puis la moyenne est calculée. On préférera donc souvent recourir à un filtre médian qui, lui, remplace chaque pixel par la médiane de ses voisins. Il uniformise localement tout en préservant la position des bords des objets. Ce dernier point est important si l'on désire par la suite mesurer les objets. Quant au filtre de Kuwahara, il présente les mêmes avantages que le filtre médian. Son kernel est divisé en quatre sous-régions et remplace un pixel de destination par la moyenne de la sous-région ayant la plus faible variance.

Le bruit observé dans une image est généralement de nature aléatoire. Parfois, il peut aussi être impulsif, quand les capteurs sont défectueux, ou plus rarement périodique, quand il résulte d'interférences électriques ou de vibrations (Russ, 1992). Les sources du bruit aléatoire sont multiples: on peut citer les capteurs CCD, l'électronique qui véhicule ou amplifie le signal, l'éclairage instable et la poussière. Certains types de matériel d'acquisition atténuent d'eux-mêmes le bruit en produisant une image résultant d'une moyenne d'images capturées à différents intervalles de temps. L'amplitude du bruit n'est pas totalement supprimée, mais elle est nettement diminuée.

Comme il existe différents types de bruit, plusieurs approches sont envisageables pour l'atténuer. Les méthodes les plus efficaces sont les mêmes que celles proposées pour l'homogénéisation, car le bruit est généralement représenté par des pixels avec des valeurs différentes de leur voisinage proche, disséminés dans l'image. Dans un contexte fréquentiel, le bruit est perçu comme une perturbation dans les hautes fréquences. Les filtres moyenneurs et les filtres gaussiens sont des filtres passe-bas qui éliminent les hautes fréquences, réduisant ainsi le bruit et rendant l'image floue. Ces deux filtres se révèlent efficaces contre le bruit aléatoire, mais beaucoup moins contre le bruit impulsif. Entre les deux, c'est le filtre gaussien qui préserve le mieux les zones de fort contraste. Quant au filtre médian, il se montre particulièrement efficace contre le bruit impulsif, mais assez peu contre le bruit aléatoire.

### 2.2.3 Soustraction d'un fond non uniforme

Lorsqu'un échantillon est éclairé par une source lumineuse ponctuelle ou que la lumière périphérique est absorbée par la caméra, l'image numérisée présente une luminosité variable, qui peut s'avérer gênante lors de traitements ultérieurs (voir Figure 2.7 (e)). Pour atténuer ce défaut, l'image originale doit être soustraite d'une image représentant uniquement le fond. Une façon simple et efficace de procéder est d'effectuer une capture sans échantillon, avec les mêmes réglages que pour l'acquisition de l'image originale. Dans les cas où il n'est pas possible d'avoir une telle capture, le fond doit être construit à partir de l'information contenue dans l'image. Dans JMicrоЩision, le fond artificiel est construit à partir d'au moins six points placés sur le fond par l'opérateur. Ces points doivent être répartis de manière que la fonction analytique polynomiale puisse restituer au mieux l'éclairage non uniforme (Russ, 1992). Une autre façon de créer un fond artificiel est de faire disparaître les objets de l'image en appliquant de manière répétée l'opérateur d'érosion (morphologie mathématique).

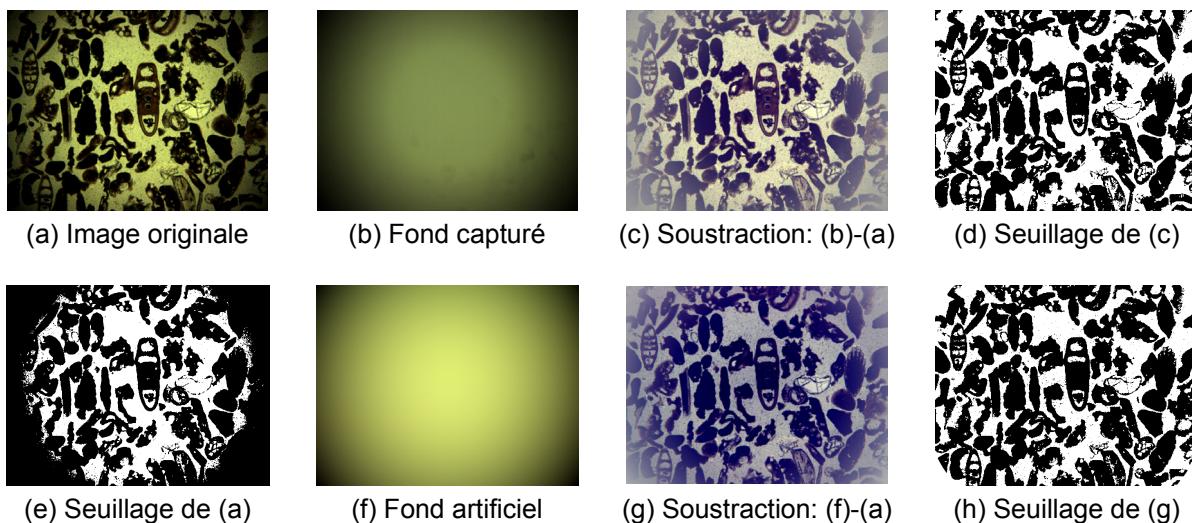
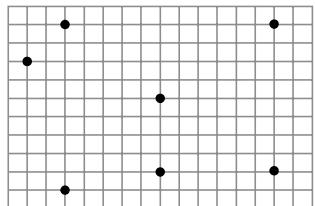


Figure 2.7 – Soustraction du fond non uniforme

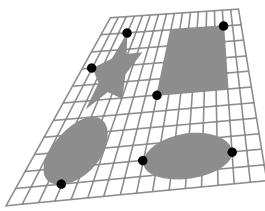
### 2.2.4 Rectification géométrique

L'outil de rectification géométrique implémenté dans JMicrоЩision permet de corriger une image à partir de points de contrôle. Ce procédé nécessite deux images: l'image de référence non déformée et l'image déformée à traiter. La distorsion entre les deux images est estimée au moyen des points de contrôle, en résolvant un système d'équations polynomiales (Russ, 1992). Cet outil peut servir à corriger les défauts d'un système d'acquisition, en capturant

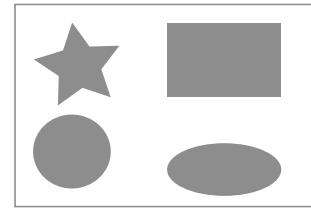
d'abord une grille pour estimer la distorsion et ensuite en appliquant la transformation aux images numérisées. Cependant, il est plus souvent utilisé pour superposer des images représentant une même scène qui ont été numérisées par des appareils différents ou dans des conditions différentes.



(a) Référence



(b) Image déformée



(c) Image rectifiée

Figure 2.8 – Correction de la déformation à partir de points de contrôle

# Chapitre 3 : Segmentation

## 3.1 Introduction

La segmentation est une étape clé dans le processus d'analyse d'image. Elle consiste à subdiviser l'image en régions présentant un intérêt pour l'opérateur. Cependant, il n'existe ni théorie générale ni méthode universelle de segmentation d'image. Par conséquent, on trouve une multitude de méthodes qui se limitent souvent à des applications particulières. Lorsqu'il s'agit de traiter des images avec un contenu complexe, la segmentation devient une étape très difficile, voire parfois impossible. Or les images pétrographiques peuvent être considérées dans la majorité des cas comme complexes, plus particulièrement quand elles contiennent des composants avec des bords mal définis, de faibles contrastes ou des textures. Dans ce chapitre, seules les méthodes de segmentation implémentées dans JMicrоЩision sont abordées.

Les algorithmes de segmentation reposent sur deux types de critères: la similarité et la discontinuité. Dans JMicrоЩision, plusieurs algorithmes ayant pour objectif de regrouper les pixels avec des valeurs d'intensité similaires ont été élaborés. Parmi eux, on distingue deux catégories: ceux basés uniquement sur la similarité des pixels (segmentation par seuillage et k-means) et ceux qui tiennent compte en plus des relations de voisinage des pixels (outil baguette magique et ligne de partage des eaux). Par contre, aucun algorithme basé exclusivement sur la discontinuité ne semblait convenir à la segmentation des images pétrographiques. En particulier, les algorithmes de détection et de fermeture des contours ne nous apparaissaient pas suffisamment robustes pour les images où les changements de contraste ne coïncident pas toujours avec les frontières des régions d'intérêt. La détection des

contours est cependant intégrée dans l'algorithme de segmentation par la ligne de partage des eaux en combinaison avec la croissance de régions (basée sur la notion de similarité).

L'acquisition et les prétraitements, abordés dans le Chapitre 2, restent des étapes importantes, car des choix appropriés en ce qui concerne le matériel d'acquisition et les éventuels prétraitements permettent d'augmenter le contraste dans l'image et d'homogénéiser les régions. La division de l'image en régions s'en trouve facilitée du fait que les critères de discontinuité et de similarité sont renforcés.

## 3.2 Définitions préliminaires

### 3.2.1 Types d'image

Les formats d'image varient principalement selon des spécifications comme la profondeur des pixels, l'espace colorimétrique, le support pour la compression et les métadonnées. Le format TIFF, le plus communément utilisé, est celui qui offre la plus grande liberté en termes de représentation et de structure. Néanmoins, très rares sont les logiciels qui prennent en charge toutes ses spécifications et ses versions. Seules les spécifications de base sont habituellement incorporées dans les logiciels lisant le format TIFF.

JMicroVision, au travers des librairies JAI, lit les formats d'image TIFF, BMP, FlashPiX, GIF, JPEG, PNG et PNM. Une spécification comme la profondeur des pixels de 16 bits/canal n'est pas directement prise en charge, car elle aurait pour conséquence d'alourdir considérablement les algorithmes, uniquement pour pouvoir traiter un type d'image marginal. Afin de faciliter l'élaboration des algorithmes de traitement et d'analyse d'image, il semblait plus judicieux de réduire toutes les spécifications des formats aux trois types d'image suivants:

- image binaire avec 1 bande et un codage de 1 bit par pixel.
- image en niveaux de gris avec 1 bande et un codage de 8 bits par pixel.
- image couleur avec 3 bandes, un modèle de couleurs RVB (rouge, vert et bleu) et un codage de 24 bits par pixel (3x8).

L'adaptation des spécifications dès l'ouverture d'un fichier image assure une plus grande stabilité à JMicrоЩision, mais conduit parfois à des pertes d'information, par exemple:

- image couleur (3 bandes + canal de transparence alpha)  $\Rightarrow$  image couleur RVB, où le canal alpha a été supprimé.
- image en niveaux de gris (16 bits par pixel, 65'536 niveaux)  $\Rightarrow$  image en niveaux de gris (8 bits par pixel), où les niveaux de gris ont été normalisés entre 0 et 255.

En revanche, la transformation d'une image en couleurs indexées (1 bande + 1 palette de couleurs) en une image couleur RVB n'entraîne aucune perte d'information.

Les images codées sur 16 ou 32 bits par bande présentent un intérêt dans des domaines comme la radiologie pour capter de faibles nuances dans les niveaux sombres ou clairs. Cependant les écrans actuels ne permettent pas d'afficher un tel codage, si bien que l'image doit être vue par tranches de 256 niveaux d'intensité. L'intérêt d'un tel codage réside plus dans sa capacité accrue de stockage de l'information que dans la visualisation de cette information, car le système visuel humain n'est de toute façon pas en mesure de différencier 256 niveaux de gris (codage sur 8 bits) ou 16,7 millions de couleurs (codage de 3x8 bits). En pétrographie, de telles images ne semblent pas apporter plus d'information significative que les images traditionnelles de 8 bits/canal.

### 3.2.1.1 Image binaire

Une image binaire ne contient que deux valeurs d'intensité. Par convention, dans les logiciels de traitement d'image, les pixels "allumés" (blanc ou d'une autre couleur) représentent les objets d'intérêt et les pixels "éteints" représentent le fond (noir). JMicrоЩision utilise cette convention. Par contre, pour toutes les figures présentées tout au long de ce manuscrit, les pixels éteints prennent la couleur du papier (blanc) et les pixels allumés celle de l'encre (noir).

Les images binaires sont souvent le résultat d'une segmentation où les pixels ont été séparés en deux catégories: les objets d'intérêt et le fond (voir Section 4.2). Elles présentent des propriétés topologiques particulières, qui permettent d'effectuer des opérations comme le remplissage des trous, la suppression des objets touchant le bord et la reconstruction à partir

de deux images. Les autres principaux traitements appliqués aux images binaires sont les opérations booléennes et les opérations de morphologie mathématique.

### 3.2.1.2 Image en niveaux de gris

Les niveaux de gris correspondent aux valeurs des pixels et expriment généralement l'intensité lumineuse de la scène capturée. Parfois, ils sont le résultat de valeurs d'intensité relatives à des propriétés physiques ou chimiques, par exemple dans une image en fluorescence ou en électrons rétrodiffusés.

Dans JMicroVision, les niveaux d'intensité varient entre 0 et 255. La transformation d'une image couleur en niveaux de gris se fait selon la recommandation 601 proposée par la Commission Internationale de l'Éclairage (CIE), où un pixel RVB est converti en un niveau de gris de la façon suivante:

$$\text{Gris} = 0.299 \text{ Rouge} + 0.587 \text{ Vert} + 0.114 \text{ Bleu} \quad (3.1)$$

Cette formule rend compte de la manière dont l'œil humain perçoit les trois composantes de la lumière (rouge, vert et bleu). Un coefficient pondère ces composantes par rapport à la sensibilité de notre perception.

### 3.2.1.3 Image couleur

Dans JMicroVision, les images couleurs sont composées de trois bandes où la couleur peut être représentée à partir de deux systèmes:

- le profil ICC (International Color Consortium) sRGB (standard red, green and blue), inclus dans le langage Java. L'espace colorimétrique sRGB, qui découle du modèle de couleurs rouge, vert et bleu (RVB ou RGB en anglais), a été défini par Microsoft et Hewlett-Packard (Stokes *et al.*, 1996)<sup>16</sup> dans le but de faire correspondre les couleurs entre des applications telles que les écrans, les scanners, les imprimantes et les caméras numériques.
- l'espace colorimétrique TLS (teinte, luminance et saturation ou en anglais HSI: hue, saturation and intensity). Fondé sur l'expérience intuitive de la perception humaine de la

---

<sup>16</sup> Cet article est directement disponible sur le site W3C à l'adresse <http://www.w3.org/Graphics/Color/sRGB>

couleur, le système TLS est beaucoup plus facile à utiliser que le modèle RVB. Les trois composantes de l'espace colorimétrique TLS sont:

- la luminance: indique l'intensité de lumière perçue indépendamment de la couleur. Dans l'espace colorimétrique, elle s'étend du noir au blanc de bas en haut. Les nuances de gris intermédiaires sont positionnées sur l'axe central.
- la teinte: indique la dominante de couleur perçue. Dans l'espace colorimétrique, elle est déterminée par une position angulaire autour de l'axe vertical, où les six principales valeurs sont le rouge à  $0^\circ$ , le jaune à  $60^\circ$ , le vert à  $120^\circ$ , le cyan à  $180^\circ$ , le bleu à  $240^\circ$  et le magenta à  $300^\circ$ .
- la saturation: exprime la pureté de la teinte dominante. Dans l'espace colorimétrique, la distance par rapport à l'axe central indique la valeur de saturation, qui croît du centre vers la périphérie.

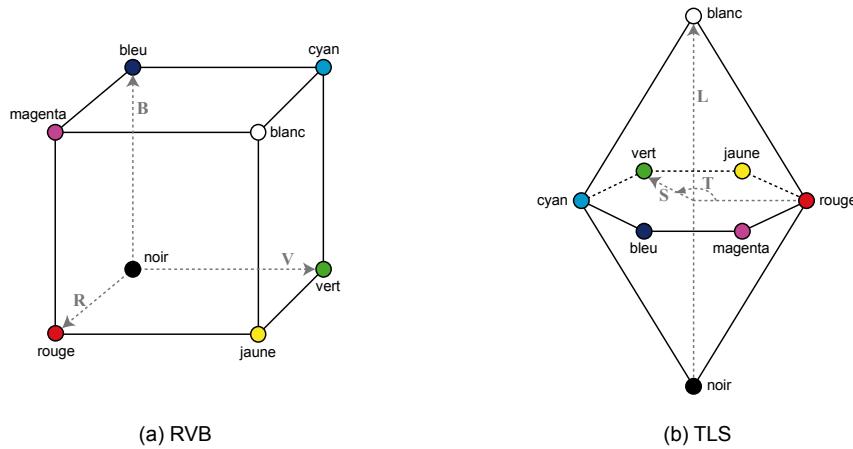


Figure 3.1 – Espaces colorimétriques RVB et TLS

L'espace TLS facilite beaucoup la sélection ou la manipulation des couleurs, par exemple dans le cas de la segmentation d'une image couleur par seuillage (voir Section 3.3). La conversion entre les deux espaces colorimétriques est décrite en détail dans l'ouvrage *Practical Algorithms for Image Analysis* (Seul et al., 2000).

### 3.2.2 Définition d'une région

La segmentation a pour but de subdiviser l'image en régions sur la base de critères relatifs aux valeurs des pixels, comme la couleur (la teinte dans le modèle TLS). Une région est donc un regroupement de pixels connexes ayant des propriétés communes. Dans une image binaire,

seuls deux types de régions cohabitent, les objets d'intérêt et les régions restantes, appelées "fond".

### 3.2.3 Géométrie discrète et connexité des pixels

Pour obtenir une représentation pertinente de l'information, il convient dans un premier temps de regrouper les pixels ayant des caractéristiques communes. Ces agrégats peuvent être définis comme des primitives d'image. On distinguera deux types de primitives: celles représentant une région et celles représentant un contour.

Pour un maillage carré de pixels, la relation d'adjacence est soit 4-connexe soit 8-connexe. En connexité 4, un pixel a 4 voisins situés à égale distance, qu'il touche par une arête. En connexité 8, un pixel possède 8 voisins, qu'il touche par une arête ou par un sommet. Les pixels situés sur les directions verticales et horizontales sont à une distance euclidienne de 1, tandis que les voisins dans les directions diagonales sont à une distance de racine de deux.

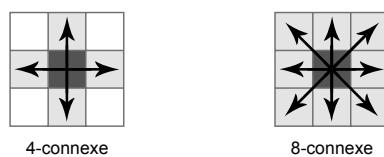


Figure 3.2 – Relation de voisinage des pixels.

Un maillage carré rend les principes topologiques plus complexes. De fait, il est nécessaire d'opposer la connexité des objets et celle du fond pour respecter le théorème de Jordan<sup>17</sup>. Autrement dit, le fond doit obligatoirement être 4-connexe quand les objets sont 8-connexes ou inversement (voir Section 4.2).

Il est possible de simplifier les relations de voisinage en choisissant un maillage hexagonal. Avec un tel maillage, les objets et le fond sont toujours 6-connexes et la distance entre les pixels est constante. Cependant, comme la structure de données d'une image numérique est formée d'un maillage carré de pixels, le maillage hexagonal est plus difficile à mettre en

---

<sup>17</sup> Ce théorème énonce que toute courbe fermée qui ne se recoupe pas sépare les points du plan en deux domaines connexes, l'intérieur et l'extérieur de la courbe.

œuvre. De plus, il introduit un biais car, en général, les capteurs des dispositifs d'acquisition sont construits avec un maillage carré. Pour ces raisons, les algorithmes de JMicrOVision sont construits exclusivement selon un maillage carré.

### 3.2.4 Algorithme de croissance de régions

L'algorithme de croissance de régions (region growing) est la procédure de regroupement des pixels connexes sur la base de critères de similarité. La croissance démarre à partir de germes (seeds). Les pixels représentant ces germes sont définis manuellement par l'opérateur ou de façon automatique selon différents critères, par exemple à partir d'une gamme de valeurs d'intensité. Le pseudo-code ci-dessous présente une procédure récursive, où  $(i_c, j_c)$  représente la position de tous les pixels connexes (4-connexes ou 8-connexes) à la position  $(i, j)$ . Les arguments transmis à la procédure avant la première récursion sont les valeurs représentatives d'un germe, à savoir sa position  $(i, j)$  et une valeur (label) qui le distingue des autres germes. La table de référence `visité[]` est un tableau booléen servant à mémoriser les endroits déjà visités par la croissance. L'algorithme fabrique au final une l'image labellisée où chaque région est caractérisée par une même valeur d'intensité. Le pseudo-code de croissance d'une région s'exprime de la façon suivante:

```
Croissance(i, j, label){
    Visité[i, j] = VRAI;
    Image_labellisée(i, j) = label;
    POUR chaque position connexe à (i, j) FAIRE
        SI (Visité[ic, jc] == FAUX) ET (Image_source(ic, jc) répond aux critères de
            croissance) ALORS
            Croissance(ic, jc, label);
    FIN POUR
}
```

Dans JMicrOVision, l'outil baguette magique, la segmentation par la ligne de partage des eaux (voir Section 3.5) et l'extraction d'objets (voir Section 4.4) exploitent cet algorithme. Les critères de croissance de la baguette magique sont la luminance, la teinte et la saturation pour une image couleur, et la luminance pour une image en niveaux de gris. Toutefois, ces critères peuvent se révéler insuffisants lorsque les objets d'intérêt contiennent des textures.

### 3.3 Segmentation par seuillage

La segmentation par seuillage est la méthode la plus utilisée en raison de sa simplicité et de sa rapidité d'exécution. Elle se base exclusivement sur la distribution des niveaux de gris de l'histogramme de l'image et par conséquent ne tient pas compte des relations de voisinage entre les pixels. Théoriquement, chaque mode de l'histogramme doit correspondre à une classe d'objets ou au fond. Malheureusement, dans la pratique, on observe fréquemment un amalgame de plusieurs classes dans un mode. Cette complication apparaît lorsque différentes classes d'objets présentent des valeurs d'intensité similaires ou lorsque les objets d'intérêt ou le fond n'ont pas des valeurs d'intensité constantes. En effet, le fond n'est souvent pas uniforme en microscopie. Néanmoins, une correction tendant à l'uniformiser peut être apportée (voir Section 2.2.3).

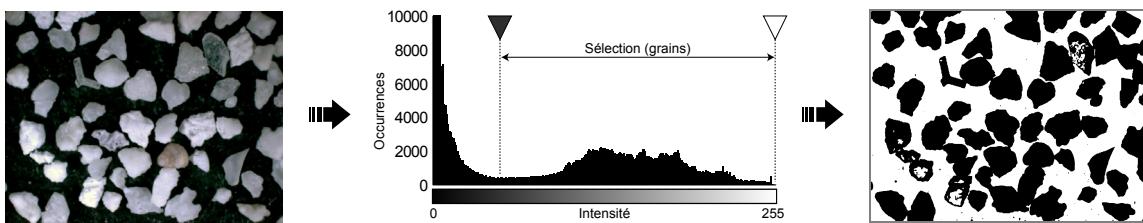


Figure 3.3 – Segmentation par seuillage de l'histogramme

Dans JMicroVision, l'opérateur doit ajuster sur l'histogramme les deux seuils qui définissent l'étendue des valeurs d'intensité représentant au mieux une classe d'objets. Sur une image couleur, l'ajustement s'effectue en combinant les valeurs de l'histogramme, à savoir celles des canaux rouge, vert et bleu ou celles des canaux luminance, teinte et saturation. À ce propos, la sélection dans l'espace colorimétrique TLS est bien plus intuitive pour l'opérateur (voir Section 3.2.1.3). La rapidité d'exécution du seuillage permet d'afficher en temps réel le résultat de la sélection par une couleur particulière en superposition de l'image. Pour faciliter la tâche de l'utilisateur, les valeurs de seuil peuvent être prises directement sur une zone de l'image avec l'outil pipette. Les paramètres ajustables de cet outil sont la taille de la zone de capture et la tolérance, dont le réglage a pour effet d'abaisser le seuil inférieur et d'augmenter le seuil supérieur. L'automatisation ultime de la segmentation par seuillage est la

détermination automatique des valeurs de seuil. JMicroVision propose plusieurs méthodes<sup>18</sup> de seuillage automatique séparant l'histogramme en deux classes. Ces méthodes sont les suivantes: la recherche itérative des vallées, la variance maximum, l'entropie maximum, l'erreur minimum et l'incertitude minimum.

### 3.4 k-means

La méthode k-means ou "des nuées dynamiques", introduite par McQueen (1967), fait partie de la catégorie des classifications non supervisées. Dans le cadre de ce travail, elle a pour objectif de regrouper les pixels avec des valeurs d'intensité similaires dans un certain nombre ( $k$ ) de classes. L'algorithme des k-means cherche à minimiser la variance intra-classe définie par la fonction suivante:

$$V = \sum_{i=1}^k \sum_{j=1}^n (x_j^{(i)} - \mu_i)^2 \quad (3.2)$$

où  $k$  est le nombre de classes,  $n$  le nombre de pixels,  $x$  la valeur d'un pixel et  $\mu_i$  le noyau (ou le centroïde, soit la moyenne de tous les points d'une classe).

Il existe plusieurs variantes pour définir le nombre de classes et l'initialisation des noyaux. Dans JMicroVision, le nombre de classes est choisi par l'opérateur et la position initiale des noyaux est répartie de façon uniforme dans l'espace. Avant d'exécuter la classification, l'opérateur définit les paramètres suivants:

- le nombre de classes ( $k$ )
- le nombre maximum d'itérations (Max\_iteration)
- le seuil d'erreur (Epsilon)

Le nombre maximum d'itérations détermine le nombre de fois que les noyaux sont recalculés. Quant au seuil d'erreur, il arrête le processus de minimisation lorsque Epsilon est inférieur à la différence entre la variance intra-classe courante et celle obtenue à l'itération précédente. Pour

---

<sup>18</sup> Ces méthodes ne sont pas développées ici, car elles proviennent des bibliothèques JAI et ne sont pas suffisamment documentées pour être discutées.

les images couleurs, la classification s'effectue dans un espace colorimétrique tridimensionnel. L'attribution d'une classe à un pixel est déterminée par la proximité de son noyau avec la valeur du pixel. Lorsque l'ensemble des pixels ont été traités, la position du noyau dans l'espace colorimétrique est recalculée pour chaque classe en effectuant la moyenne des valeurs des pixels qui lui sont associés. Ensuite, le processus d'attribution est relancé avec les nouvelles positions des noyaux. Ces opérations sont répétées jusqu'à ce que les conditions d'arrêt évoquées précédemment soient remplies. Le pseudo-code des k-means se présente de la façon suivante:

```

Initialisation de la position des noyaux dans l'espace colorimétrique;
i = 0;
TANT QUE (Epsilon < |Vi-Vi-1| ET i < Max_iteration) FAIRE
    POUR chaque tuile FAIRE
        POUR chaque pixel FAIRE
            Attribution de la classe dont le noyau est le plus proche;
        FIN POUR
    FIN POUR
    Calcul de la position des noyaux;
    i = i + 1;

```

L'algorithme ci-dessus respecte la technique du tuilage, dont l'avantage principal réside dans le fait que le système n'a jamais besoin de charger plus d'une tuile en mémoire simultanément (voir Section 1.2.4). Dans JMicroVision, une fois l'opération des k-means terminée, seule l'information concernant la position des noyaux est conservée d'une manière permanente. Par la suite, une tuile de l'image pourra être construite à la demande par l'attribution à chaque pixel de la valeur correspondant à la classe dont le noyau est le plus proche.

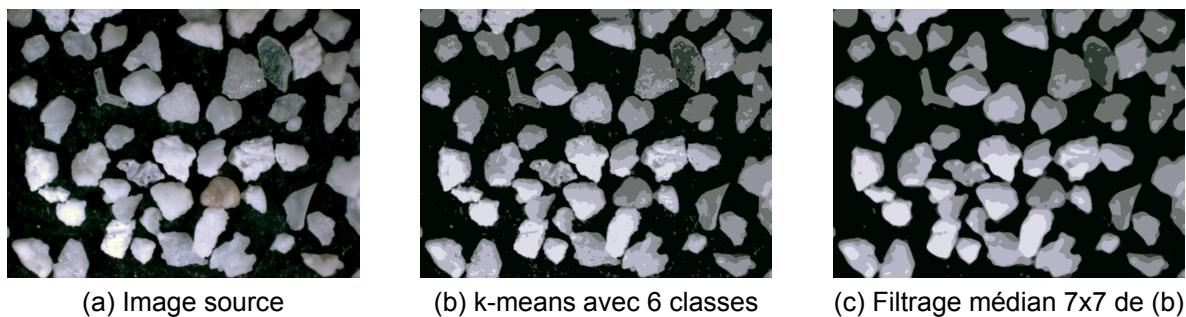


Figure 3.4 – Segmentation par l'algorithme des k-means

A l'instar de la méthode de segmentation par seuillage, cet algorithme ne tient pas compte des relations de voisinage des pixels. Pour uniformiser localement l'image, un post-traitement à

l'aide d'un filtre non linéaire (voir Section 2.2.2), comme un filtre médian, peut néanmoins être appliqué (voir Figure 3.4 (c)).

### 3.5 Ligne de partage des eaux (LPE)

La segmentation par la ligne de partage des eaux (Beucher & Lantuejoul, 1979) est une méthode issue de la morphologie mathématique. Le concept de la LPE provient de l'hydrogéologie et fait appel aux notions de bassins versants et de lignes de crêtes. Toutefois, à l'opposé d'une réalité hydrogéologique, l'objectif de la LPE est l'inondation des vallées à partir des minima (ou cuvettes). Le niveau d'eau s'élève progressivement jusqu'à ce que les eaux provenant de minima différents se rencontrent. Pour empêcher le mélange des eaux, on construit alors un barrage à l'interface des eaux. Lorsque la topologie est complètement inondée, le processus est terminé et les barrages représentent la ligne de partage des eaux. La Figure 3.5 illustre ce principe d'inondation en seulement deux dimensions, les vallées devant être considérées comme des bassins versants isolés qui ne sont pas reliés entre eux dans la troisième dimension non représentée (y). L'eau se met à jaillir des minima uniquement lorsque l'élévation générale du niveau d'eau atteint l'altitude des minima.

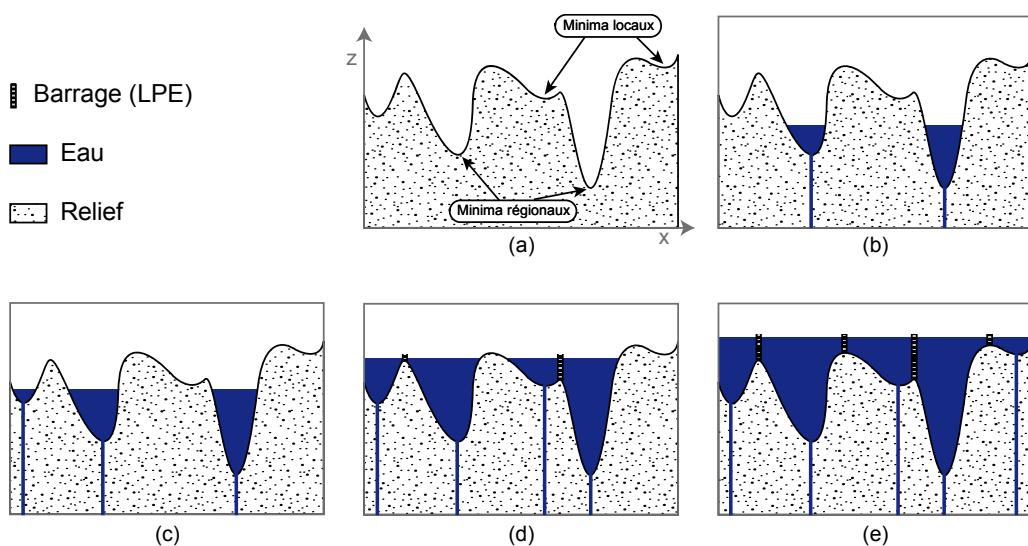
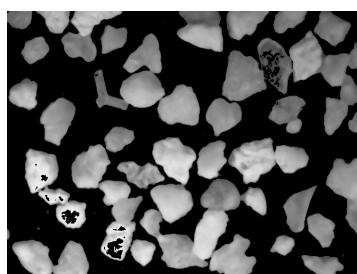


Figure 3.5 – Principe de la ligne de partage des eaux

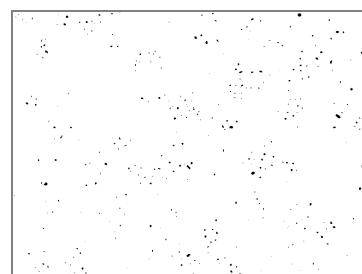
En transposant le concept à la segmentation d'image, la topologie se transforme en une image à deux dimensions ( $x,y$ ) à laquelle on ajoute l'altitude ( $z$ ) construite à partir des valeurs d'intensité des pixels.

L'application de la LPE sur une image quelconque provoque systématiquement une sursegmentation à cause de la présence de nombreux minima locaux. Il s'agit donc de supprimer les petites cuvettes conduisant aux séparations non désirées. Une première technique consiste à uniformiser localement les zones de l'image source par un filtre médian. Toutefois, après cette opération, le nombre de régions est encore trop élevé (voir Figure 3.6 (c)). Pour réduire les régions à un nombre acceptable, il faut donc recourir à un deuxième traitement. Deux types de méthode sont possibles: la LPE construite à partir de marqueurs (voir Section 3.5.1) et le regroupement a posteriori de régions selon divers critères. Cette dernière méthode n'est pas abordée dans le cadre de ce travail.

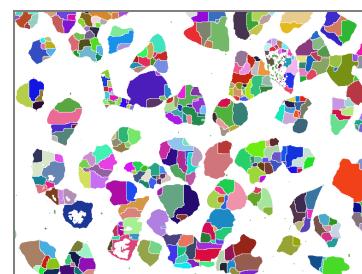
Une autre amélioration de la LPE illustrée dans la Figure 3.6 est l'utilisation de l'image gradient, qui offre de meilleures lignes de crêtes. Ces lignes, placées sur les zones de fort contraste, délimitent généralement mieux les régions d'intérêt que celles résultant des valeurs extrêmes de luminance ou de couleur. Toutefois, le recours à l'image gradient accentue la sursegmentation en raison des nombreux minima locaux.



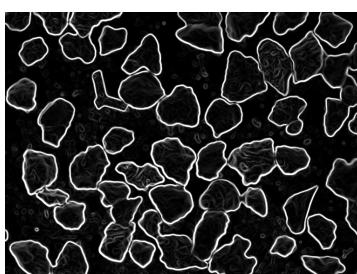
(a) Soustraction du fond, puis filtrage médian 5x5



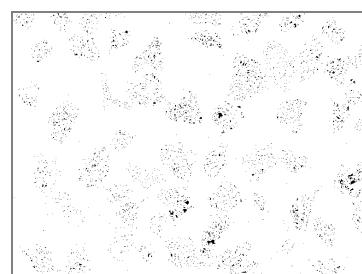
(b) Minima de (a)



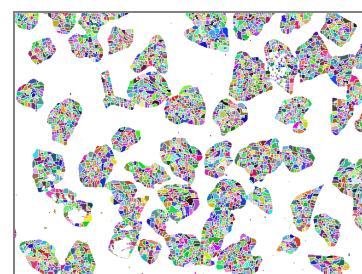
(c) Segmentation de (a) par LPE



(d) Gradient de (a)



(e) Minima de (d)

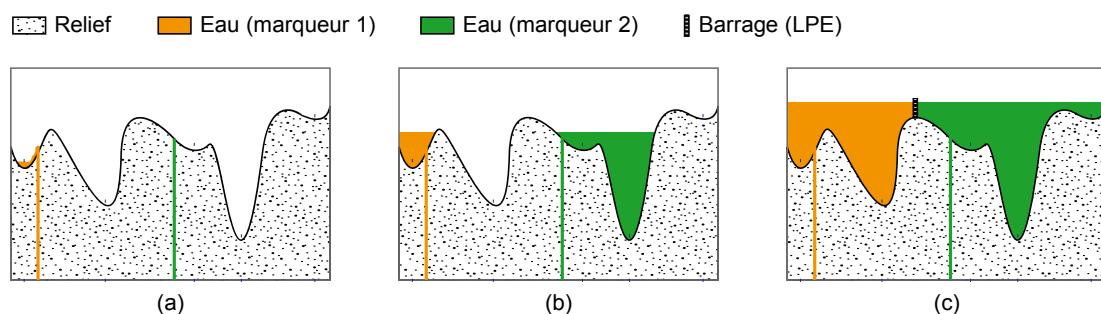


(f) Segmentation de (d) par LPE

Figure 3.6 – Segmentation par LPE

### 3.5.1 Segmentation à l'aide de marqueurs

La segmentation construite à partir de marqueurs a été introduite par Meyer et Beucher (1990) et s'avère une solution efficace au problème de la sur-segmentation. Sur le relief dans la Figure 3.7, l'eau ne jaillit plus des cuvettes, mais uniquement de l'endroit où les marqueurs sont positionnés. Par conséquent, le nombre de régions obtenu dans l'image segmentée correspond strictement au nombre de marqueurs. Une des propriétés les plus intéressantes de cet algorithme est sa robustesse quant à la variabilité de la position des marqueurs.



Dans JMicrоЩision, l'image source est, si nécessaire, convertie en niveaux de gris et puis traitée avec un filtre médian qui uniformise localement les régions tout en préservant la position de leurs bords. Ensuite, un filtre de Sobel<sup>19</sup> horizontal et vertical est appliqué pour obtenir l'image du gradient des intensités, où les lignes de crêtes se placent sur les zones à fort contraste. Les marqueurs sont générés soit par l'opérateur, qui les dessine, soit par une image en niveaux de gris où chaque région d'un même niveau représente un marqueur.

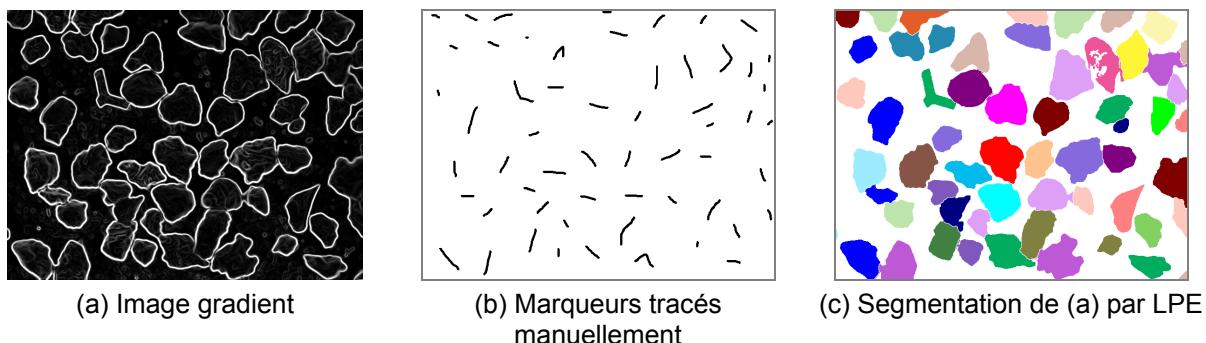


Figure 3.8 – Segmentation par LPE à l'aide de marqueurs

<sup>19</sup> Opérateur qui calcule le gradient de l'intensité de chaque pixel.

Dans le pseudo-code ci-dessous, le type `Queue` est une structure de données de 256 files d'attente de type FIFO. Chaque file d'attente représente un niveau de gris et contient des instances du type `Elément`, où sont stockés la position d'un pixel et une valeur d'identification liée au marqueur auquel le pixel est rattaché (label). Ces files d'attente sont hiérarchisées dans une pile selon les niveaux de gris, le premier élément à extraire étant la file d'attente du niveau 0. Cette hiérarchisation des niveaux de gris permet d'élever l'inondation de manière uniforme sur toute l'image.

```

Img_Source = Gradient_Sobel(Img_Source);
Etiquette = 2;
POUR chaque pixel FAIRE
    SI (Img_Marqueurs(i,j) != 0) ALORS
        Croissance(i,j,Etiquette);
        où Image_labelisée est représentée par Img_LPE, et le tableau
        Visité[] par Img_Marqueurs qui reçoit 0 comme valeur à la place de
        VRAI (voir l'algorithme de la Section 3.2.4).
        Etiquette = Etiquette + 1;
    FIN POUR
    POUR chaque pixel FAIRE
        SI (Img_LPE(i,j) != 0) ALORS
            POUR chaque position connexe à (i,j) FAIRE
                SI (Img_LPE(i,j) != Img_LPE(ic,jc)) ALORS
                    Queue.ajout(i, j, Img_Source(i,j), Img_LPE(i,j));
                    où le 3ème paramètre donne le niveau de gris de la file
                    d'attente et le 4ème paramètre donne la valeur spécifique à
                    un marqueur.
                FIN POUR
                SINON Img_LPE(i,j) = 1;
            FIN POUR
            TANT QUE (Queue contient des éléments) FAIRE
                Element = Queue.retirer();
                i = Element.i;
                j = Element.j;
                POUR chaque position connexe à (i,j) FAIRE
                    Etiquette = Img_LPE(ic,jc);
                    SI (Etiquette == 1) ALORS
                        Img_LPE(ic,jc) = Element.label;
                        Queue.ajout(ic, jc, Img_LPE(i,j), Element.label);
                    SINON SI (la ligne de partage des eaux est visible ET Etiquette > 1 ET
                        Etiquette < Element.label) ALORS
                        Img_LPE(ic,jc) = 0;
                FIN POUR
        FIN POUR
    FIN POUR

```

La première boucle de l'algorithme ci-dessus permet d'identifier les régions correspondant aux marqueurs. La deuxième boucle ajoute dans les files d'attente les pixels de bordure des marqueurs (front de propagation de l'inondation). Dans la dernière partie, le processus inonde

progressivement depuis les niveaux sombres jusqu'aux niveaux clairs et donne la possibilité d'afficher les lignes de partage des eaux en attribuant une valeur 0 à l'image de la LPE. Pour simplifier le pseudo-code, certaines propriétés présentes dans JMicroVision, comme la fusion de régions connexes appartenant à un marqueur de même classe et l'exclusion de certains pixels de la segmentation en fixant un seuil, n'ont pas été intégrées.

La LPE construite au moyen des marqueurs souffre cependant de deux problèmes majeurs. Lorsqu'il s'agit de séparer un grand nombre de régions d'intérêt, la création automatique de marqueurs devient indispensable. Or, l'extraction automatique des marqueurs s'effectue difficilement. Cependant, on y arrive dans certains cas par des opérations de seuillage et d'érosion successive ou, pour les images binaires, par l'érodé ultime de l'image distance (voir Section 3.5.2). L'autre difficulté provient du fait que les lignes de crêtes définies par l'image gradient ne coïncident pas forcément avec celles désirées. Par ailleurs, lorsque la couleur différencie de façon déterminante les régions d'intérêt, l'image de la teinte selon le modèle TLS est bien plus adaptée comme image source que celle de la luminance (résultat de la conversion d'une image couleur), qui fixe les lignes de crêtes au changement de saturation ou de luminosité.

### 3.5.2 Segmentation d'images binaires par l'érodé ultime

La séparation d'objets connexes dans une image binaire est une tâche difficile à réaliser. La LPE est fréquemment utilisée pour résoudre ce problème. Seul le début de l'algorithme se différencie alors du pseudo-code de la section précédente. Il s'exprime de la manière suivante:

```

img_Source = Distance(img_Source);
img_Source = Filtrage_Gaussien_5x5(img_Source);
img_Marqueurs = Erodé_Ultieme(img_Source);
img_Source = Inverse(img_Source);

Etiquette = 2;
POUR chaque pixel FAIRE
... suite identique au pseudo-code de la segmentation à l'aide des marqueurs

```

Ainsi, l'algorithme ci-dessus est identique à celui de la LPE à l'aide de marqueurs, mis à part les images en entrée: l'image source est l'inverse de l'image distance et l'image des marqueurs est l'image des minima obtenue après l'érodé ultime. L'image distance est une carte des

distances entre les pixels des objets d'intérêt de l'image binaire. Le calcul des distances est réalisé avec l'algorithme proposé par Russ (1992), qui se rapproche de vraies distances euclidiennes. L'image des minima s'obtient par des érosions successives de l'image distance jusqu'à ce que l'on atteigne les niveaux ultimes avant leur disparition.

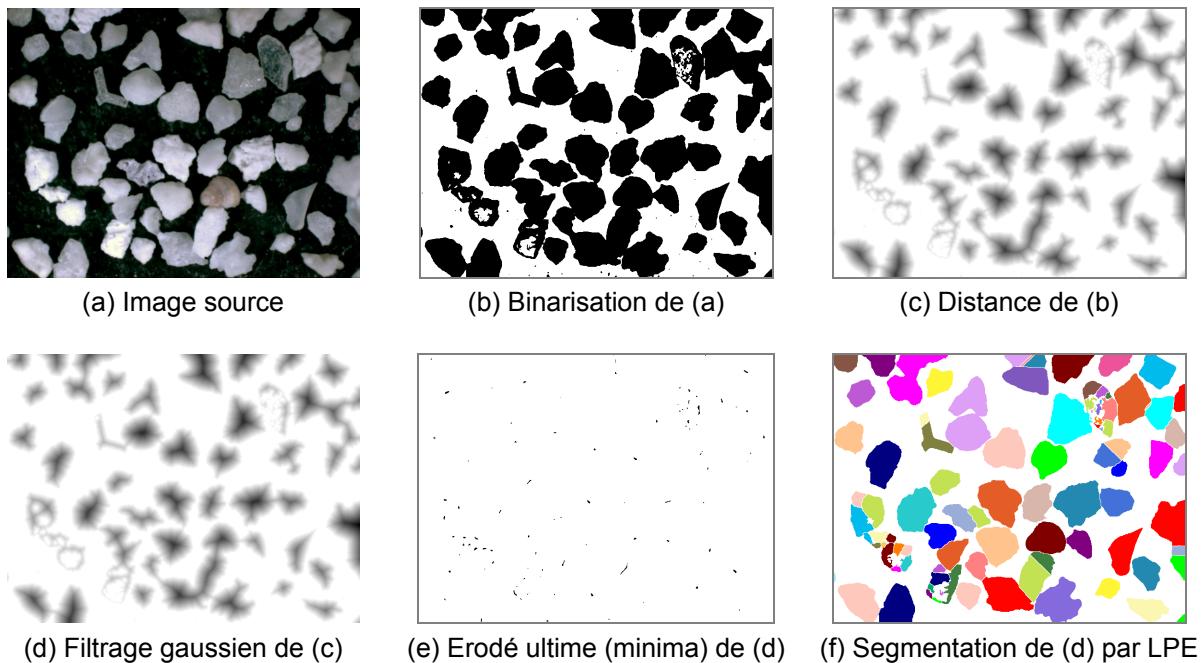


Figure 3.9 – Segmentation d'une image binaire par l'érodé ultime

Malgré un lissage avec un filtre gaussien de l'image distance pour fusionner les minima très proches, il subsiste des lignes de crêtes divisant certains grains de la Figure 3.9 (f). La LPE par l'érodé ultime donne de bon résultats uniquement lorsque les objets ont une forme très ramassée se rapprochant d'une sphère. Une méthode alternative pour la séparation des objets connexes est proposée à la Section 4.5.

# **Chapitre 4 : Caractérisation d'objets 2D**

## **4.1 Introduction**

Une fois l'image segmentée en régions par une des méthodes présentées dans le Chapitre 3, la caractérisation des objets d'intérêt au moyen de descripteurs permet de comparer ou de classer ces objets. En raison de l'immense diversité des formes, il nous paraissait important de pouvoir obtenir un grand nombre de mesures afin de caractériser au mieux les particules ou les composantes d'une image pétrographique.

Dans la littérature, on trouve une multitude de descripteurs, répondant souvent à des besoins particuliers. Seuls les descripteurs implémentés dans JMicrоЩision sont décrits dans ce chapitre. Nous proposons un nouveau descripteur nommé "contexte" à la Section 4.7.5.2 , qui vise à améliorer la classification des types de porosité dans les roches carbonatées (voir Section 7.3).

Dans l'architecture logicielle, un effort particulier a été accompli en ce qui concerne le traitement des images de grande, voire de très grande taille (de l'ordre du giga-octet), afin d'obtenir des performances optimales tant en rapidité de calcul qu'en termes de gestion de la mémoire. L'optimisation la plus importante concernant la gestion de la mémoire est l'adaptation à la technique du tuilage de l'algorithme de croissance de régions, qui est présenté à la Section 4.4. Ce chapitre traite aussi, à la Section 4.5, d'une méthode innovante de post-traitement à la segmentation qui vise à séparer les objets connexes.

## 4.2 Objets, trous et fond

L'individualisation des objets se fait à partir d'une image binaire obtenue selon l'une des méthodes évoquées dans le Chapitre 3. Les objets, les trous et le fond sont des regroupements de pixels connexes de même couleur. La connexité est un élément topologique important dans les images binaires, car elle permet de définir les relations de voisinage entre les pixels (voir Section 3.2.3).

Le théorème de Jordan énonce que toute courbe fermée qui ne se recoupe pas sépare les points du plan en deux domaines connexes, l'intérieur et l'extérieur de la courbe. Cette définition est faite pour un cas continu. Pour un cas discret, comme dans la Figure 4.1 (b), cette affirmation n'est plus valable, car l'objet est censé être fermé diagonalement en haut à droite. Si le fond et les trous sont tous deux 8-connexes, ils sont liés par les sommets, donc l'extérieur et l'intérieur sont connectés! Il est possible de surmonter ce paradoxe en opposant la connexité des pixels des objets et celle des pixels du fond et des trous (Kong & Rosenfeld, 1989). Dans le cadre de ce travail, les pixels des objets sont définis comme 8-connexes et ceux du fond et des trous comme 4-connexes (l'inverse serait aussi possible).

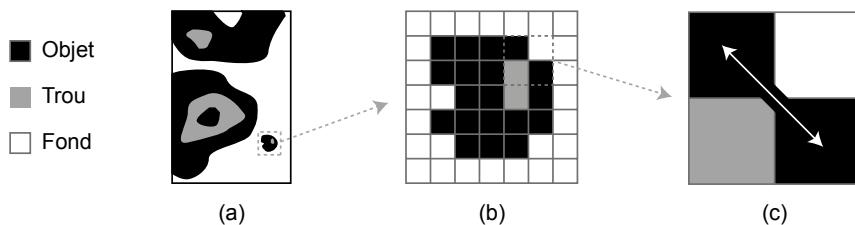


Figure 4.1 – Définition des objets, des trous et du fond: (a) représentation de 4 objets et 3 trous; (b) représentation discrète; (c) opposition de la connexité entre l'objet et le fond

Dans une image binaire, le fond et les trous ont des valeurs de pixels identiques. Néanmoins, la Figure 4.1 les représente par une couleur différente dans le but de faciliter leur distinction. Le fond est défini comme le regroupement de toutes les régions ayant au moins un pixel qui touche le bord de l'image, tandis que les trous n'atteignent jamais un bord. Un objet à l'intérieur d'un trou (voir Figure 4.1 (a)) se différencie de l'objet qui l'englobe selon la notion de connexité donnée dans la définition d'un objet.

## 4.3 Représentation vectorielle des objets

### 4.3.1 Introduction

La représentation d'un objet sous forme de contour facilite certains traitements et est surtout bien moins coûteuse qu'une représentation matricielle en termes de mémoire. Le chaînage par code de Freeman (1961) permet de stocker efficacement la bordure d'un objet d'intérêt et de ses trous. Le principe consiste à lier les pixels formant le contour de l'objet en utilisant une des directions définies par les codes de Freeman (voir Figure 4.2). Un contour 8-connexe a l'avantage de stocker moins de codes que le même contour 4-connexe.

Dans JMicrоЩision, les objets sont stockés sous forme de contours chaînés 8-connexes dans le fichier du projet. La structure de données d'un objet est composée d'un tableau unidimensionnel contenant la suite des codes de Freeman et de la position du point de départ de la chaîne. Les codes indiquent le déplacement relatif de chaque pixel de la bordure par rapport au pixel précédent.

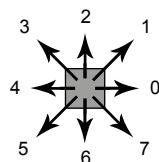


Figure 4.2 – Directions des huit codes de Freeman

Une chaîne est donc constituée d'une suite de pixels qui forme un chemin 8-connexe. Les directions des codes sont comptées dans le sens trigonométrique, et comme il s'agit d'objets à deux dimensions, le contour se ferme toujours.

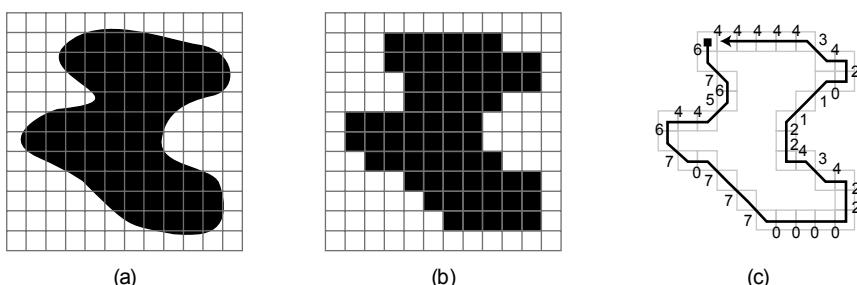


Figure 4.3 – Digitalisation et extraction de la bordure: (a) forme originale; (b) forme discrète; (c) extraction de la bordure selon le codage de Freeman

Il existe d'autres méthodes de stockage efficaces de l'information binaire, comme la compression de type RLC (Run Length Coding). Son principe consiste à compter le nombre de pixel noirs successifs et à transformer cette succession en indiquant le nombre de répétitions et la position sur la ligne. Toutefois, la compression RLC appliquée de manière individuelle aux objets nécessite une méthode qui puisse au préalable les identifier (comme la croissance de régions). Avec le codage de Freeman, les contours des objets peuvent être extraits directement par l'algorithme proposé à la Section 4.3.2. De plus, la représentation sous forme de contour permet de calculer aisément le périmètre (voir Section 4.7.2) et la forme de l'objet peut être modifiée directement par l'information du contour (voir Section 4.6).

### 4.3.2 Algorithme d'extraction de contour

L'algorithme présenté ci-dessous permet, à partir d'une image binaire, de lier le contour 8-connexe de l'objet à l'aide des codes de Freeman (voir Figure 4.2). Il est capable de prendre en charge les pincements ou les filaments de l'objet (voir Figure 4.4). Dans ces zones, deux portions de la bordure se superposent et par conséquent ces positions possèdent deux codes de Freeman.

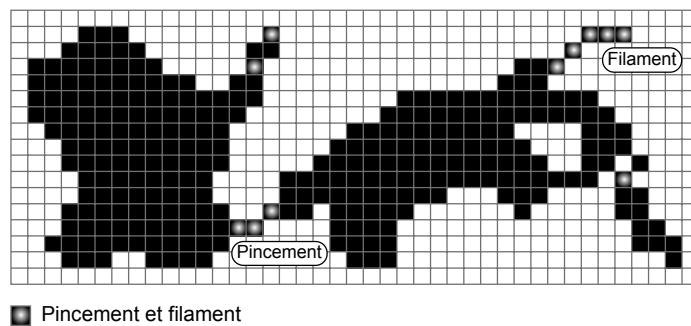


Figure 4.4 – Pincements et filaments d'un objet

Le point de départ de la chaîne est le point le plus en haut à gauche de l'objet. Il est le premier pixel d'un objet à être détecté lorsque l'image binaire est scannée ligne par ligne et de gauche à droite. Ensuite, la recherche des prochains points du contour s'effectue dans le sens inverse des aiguilles d'une montre selon les directions de Freeman.

Dans le pseudo-code ci-dessous, les variables  $x_0$  et  $y_0$  représentent la position de départ et les codes de Freeman sont stockés dans le tableau `Codes`. Les constantes `DIRX` et `DIRY` expriment le déplacement relatif par rapport au pixel courant selon la valeur du code de Freeman. Par exemple, le code 0 représente la position dans les tableaux `DIRX[0]` et `DIRY[0]`, qui expriment un déplacement de 1 pixel vers l'ouest par la coordonnée (1,0).

```

DIRX = [1, 1, 0, -1, -1, -1, 0, 1];
DIRY = [0, -1, -1, -1, 0, 1, 1, 1];
Dernière_direction = 5;
i = x0; j = y0; k = 0;
FAIRE
POUR i = Dernière_direction à i + 6 FAIRE
    Direction = i modulo 8;
    dx = DIRX[Direction];
    dy = DIRY[Direction];
    SI (Image(dx + i, dy + j) == NOIR) ALORS
        k = k + 1;
        Codes[k] = Direction;
        i = i + dx;
        j = j + dy;
        Dernière_direction = (Direction + 6) modulo 8;
    FIN POUR
TANT QUE (i != x0 ET j != y0)

```

### 4.3.3 Cas particulier

Dans l'algorithme proposé à la Section 4.3.2, la condition d'arrêt est satisfaite lorsque la position courante est identique à la position de départ. Cette condition peut s'avérer insuffisante lorsque la position courante correspond à celle de départ sans que le contour de l'objet ait été parcouru entièrement. Ce cas exceptionnel peut survenir à partir des configurations (a), (b) et (c) présentées dans la Figure 4.5. Lorsque l'algorithme est exécuté sur l'objet de l'exemple (d), il parcourt uniquement la partie gauche et s'arrête sur le point de départ sans chaîner la partie droite.

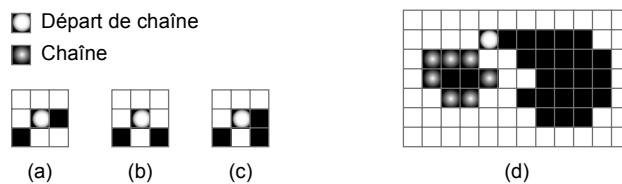


Figure 4.5 – Cas particulier d'extraction de contour

Pour surmonter le problème posé par ce cas particulier, il faut d'abord tester si l'objet correspond à l'un des masques (a), (b) ou (c) de la Figure 4.5. Voici le pseudo-code qui doit être placé tout au début de l'algorithme de la Section 4.3.2:

```
Cas_Particulier = FAUX;
SI (Image(DIRX[5]+i, DIRY[5]+j) == NOIR ET Image(DIRX[6]+i, DIRY[6]+j) == BLANC
    ET (Image(DIRX[0]+i, DIRY[0]+j) == NOIR OU Image(DIRX[7]+i, DIRY[7]+j) ==
        NOIR)) ALORS
    Cas_Particulier = VRAI;
```

Ensute, une deuxième partie de code doit être ajoutée après " $j = j + dy;$ ":

```
SI (Cas_Particulier == VRAI ET (i == DIRX[5] + x0) ET (j == DIRY[5] + y0))
    Passerelle = VRAI;
    CAS Direction DE
        '0' : Prochaine_direction = 6;
        '1' : Prochaine_direction = 7;
        '2' : '3' : Prochaine_direction = 8;
        '5' : Prochaine_direction = 4;
    AUTREMENT
        Prochaine_direction = 8;
        Passerelle = FAUX;
    FIN CAS
    POUR k = Prochaine_direction à 7 FAIRE
        Direction = i modulo 8;
        SI (Image(DIRX[Prochaine_direction] + i, DIRY[Prochaine_direction] + j)
            == NOIR) ALORS
            Passerelle = FAUX;
        FIN POUR
        SI (Passerelle == VRAI) ALORS
            k = k + 1;
            Codes[k] = 0;
            i = i + DIRX[0];
            j = j + DIRY[0];
            Direction = 2;
```

Cette deuxième partie s'exécute chaque fois que la position de l'extraction se trouve au sud-ouest du point départ. Dans ce cas, elle établit, si nécessaire, une passerelle pour continuer le chaînage dans la partie droite de l'objet. La variable `Passerelle` a comme valeur `VRAI` lorsque la bordure passe par le point de départ et qu'elle se prolonge dans la partie droite de l'objet. Dans cette situation, on ajoute un pixel au-dessous du point de départ afin d'éviter la condition d'arrêt. Les directions 4, 6 et 7 ne sont pas traitées car elles sont impossibles selon les trois masques présentées à la Figure 4.5.

On notera pour finir que ce cas particulier peut être évité en rendant les objets 4-connexes ou en supprimant les pixels de liaison (voir Section 4.5.2.1).

## 4.4 Algorithme de croissance de régions adapté au tuilage

La méthode que nous proposons ici adapte l'algorithme de croissance de régions (voir Section 3.2.4) à la technique du tuilage (voir Section 1.2.4) en limitant la croissance à une tuile.

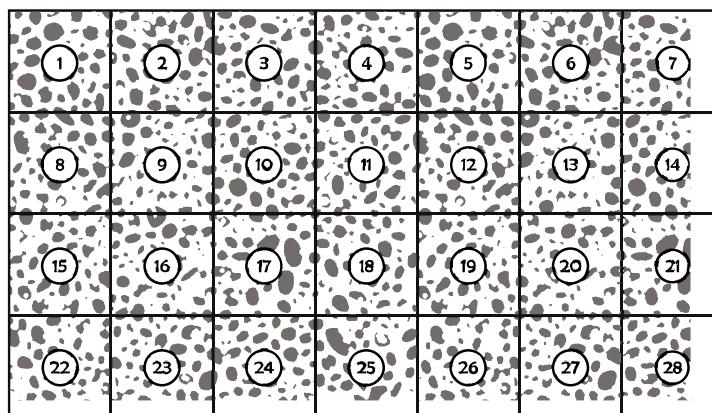


Figure 4.6 – Tuilage d'une image binaire

Le principe général de l'algorithme, que se soit pour l'extraction des objets (voir Section 4.4.1) ou pour les différentes opérations binaires (voir Section 4.4.2), consiste à parcourir l'image tuile après tuile (voir Figure 4.6) et à exécuter la croissance de régions sur chaque tuile. Néanmoins, cette approche conduit inévitablement à la division des objets qui se trouvent à cheval sur plusieurs tuiles. Pour corriger ce problème, l'algorithme cherche à fusionner les objets qui touchent le haut et le côté gauche de la tuile courante avec ceux positionnés sur les tuiles voisines. Pour que l'algorithme s'exécute dans des conditions optimales, la dimension des objets doit être inférieure à celle des tuiles.

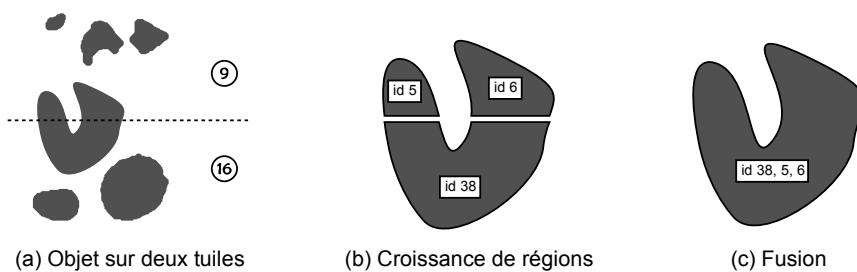


Figure 4.7 – Fusion d'un objet à cheval sur deux tuiles

#### 4.4.1 Extraction des objets

L'algorithme d'extraction des objets sert à vectoriser les régions d'intérêt d'une image binaire. Dans le pseudo-code ci-dessous, le tableau `Germes` est une table de référence où chaque élément ayant une valeur `Vrai` correspond à un pixel doté d'un potentiel de croissance. Lorsque l'algorithme arrive sur un germe (un élément `Vrai`), il extrait le contour de l'objet (voir Section 4.3.2), puis lance la croissance sur cette région (voir Section 3.2.4, où le tableau `visité[]` correspond à `Germes[]`, et l'image labellisée à `Région[]`; le label n'est pas nécessaire car un seul objet est écrit dans `Région[]`), qui est en même temps copiée dans le tableau booléen `Région`. L'objet vectoriel est ensuite créé à condition qu'il ne soit pas en contact avec un bord de la tuile. Cette vectorisation directe évite de créer une image intermédiaire de type entier (32 bits par pixel), qui serait utilisée pour labelliser les régions par des valeurs spécifiques (comme dans l'algorithme présenté à la Section 3.2.4).

```

POUR chaque tuile FAIRE
    POUR chaque pixel FAIRE
        SI (Image_Source(i,j) == Noir) ALORS
            Germes[i,j] = Vrai;
        FIN POUR
        POUR tous les Germes[i,j] == Vrai FAIRE
            Extraction du contour de l'objet;
            Création du tableau Région;
            où la longueur et la largeur du tableau sont calculées lors de
            l'extraction du contour.
            Croissance de la région;
            SI (Région touche un des bords de la tuile) ALORS
                Région = connection(Région);
            SI (Région != NULL)
                Création d'un objet vectoriel;
            FIN POUR
        FIN POUR
    POUR chaque élément de Ossature FAIRE
        Création d'un objet vectoriel à partir de l'élément Objet_coupé;
    FIN POUR

```

Lors de la création d'un objet vectoriel, le contour est stocké et, par ailleurs, les contours de ses trous sont extraits et conservés. Sa structure de données se définit de la façon suivante:

- Classe ID (identificateur permettant de regrouper les objets en catégories)
- Contour de l'objet
  - Point de départ
  - Codes de Freeman
  - Surface (nombre de pixels de l'objet)

- Liste des trous, où chaque trou a son point de départ, ses codes de Freeman et sa surface.

La procédure `connection()` est déclenchée lorsque l'objet courant touche un bord de la tuile.

Son pseudo-code se présente de la façon suivante:

```

connection(Région) {
    SI (Région touche le haut ou le bord gauche de la tuile) ALORS
        POUR chaque élément de Ossature qui est connexe à Région FAIRE
            Région = fusion(Région, Objet_coupé.Région);
    SI (Région touche le bord droit ou le bas de la tuile) ALORS
        Création de Objet_coupé à partir de Région;
        Ossature.ajout(Objet_coupé);
    RETURN NULL;
    RETURN Région;
}

```

Le type `Objet_coupé` est un objet temporaire qui touche le bord droit ou le bas d'une tuile. Sa structure de données est la suivante:

- Région (tableau booléen qui représente l'objet)
- Surface (nombre de pixels de l'objet)
- Rectangle englobant l'objet (donne la position du tableau Région dans l'image)

Le type `Ossature` est un tableau qui représente le cadre des tuiles (ou la découpe de l'image) et qui a pour fonction de stocker les éléments de type `Objet_coupé` selon leur position. Lorsqu'un objet touche le haut ou le côté gauche de la tuile, l'algorithme cherche dans le tableau `Ossature` les éléments 8-connexes qui pourraient fusionner avec lui (voir Figure 4.7).

L'algorithme implémenté dans JMicrоЩision s'avère un peu plus complexe que celui présenté ici, car il intègre des fonctionnalités supplémentaires, comme l'exclusion des objets inférieurs ou supérieurs à une certaine taille, le remplissage des trous, l'extraction limitée à une zone rectangulaire, elliptique ou polygonale, et la suppression des objets touchant les bords de cette zone.

#### 4.4.2 Opérations binaires

La croissance de régions se retrouve aussi dans des opérations binaires comme la reconstruction, le remplissage des trous (hole fill) et la suppression des objets touchant les bords de l'image (border kill). L'algorithme relatif à ces opérations a pour mission de produire

une image binaire tuilée, contrairement à l'extraction des objets où l'objectif était de vectoriser les contours. Un pixel résultant d'une opération binaire incluant la croissance de régions peut dépendre de pixels diffus dans toute l'image source. Pour cette raison, les principes de l'exécution différée (voir Section 1.2.4) ne peuvent être appliqués de la même façon que pour les opérations conventionnelles. Deux étapes supplémentaires sont requises avant l'exécution différée. Elles sont exécutées une seule fois et de façon immédiate. Dans la première étape, commune à toutes les opérations, l'algorithme parcourt l'image tuile après tuile et procède à la croissance de régions en attribuant un identificateur unique à chaque germe. Contrairement à l'algorithme d'extraction des objets, le type `Ossature` est ici composé d'une bordure interne de 1 pixel de large dans laquelle sont stockées des valeurs qui identifient de façon unique chaque objet (voir Figure 4.8 (c)). Les identificateurs connexes sont mémorisés dans un tableau et fusionnés lorsque toutes les tuiles ont été parcourues (voir Figure 4.8 (d)).

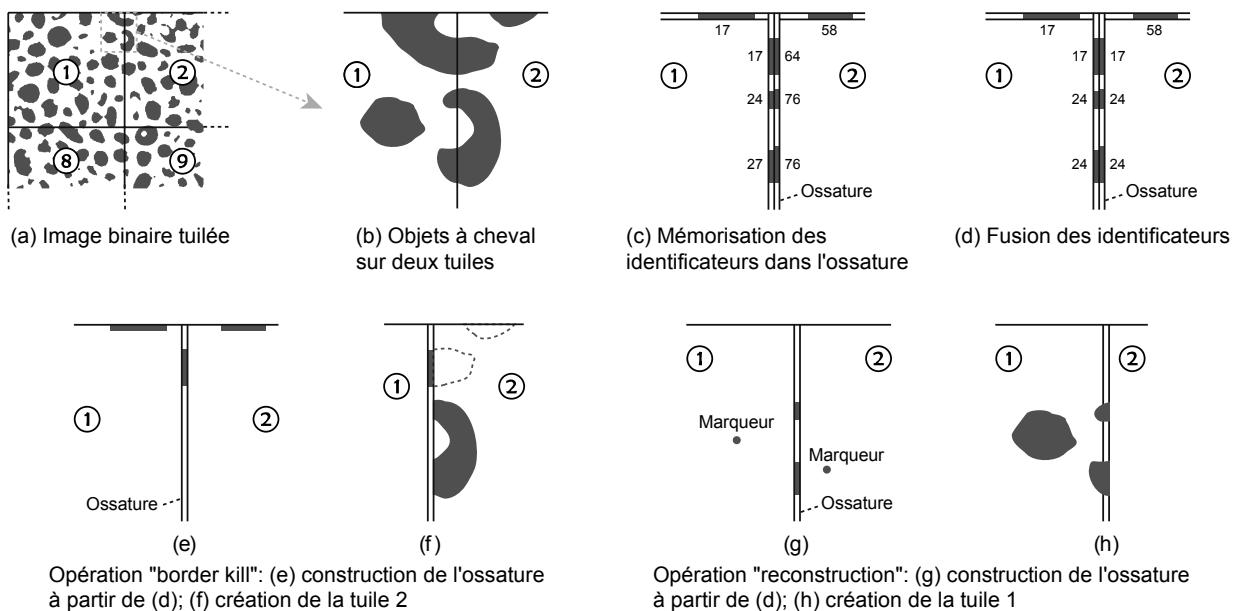


Figure 4.8 – Incription des identificateurs dans l'ossature

La seconde étape consiste à créer une nouvelle ossature de 1 pixel de large passant par le bord droit et le bas de chaque tuile. Ce nouveau type `Ossature` contient une information binaire provenant de la transformation des identificateurs en fonction d'une opération (voir (e) et (g) dans la Figure 4.8). Par exemple, pour l'opération "border kill", seuls les identificateurs qui touchent un bord de l'image sont conservés et marqués comme "vrai" dans la nouvelle ossature; pour l'opération "reconstruction", qui consiste à sélectionner des objets par le biais

de marqueurs, seuls les identificateurs des objets en contact avec un marqueur sont marqués comme "vrai". Cette ossature est ensuite conservée avec l'opération de façon permanente.

Après ces deux étapes, une tuile peut être construite de manière indépendante à l'aide de l'information contenue dans l'ossature (voir (f) et (h) dans la Figure 4.8), tout en respectant les principes de l'exécution différée et en tenant compte du contexte global de l'image.

## 4.5 Séparation d'objets connexes

### 4.5.1 Méthodes de séparation

Les différentes mesures effectuées sur les objets (voir Section 4.7) sont une source d'informations précieuses servant à la caractérisation, à l'identification et à la classification. Malheureusement, il est fréquent que des objets jointifs biaissent ces mesures. Les méthodes habituellement employées pour séparer les objets connexes sont l'érosion-dilatation (Russ, 1992) et la ligne de partage des eaux (voir Section 3.5.2). Cependant dans la pratique, elles se montrent parfois peu efficaces. Aussi des méthodes plus robustes sont-elles recherchées non seulement en géologie, mais également dans d'autres domaines comme la biologie, l'environnement, l'agro-alimentaire ou la médecine.

Dans la Figure 4.9, l'image originale (a) a été seuillée (voir Section 3.3) afin d'obtenir une image binaire (b) sur laquelle les deux méthodes de séparation sont ensuite appliquées. Le résultat final est mis en évidence par des objets de couleurs différentes. Les érosions-dilatations successives (opérations de morphologie mathématique) se montrent particulièrement inadaptées, car elles altèrent la forme des particules et ne se prêtent pas à la séparation d'objets contenant des trous ou ayant une zone de fusion importante. Quant à la ligne de partage des eaux, elle s'avère peu efficace sur les particules qui s'éloignent d'une forme ramassée ou qui contiennent des trous.

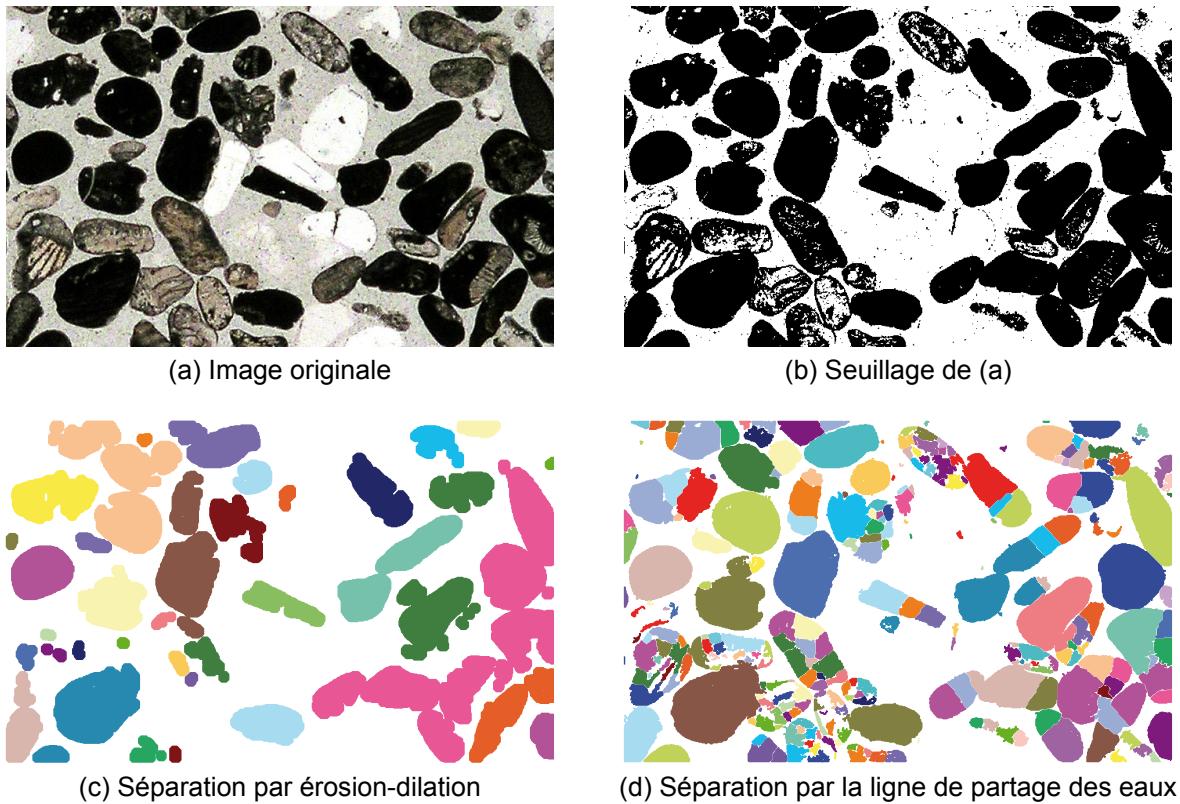


Figure 4.9 – Méthodes classiques de séparation des objets connexes

#### 4.5.2 Algorithme de séparation par recherche des concavités (SRC)

Nous proposons dans cette section un nouvel algorithme de séparation, implémenté dans JMicrоЩision. L'algorithme SRC a l'avantage d'être plus efficace pour la séparation des objets allongés ou contenant des trous.

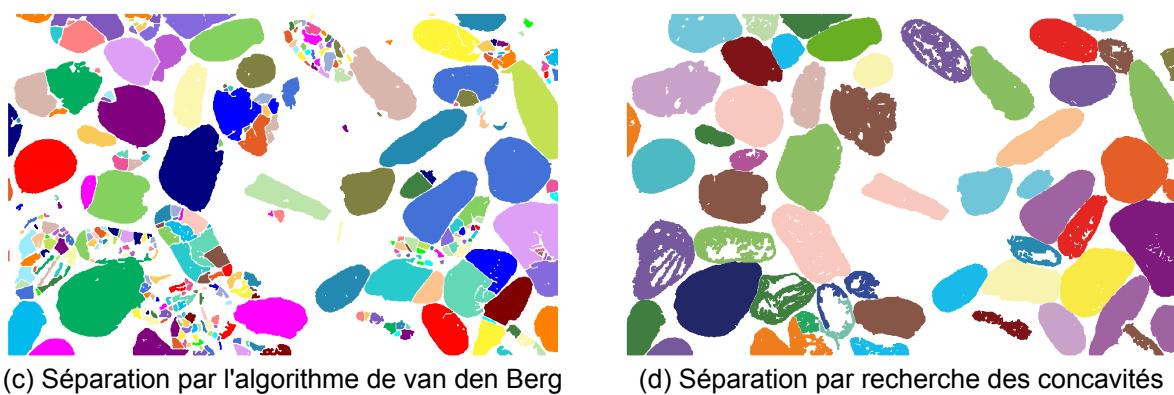


Figure 4.10 – Nouvelles méthodes de séparation des objets connexes

L'idée de base, qui est de découper des régions par le biais d'une ligne reliant deux concavités, apparaît déjà dans la méthode proposée par van den Berg (Van den Berg *et al.*, 2002). Cependant, l'algorithme SRC se distingue par le fait que la séparation est réalisée à partir des angles concaves du contour polygonal, et non, comme dans celui de van den Berg, par la recherche des concavités dans l'image matricielle au moyen d'une fenêtre mobile, avec comme critère de séparation le centre de gravité d'une portion du contour. De plus, il permet de contraindre la découpe afin d'obtenir des objets ayant une forme spécifique. Les six étapes principales de l'algorithme SRC sont décrites dans les sections ci-après.

#### 4.5.2.1 Suppression des pixels de liaison

Cette opération supprime les pincements liant deux portions d'un objet ou les filaments en périphérie. Elle permet, lors de l'approximation polygonale (voir Section 4.5.2.4), d'éviter les croisements de segments qui conduisent dans certains cas à un calcul erroné des angles (un angle convexe peut devenir concave et inversement). Par ailleurs, les pincements ou les filaments sont parfois le résultat du bruit dans l'image, aussi leur suppression est souhaitable.

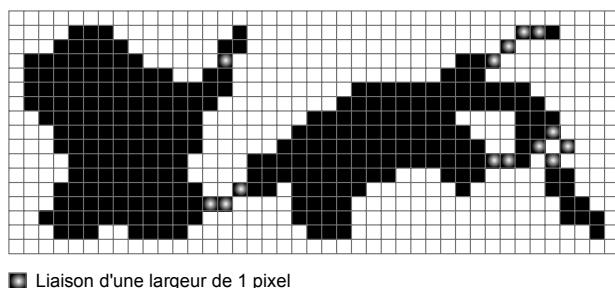


Figure 4.11 – Pixels de liaison

L'algorithme parcourt les pixels de l'image binaire en les comparant aux masques de la Figure 4.12 et, lorsque les deux configurations coïncident, alors la valeur du pixel courant est remplacée par la valeur attribuée au fond. En raison de la configuration des masques, un pixel se trouvant à l'extrémité d'un filament n'est pas effacé. Toutefois, il ne sera pas pris en compte lors de l'extraction des objets car la surface minimale requise est d'au moins 10 pixels.

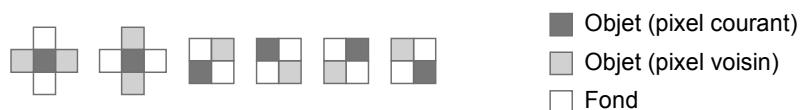


Figure 4.12 – Masques de détection des zones de liaison

#### 4.5.2.2 Différenciation des trous et du fond

Cette étape de l'algorithme SRC vise à différencier le fond des trous (voir définition à la Section 4.2). Elle s'effectue par une croissance de régions (voir Section 3.2.4) où les germes sont les pixels du fond touchant les quatre bords de l'image. La croissance du fond marque les pixels par une nouvelle valeur afin d'obtenir trois types de région: les objets, les trous et le fond. Un trou contenu dans un objet n'est d'aucune utilité pour la séparation, mais son intérêt provient du fait qu'il puisse être un interstice, c'est-à-dire une zone positionnée entre des objets coalescents qui ne peut être atteinte par l'opération de croissance du fond, mais qui néanmoins devrait appartenir au fond.

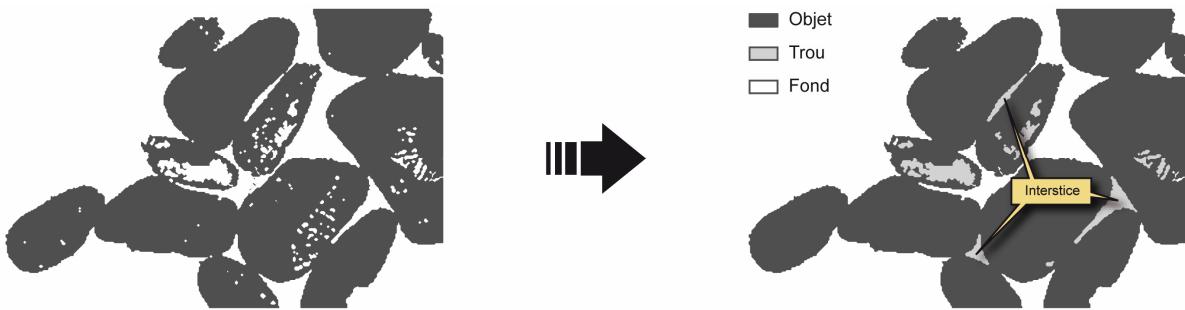


Figure 4.13 – Différenciation des trous et du fond

Dans JMicroVision, l'opérateur fixe la valeur minimale de la surface des interstices pour que tous les trous inférieurs à cette valeur soient considérés comme de vrais trous et, par conséquent, qu'ils puissent être supprimés afin qu'ils n'interfèrent pas lors de la phase de séparation. Par défaut, cette valeur est de 10 pixels et n'est pas autorisée à être inférieure pour que l'approximation polygonale du contour reste réalisable.

#### 4.5.2.3 Extraction du contour

A partir de l'image obtenue par différentiation des trous et du fond, le contour général et le contour des trous sont extraits par l'algorithme proposé à la Section 4.3.2.

#### 4.5.2.4 Approximation polygonale

L'approximation polygonale est réalisée par l'algorithme de la corde présenté à la Section 4.6.3, où l'erreur d'approximation est ici un paramètre défini par l'opérateur. Cette opération permet de simplifier le contour afin de faciliter et d'optimiser la recherche des concavités. Elle peut aussi réduire le bruit sur le contour.

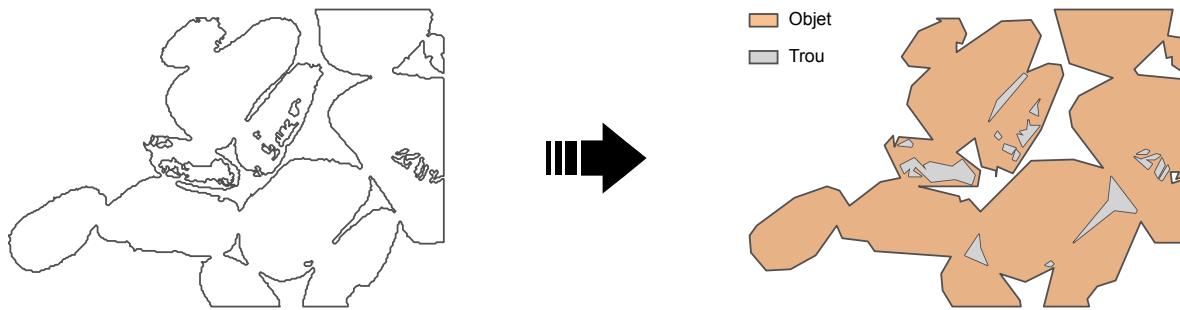


Figure 4.14 – Approximation polygonale des contours

La structure de données d'un contour polygonal est ordonnée en une liste liée, où chaque point connaît le point qui le précède et le point qui le suit.

#### 4.5.2.5 Recherche des points de coupe

Cette opération, qui s'effectue sur le contour polygonal, détermine les zones concaves qui pourraient par la suite être employées dans la découpe des objets connexes. Un point est désigné comme un point de coupe lorsque l'ouverture entre les deux segments d'une zone concave est inférieure à un angle défini par l'opérateur.

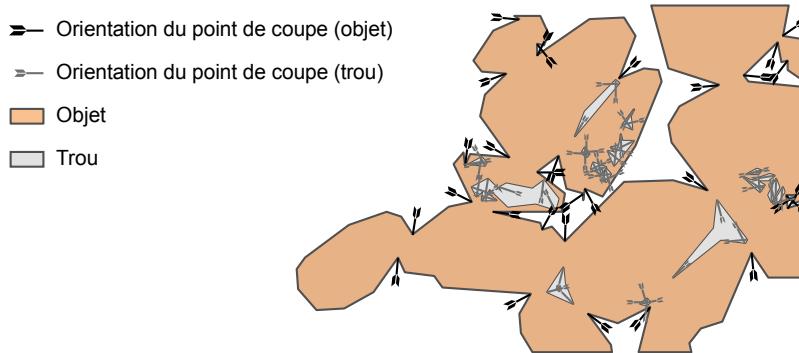


Figure 4.15 – Recherche des points de coupe dans les concavités du contour polygonal

Le point de coupe est lié par sa position au contour polygonal afin d'être facilement accessible lors de la phase de séparation. Sa structure de données est définie par sa position, par l'angle des deux segments qu'il relie et par l'orientation de cet angle.

#### 4.5.2.6 Séparation des particules

La séparation s'effectue à partir des points de coupe sur le contour polygonal, selon des critères de distance, d'orientation, de taille et parfois de forme. En effet, l'opérateur a la possibilité de limiter la séparation à une forme souhaitée, en dessinant cette dernière ou en

sélectionnant plusieurs objets vectoriels représentatifs. Ce critère de forme est défini par deux descripteurs (voir Section 4.7): l'*ellipticité*, qui décrit la forme de l'objet selon sa déviation par rapport à une ellipse, et l'*excentricité*, qui donne une indication sur l'allongement. L'opérateur peut ajuster la tolérance de ces deux descripteurs afin d'adapter au mieux la déviation possible par rapport à la forme type. Contrairement aux critères de forme, la *longueur maximale de coupe*, le *périmètre minimal* et le *périmètre maximal* sont des paramètres indispensables de l'algorithme. Le premier paramètre est transmis directement par l'opérateur. Quant aux deux autres, ils sont calculés à partir de la forme type (dessinée ou la plus proche de la moyenne des critères de forme) et à partir des dimensions minimale et maximale, définies par l'opérateur. Lorsque aucune forme n'est désignée, le calcul se fait à partir d'une ellipse.

Les points ci-après exposent chronologiquement comment l'algorithme SRC procède pour séparer les particules. Les contours polygonaux relatifs à une région 8-connexe sont stockés dans une pile et traités de la façon suivante:

1. Le contour polygonal situé au sommet de la pile est retiré.
2. Si sa longueur est supérieure au *périmètre minimal*, alors il est découpé lors de trois passages successifs sur ce même contour.
3. Sinon on retourne en 1.

#### **1<sup>er</sup> passage:**

1. Le contour polygonal est parcouru dans le sens inverse des aiguilles d'une montre jusqu'à ce qu'un point de coupe (A) soit atteint (voir Figure 4.16, cas (a) et (c)).
2. Un nouveau point de coupe est recherché (selon des critères définis plus bas).
3. Si un point (B) est trouvé et si, de façon réciproque, le point B trouve aussi le point A comme point le plus proche, alors on passe à l'étape 4. Sinon on retourne en 1.
4. Si le point B se trouve sur le contour général (voir cas (c)), alors on coupe le polygone à condition que ce dernier respecte les critères de taille et de forme. Sinon on retourne en 1. Le segment de coupe est inclus au contour des deux nouveaux polygones. Le polygone qui n'a pas été parcouru est remis sur la pile.
5. Par contre, si le point B est localisé dans un trou (voir cas (a)), on se déplace dans le sens des aiguilles d'une montre sur le contour du trou jusqu'au prochain point de coupe

(C), pour lequel un nouveau point de coupe sera recherché (voir cas (b)). Le processus est poursuivi jusqu'à ce que le point trouvé (D) soit situé sur le contour général. Ensuite, les segments de coupe et les portions de segment des trous impliqués dans la séparation sont réattribués aux deux nouveaux contours polygonaux. Le polygone qui n'a pas été parcouru est remis sur la pile.

**2<sup>ème</sup> passage:** la procédure est identique à celle du 1<sup>er</sup> passage, hormis que le critère de réciprocité (énoncé en 3) n'est plus requis.

**3<sup>ème</sup> passage:** le point B recherché n'a plus besoin d'être un point de coupe, mais peut être n'importe quel point du contour polygonal (voir cas (d)). Les critères de recherche restent les mêmes, mis à part que les trous ne sont plus pris en compte et que certaines conditions, énoncées au paragraphe suivant, ne sont plus requises: la deuxième condition en 2 et la condition en 4.

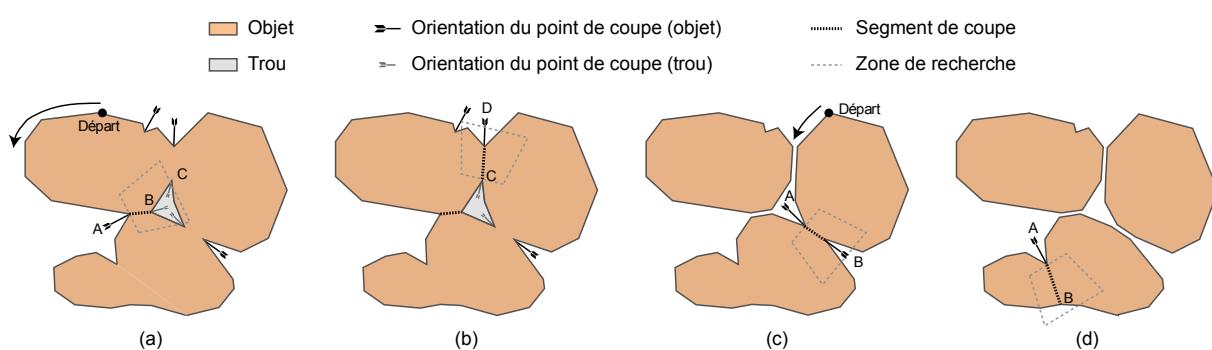


Figure 4.16 – Processus de séparation

La recherche d'un nouveau point de coupe s'effectue dans une zone définie de manière empirique par un trapèze. Ce dernier est orienté et positionné selon le point de coupe (A ou C) à partir duquel la recherche est lancée. La longueur des côtés du trapèze est égale à la *longueur maximale de coupe*. Les côtés forment un angle de 18 degrés par rapport à la hauteur et la base correspond à la longueur d'un côté divisée par deux, plus cinq pixels. Voici les étapes successives qui permettent de trouver un nouveau point de coupe:

1. Le contour polygonal et les contours des trous sont parcourus jusqu'à ce que l'on arrive sur un point de coupe situé à l'intérieur du trapèze. Lorsque tous les contours ont été parcourus, le point de coupe avec la plus courte distance est retourné.

2. Si la distance du point trouvé (B ou D) par rapport au point de départ (A ou C) est inférieure à la *longueur maximale de coupe* et si la différence d'orientation entre les angles des deux points de coupe est inférieure à 82 degrés (soit la tolérance de l'orientation des angles), alors on continue en 3. Sinon on retourne en 1.
3. Si le segment de coupe passe sur des pixels associés au fond ou à un trou, alors on retourne en 1. Cette restriction permet de ne pas aboutir à des situations qui ne respecteraient plus les propriétés topologiques définies à la Section 4.2.
4. Si le segment de coupe forme bien un angle entre 143 et 288 degrés<sup>20</sup> avec le segment précédent et avec le segment suivant, alors sa distance est conservée pour être par la suite comparée (en 1), et on retourne en 1.

Comme les points du contour polygonal sont des points issus de la bordure avant l'approximation polygonale, les segments de coupe (lignes 4-connexes appartenant au fond) peuvent être placés directement dans l'image obtenue à la Section 4.5.2.2 pour produire l'image finale à partir de laquelle les objets peuvent alors être vectorisés correctement.

### 4.5.3 Discussion

L'algorithme SRC donne des résultats satisfaisants lorsque les images contiennent des objets<sup>21</sup> qui ne se superposent pas ou se superposent très peu, qui ont une forme avec peu de concavités, qui n'ont pas de grande différence de taille, et qui peuvent être différenciés du fond. La rapidité d'exécution de l'algorithme est directement liée à la taille du contour des objets connectés, car la recherche des points de coupe sur le contour sera d'autant plus longue que leur nombre est grand.

La technique du tuilage n'a pas été intégrée dans l'algorithme SRC, car celui-ci progresse de façon indéterminée dans l'image. De plus, lorsque les objets sont très interconnectés, ils forment des régions qui peuvent être plus grandes qu'une tuile, ce qui rend le tuilage inapproprié. Pour optimiser l'algorithme, il serait tout de même possible d'exécuter en

---

<sup>20</sup> Ces valeurs sont exprimées en radians dans l'algorithme et converties ici en degrés. Elles représentent la tolérance de la courbure du contour.

<sup>21</sup> Le mot "objet" désigne ici la notion commune et non informatique.

parallèle la découpe des régions, où chaque contour polygonal figurant dans la pile pourrait être lancé dans un processus indépendant. Toutefois, le temps de calcul serait optimal seulement avec des régions de taille comparable.

Plusieurs améliorations de l'algorithme SRC sont envisageables:

- Le gradient de l'image originale pourrait être utilisé comme un critère supplémentaire de séparation et pour le tracé de la découpe finale.
- Au lieu de fixer une valeur constante pour la *longueur maximale de coupe*, il serait plus judicieux de l'adapter en fonction de la taille du polygone en phase de coupe. Cela serait utile lorsque les objets ont des tailles variables.
- Une méthode pourrait être élaborée pour différencier les interstices des trous selon des critères de forme.

## 4.6 Modificateurs de forme

### 4.6.1 Introduction

JMicroVision implémente plusieurs méthodes qui modifient la forme des objets: le lissage, l'approximation polygonale et l'enveloppe convexe. Ces modifications s'effectuent directement sur le contour de l'objet. Toutefois le contour, qui est représenté sous forme de codes de Freeman (voir Section 4.3), nécessite une transformation en coordonnées absolues. L'algorithme de conversion est simple à mettre en œuvre, voici son pseudo-code:

```

DIRX = [1, 1, 0, -1, -1, -1, 0, 1];
DIRY = [0, -1, -1, -1, 0, 1, 1, 1];
i = x0;
j = y0;
Polygone.nouveau_point(i,j);

POUR k = 0 à nombre de codes de Freeman FAIRE
    Direction = Codes_Freeman[k];
    i = i + DIRX[Direction];
    j = j + DIRY[Direction];
    Polygone.nouveau_point(i,j);
FIN POUR

```

### 4.6.2 Lissage par filtrage

La méthode proposée ici consiste à appliquer un filtre sur le contour de l'objet afin d'en émousser la forme. Elle est comparable au filtrage d'un signal à une dimension. Cependant, il est difficile de mettre en œuvre un lissage sans que celui-ci n'affecte aussi la forme générale du fait que le contour est dans un référentiel à deux dimensions. Un filtre de grande dimension aura tendance à réduire considérablement la surface de l'objet, cette diminution étant proportionnelle à la largeur du filtre et inversement proportionnelle à la longueur de la bordure. Pour éviter ce désagrément, il existe différentes techniques. Celle proposée ici a l'avantage d'être simple à implémenter. Elle adapte la taille du filtre par une contrainte empirique. Dans JMicrоЩision, la dimension d'un filtre moyen ou d'un filtre gaussien est réduite de façon qu'elle ne soit jamais supérieure au nombre de pixels formant la bordure divisé par quatre. Cette contrainte permet de mieux préserver la forme générale, plus particulièrement pour les objets de petite taille.

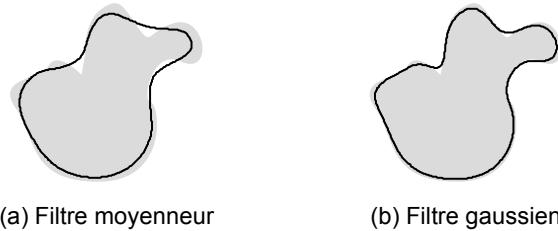


Figure 4.17 – Lissage des objets

La forme générale peut aussi être préservée en transformant la bordure en coordonnées polaires, puis en la filtrant. Néanmoins, cette méthode n'a pas été retenue car elle est bien plus gourmande en temps de calcul, en raison des transformations de coordonnées, et parce qu'elle a tendance à arrondir la forme générale avec des filtres de grande taille.

L'ouverture ou la fermeture en morphologie mathématique sont des méthodes alternatives de lissage qui requièrent une représentation matricielle (sous forme d'image) des objets. Ces opérations peuvent être accomplies dans JMicrоЩision. Toutefois, elles possèdent des propriétés particulières. En effet, l'ouverture peut supprimer des objets de petite taille et la fermeture a tendance à remplir les trous et à fusionner les objets ayant une certaine proximité.

### 4.6.3 Approximation polygonale

L'approximation polygonale transforme le contour de l'objet en une suite de segments de droite. Il existe plusieurs algorithmes de ce type; celui qui a été implémenté dans JMicroVision est basé sur l'algorithme de la corde ou de Ramer (1972). En termes de complexité et d'espace mémoire utilisé, cet algorithme n'est pas le plus efficace, mais il présente l'avantage d'être facile à mettre en œuvre et d'offrir un contrôle précis du seuil d'approximation.

L'algorithme de la corde réduit le nombre de points du contour polygonal par un processus de subdivision par itération. Le segment [AB] représente la corde où A est le point de départ et B le point le plus éloigné de A. Sur le contour entre A et B, on recherche le point M, le plus éloigné de la corde. Si la distance euclidienne entre M et la corde est inférieure au seuil d'approximation, alors on arrête le processus d'itération pour le segment [AB], sinon on continue le processus avec les segments [AM] et [MB].

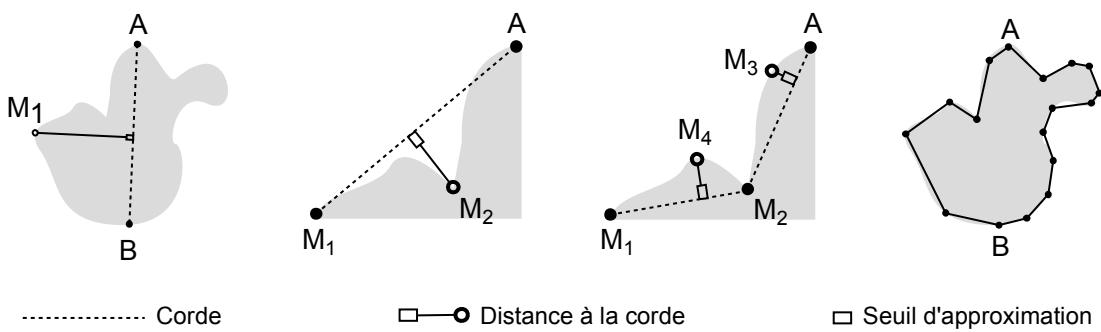


Figure 4.18 – Approximation polygonale: algorithme de la corde

Parmi les autres méthodes d'approximation polygonale, on trouve des algorithmes basés sur la variation de la courbure, qui ont pour inconvénient majeur une forte sensibilité au bruit. Des auteurs comme Lowe (1989) ont proposé des améliorations, avec cependant pour conséquence un alourdissement considérable de la méthode de départ. Une autre méthode est l'approximation au sens des moindres carrés qui, contrairement aux méthodes précédentes, déplace tous les points du polygone, ce qui provoque un effet de lissage sur la forme générale. Elle s'apparente donc plus à une méthode de lissage qu'à une approximation polygonale.

#### 4.6.4 Enveloppe convexe

L'enveloppe convexe d'un objet peut être comparée à un élastique entourant la forme (voir Figure 4.20). Il existe plusieurs approches pour fabriquer un algorithme qui détermine l'enveloppe convexe d'un objet. Celle qui est implémentée dans JMicrоЩision est l'algorithme de Graham (Graham & Yao, 1983). Ce dernier compte parmi les algorithmes les plus rapides, mais il manque de robustesse dans certains cas particuliers. L'algorithme QuickHull (Barber *et al.*, 1996) représente une alternative intéressante, car il semble être plus robuste et d'une complexité quasi équivalente.

Quelques adaptations ont été nécessaires pour rendre l'algorithme de Graham plus robuste. Lorsque les points sont triés par ordre croissant de leur angle par rapport à l'axe des abscisses, certains peuvent avoir le même angle. Il est alors nécessaire d'ajouter un critère de tri correspondant à leur distance croissante. De plus, lorsque le contour de l'objet contient des points colinéaires, l'algorithme peut engendrer parfois des allers et retours entre les points. Pour palier cette instabilité lors de la transformation de la chaîne de Freeman en polygone décrite au début de la Section 4.6, un traitement particulier est requis pour deux situations:

- lorsque plusieurs directions identiques se suivent, seul le dernier point doit être ajouté au polygone.
- lorsque deux directions opposées se suivent, seul le point de la première direction est ajouté au polygone.

### 4.7 Descripteurs

Les descripteurs (ou paramètres) servent à caractériser la forme, la position ou la texture d'un objet (voir définition à la Section 4.2) par une valeur. Cette caractérisation rend possible, dans un second temps, la classification, la comparaison ou l'identification des objets. On peut considérer deux objets comme similaires si leurs descripteurs ont des valeurs semblables. Pour obtenir des valeurs comparables, il faut souvent recourir à des descripteurs possédant des propriétés d'invariance à certaines transformations. Par exemple, lorsque des objets similaires sont orientés différemment, seuls les descripteurs invariants à la rotation fournissent des valeurs comparables. Des propriétés d'invariance à la position, au changement d'échelle ainsi qu'à la déformation affine et de la perspective sont par ailleurs souvent requises pour une

comparaison appropriée. Les moments invariants (voir Section 4.7.3) et les rapports entre plusieurs paramètres (voir Section 4.7.4) contiennent divers critères d'invariance.

Les descripteurs sont généralement mal adaptés aux objets de petite taille regroupant seulement quelques pixels, car ils sont souvent définis pour des formes continues. En géométrie discrète (voir Section 3.2.3), ils sont parfois même impossibles à calculer. C'est la raison pour laquelle, dans JMicrоЩision, les objets inférieurs à dix pixels ne sont pas conservés après l'opération de croissance de régions.

### 4.7.1 Position

Le référentiel utilisé dans JMicrоЩision est orthogonal selon deux dimensions. L'axe des X est orienté à droite et l'axe des Y vers le bas, selon la convention généralement adoptée pour un référentiel d'image. Cependant, l'application offre la possibilité, à tout moment, d'inverser la direction de l'axe des Y et de déplacer l'origine qui se trouve par défaut en haut à gauche de l'image.

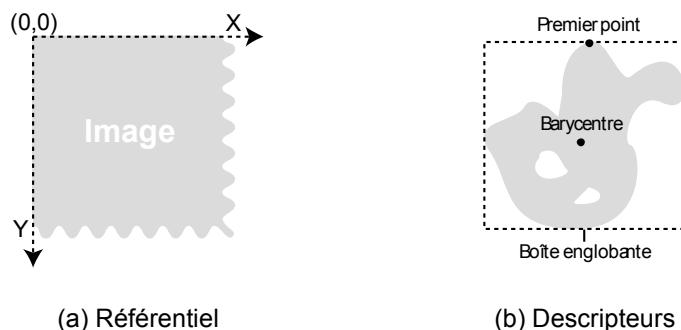


Figure 4.19 – Paramètres de position

**Premier point:** coordonnée du point le plus en haut à gauche de l'objet.

**Boîte englobante (bounding box):** plus petit rectangle entourant l'objet, orienté selon les axes X et Y.

**Barycentre:** coordonnée du centre de gravité de l'objet ( $\bar{x}, \bar{y}$ ) (voir Section 4.7.3). A noter que le barycentre peut être positionné en dehors de l'objet.

### 4.7.2 Descripteurs de base

**Aire:** somme des pixels 8-connexes. Les trous ne sont pas comptabilisés.

**Périmètre:** longueur de la bordure. La bordure des trous n'est pas comptabilisée. Le périmètre se calcule à partir des codes de Freeman (voir Section 4.3) selon la formule empirique définie par Vossepoel (Vossepoel & Smeulders, 1982):

$$\text{Périmètre} = 0.980 * N_p + 1.406 * N_i + 0.091 * N_c \quad (4.1)$$

où  $N_p$  est le nombre de codes pairs dans la chaîne (déplacement parallèle à la grille),  $N_i$  le nombre de codes impairs (déplacement diagonal) et  $N_c$  le nombre de coins (lorsque la chaîne change de direction).

**Aire convexe:** aire de l'enveloppe convexe (voir Section 4.6.4).

**Périmètre convexe:** périmètre de l'enveloppe convexe (voir Section 4.6.4).

**Nombre de trous:** nombre de trous dans un objet (voir Section 4.2).

**Surface des trous:** somme des pixels 4-connexes appartenant aux trous.

**Orientation:** angle de 0-180 degrés, défini selon le sens trigonométrique entre l'axe horizontal et l'axe majeur de l'ellipse calculée à partir des moments d'inertie (voir Section 4.7.3).

$$\text{Orientation} = \frac{180}{2\pi} \tan^{-1} \left[ \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right] \quad (4.2)$$

**Longueur:** diamètre de Feret mesuré parallèlement à l'axe d'inertie défini par l'orientation (voir Section 4.7.5.1).

**Largeur:** diamètre de Feret mesuré perpendiculairement à l'axe d'inertie (voir Section 4.7.5.1).

**Excentricité:** rapport entre l'axe majeur et l'axe mineur de l'ellipse calculée à partir des moments d'inertie (voir Section 4.7.3).

$$\text{Excentricité} = \frac{\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}} \quad (4.3)$$

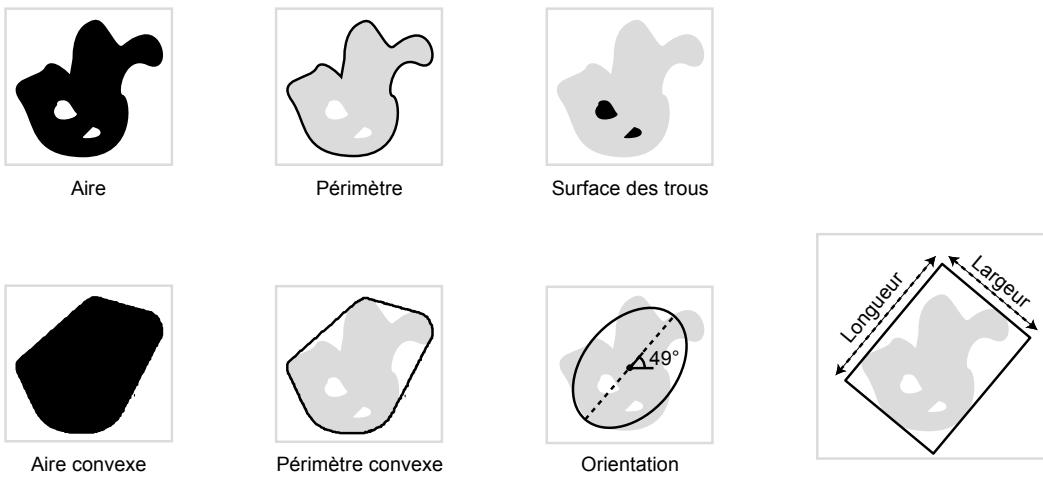


Figure 4.20 – Descripteurs de base

### 4.7.3 Moments

La forme d'un objet peut être caractérisée par divers moments. Les propriétés des moments présentés ici présentent une forte analogie avec ceux utilisés en statistique ou en mécanique. Les moments géométriques discrets sont calculés de la façon suivante:

$$m_{pq} = \sum_{(x,y) \in \text{Objet}} x^p y^q \quad (4.4)$$

où  $x$  et  $y$  sont les coordonnées de tous les pixels appartenant à l'objet,  $p$  définit l'ordre du moment pour les valeurs de  $x$ , et  $q$  celui pour les valeurs de  $y$ . L'ordre d'un moment est égal à  $p + q$ .

La formule du centre de gravité est:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{et} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (4.5)$$

Pour que les moments deviennent invariants à la translation, il est nécessaire de calculer les moments centrés:

$$\mu_{pq} = \sum_{(x,y) \in \text{Objet}} (x - \bar{x})^p (y - \bar{y})^q \quad (4.6)$$

**3 moments centrés:**  $\mu_{11}$ ,  $\mu_{20}$  et  $\mu_{02}$  sont invariants à la translation et décrivent bien l'allongement de formes régulières comme les ellipses. Par contre, ils sont plus ambigus sur des formes complexes.

**7 moments invariants (moments de Hilbert):**  $\phi_1$ -  $\phi_7$  sont invariants à la translation, à la rotation et au changement d'échelle. Ils se construisent à partir des moments normés:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{(p+q+2)/2}} \quad (4.7)$$

Les 7 moments invariants du second et troisième ordre proposés par Hu (1962) sont:

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(3\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}^2(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{12} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (4.8)$$

**4 moments invariants:**  $I_1$ -  $I_4$  sont invariants aux transformations affines<sup>22</sup>. L'invariance algébrique des moments est discutée dans les travaux de Flusser (Flusser & Suk, 1993). Voici les quatre moments qu'il propose:

$$I_1 = \frac{1}{\mu_{00}^4} (\mu_{20}\mu_{02} - \mu_{11}^2)$$

---

<sup>22</sup> Une transformation affine est une transformation qui préserve la colinéarité (un point sur une ligne avant transformation reste sur une ligne après transformation). Les transformations affines les plus communes sont la dilatation, la translation, la réflexion, la rotation et le cisaillement.

$$\begin{aligned}
I_2 &= \frac{1}{\mu_{00}^{10}} (\mu_{30}^2 \mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 + 4\mu_{03}\mu_{21}^3 - 3\mu_{21}^2\mu_{12}^2) \\
I_3 &= \frac{1}{\mu_{00}^7} (\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)) \\
I_4 &= \frac{1}{\mu_{00}^{11}} \left( \begin{array}{l} \mu_{20}^3 \mu_{03}^2 - 6\mu_{20}^2 \mu_{11} \mu_{12} \mu_{03} - 6\mu_{20}^2 \mu_{02} \mu_{21} \mu_{03} + 9\mu_{20}^2 \mu_{02} \mu_{12}^2 \\ + 12\mu_{20} \mu_{11}^2 \mu_{21} \mu_{03} + 6\mu_{20} \mu_{11} \mu_{02} \mu_{30} \mu_{03} - 18\mu_{20} \mu_{11} \mu_{02} \mu_{21} \mu_{12} \\ - 8\mu_{11}^3 \mu_{30} \mu_{03} - 6\mu_{20} \mu_{02}^2 \mu_{30} \mu_{12} + 9\mu_{20} \mu_{02}^2 \mu_{21}^2 + 12\mu_{11}^2 \mu_{02} \mu_{30} \mu_{12} \\ - 6\mu_{11} \mu_{02}^2 \mu_{30} \mu_{21} + \mu_{02}^3 \mu_{30}^2 \end{array} \right) \tag{4.9}
\end{aligned}$$

#### 4.7.4 Ratios

Les ratios sont des rapports entre plusieurs descripteurs. Tous ceux qui sont présentés ci-après renferment des propriétés d'invariance.

**Compacité:** facteur de forme indiquant le degré de compacité de l'objet. L'aire et le périmètre sont deux paramètres invariants à la translation et à la rotation. La division de l'aire par le carré du périmètre annule l'unité de mesure et, par conséquent, on obtient une invariance au changement d'échelle. Le  $4\pi$  permet de normaliser la valeur de compacité à 1 pour une sphère.

$$\text{Compacité} = \frac{4\pi * \text{Aire}}{(\text{Périmètre})^2} \tag{4.10}$$

**Diamètre équivalent sphérique:** diamètre d'un cercle ayant une surface identique à celle de l'objet.

$$D = \sqrt{\frac{4 * \text{Aire}}{\pi}} \tag{4.11}$$

**Elongation:** paramètre d'allongement. L'elongation peut aussi s'exprimer par le rapport longueur sur largeur. Cependant, la formule ci-dessous présente l'avantage de donner des valeurs comprises entre 0 et 1.

$$\text{Elongation} = \frac{\text{Largeur}}{\text{Longueur}} \quad (4.12)$$

**Ellipticité:** différence entre la surface d'une ellipse (ayant comme axes la longueur et la largeur) et la surface de l'objet. La division par l'aire permet de normaliser la valeur de l'ellipticité afin de la rendre invariante au changement d'échelle.

$$\text{Ellipticité} = \frac{\frac{\pi * \text{Longueur} * \text{Largeur}}{4} - \text{Aire}}{\text{Aire}} \quad (4.13)$$

**Rectangularité:** différence entre la surface du plus petit rectangle entourant l'objet et la surface de l'objet. La division par l'aire permet de normaliser la valeur de la rectangularité afin de la rendre invariante au changement d'échelle.

$$\text{Rectangularité} = \frac{\text{Longueur} * \text{Largeur} - \text{Aire}}{\text{Aire}} \quad (4.14)$$

**Solidité:** rapport entre la surface de l'objet et la surface de l'enveloppe convexe. La solidité indique le degré de déviation de la forme par rapport à la forme convexe.

$$\text{Solidité} = \frac{\text{Aire}}{\text{Aire}_{\text{convexe}}} \quad (4.15)$$

**Convexité:** rapport entre le périmètre convexe et le périmètre de l'objet. La convexité indique le degré de déviation du périmètre par rapport au périmètre convexe.

$$\text{Convexité} = \frac{\text{Périmètre}_{\text{convexe}}}{\text{Périmètre}} \quad (4.16)$$

### 4.7.5 Descripteurs additionnels

La distinction faite entre les descripteurs additionnels et les autres provient du fait que les premiers nécessitent l'intervention de l'opérateur pour ajuster certains paramètres.

#### 4.7.5.1 Diamètres de Feret

Un diamètre de Feret est la longueur de la projection de la forme selon une direction. Il peut être comparé à une mesure effectuée avec un pied à coulisse comme illustré dans la Figure 4.21. Dans JMicrоЩision, l'opérateur choisit un pas qui détermine la suite arithmétique des diamètres à mesurer entre 0 et 180 degrés. Par exemple, avec un pas de 10 degrés : 0, 10, 20, 30, ... 160, 170 sont les directions, comptées dans le sens trigonométrique, selon lesquelles les diamètres sont mesurés.

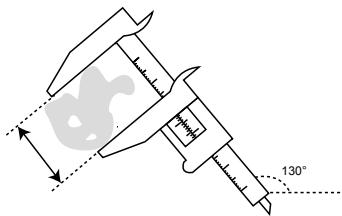


Figure 4.21 – Mesure d'un diamètre de Feret à 130 degrés

L'algorithme des diamètres de Feret opère une rotation de l'objet de façon à positionner le diamètre sur l'axe horizontal. Ensuite, la longueur du rectangle entourant la forme (boîte englobante, voir Section 4.7.1) est calculée.

#### 4.7.5.2 Contexte

Nous proposons ici un nouveau descripteur permettant de caractériser l'environnement proche d'un objet par la mesure de la présence d'un certain type de régions. La quantité de ces régions à proximité de l'objet se calcule en comptabilisant le nombre de pixels inclus dans une bordure positionnée à l'extérieur de l'objet (voir Figure 4.22).

L'intervention de l'opérateur est requise pour choisir plusieurs paramètres:

5. le type de régions, qui regroupe un ensemble de régions issues d'une segmentation (image binaire).
6. le diamètre de la bordure externe, qui définit l'environnement proche à partir duquel est calculée la présence d'un type de régions.

7. la méthode de pondération des niveaux de la bordure extérieure:
- identique pour tous les niveaux de la bordure.
  - pondération croissante des niveaux intérieurs vers les niveaux extérieurs.
  - pondération décroissante des niveaux intérieurs vers les niveaux extérieurs.

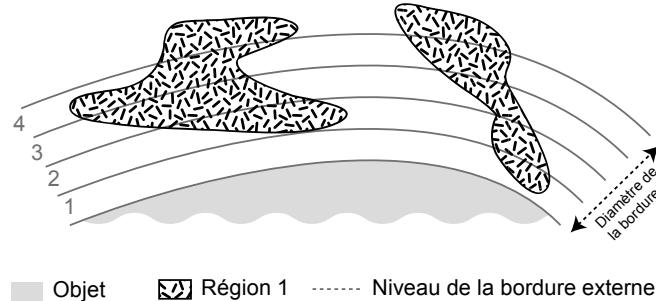


Figure 4.22 – Principe de la mesure contexte

L'algorithme réalise des dilatations successives de 1 pixel à partir de la bordure d'un objet et comptabilise le nombre de pixels trouvés dans la zone dilatée (niveaux numérotés de 1 à 4 dans la Figure 4.22) appartenant au type de régions choisi. Pour rendre compte de la proximité de la surface des régions par rapport à l'objet, une pondération croissante ou décroissante peut être appliquée sur les surfaces obtenues par les différents niveaux.

#### 4.7.5.3 Texture

Le concept de la texture semble évident pour l'observateur, mais n'a pas de définition précise en analyse d'image. Les définitions mathématiques n'arrivent pas à englober la diversité des cas. Le domaine des textures est très vaste, seule l'approche statistique est abordée ici.

Une texture peut être caractérisée de façon simple par les moments statistiques. Dans JMicrоЩision, la texture d'un objet est calculée à partir des valeurs d'intensité des pixels qui proviennent d'une image choisie par l'opérateur et qui sont inclus dans la surface de l'objet. Pour une image couleur, le calcul se fait sur les trois bandes RVB ou TLS (voir Section 3.2.1.3), où sont déterminés pour chaque bande:

- la moyenne ( $\bar{m}$ ) des intensités de pixels appartenant à la région.
- la déviation standard ( $\sigma = \sqrt{\mu_2}$ ), qui mesure la répartition des intensités autour de la valeur moyenne.

- l'asymétrie (skewness) ( $\frac{\mu_3}{\sigma^3}$ ), qui mesure la déviation de la distribution des intensités par rapport à une distribution symétrique. Une faible symétrie de l'histogramme peut donner une valeur négative.
- L'aplatissement (kurtosis) ( $\frac{\mu_4}{\sigma^4}$ ), qui caractérise la forme de l'histogramme. Plus la valeur est faible, plus le sommet de l'histogramme est arrondi.

Les moments centrés d'ordre  $k$  s'expriment de la façon suivante:

$$\mu_k = \frac{1}{N} \sum_{i,j} (g(i,j) - \bar{m})^k \quad (4.17)$$

où  $g(i,j)$  représente la valeur d'intensité du pixel à la position (i,j),  $N$  le nombre total de pixels, et  $\bar{m}$  la moyenne:

$$\bar{m} = \frac{1}{N} \sum_{i,j} g(i,j) \quad (4.18)$$

Le rapport "signal-sur-bruit" (SNR, signal to noise ratio) renseigne sur l'hétérogénéité de la texture. Il se construit à partir de la moyenne et de la déviation standard de la façon suivante:

$$SNR = \frac{\bar{m}}{\sqrt{\mu_2}} \quad (4.19)$$

Un autre descripteur statistique proposé dans JMicroVision est l'entropie. Sa valeur est maximale lorsque l'histogramme est plat, c'est-à-dire que les différents niveaux d'intensité ont le même nombre de pixels. L'entropie provient de la théorie de l'information (Shannon, 1948) et s'exprime de la manière suivante:

$$E = - \sum_n p(n) \cdot \log p(n) \quad (4.20)$$

où  $p(n)$  est la probabilité empirique d'un niveau d'intensité  $n$ .

#### 4.7.6 Discussion

L'aire et le périmètre méritent d'être examinés plus en détail, car ils représentent les mesures les plus fréquemment utilisées et plusieurs descripteurs en dépendent directement. D'une façon générale, les descripteurs sont définis par des méthodes issues de la géométrie continue. Le passage à une géométrie discrète engendre une perte de précision liée à la digitalisation de l'image. Plus la résolution de l'image est basse (ou l'intervalle d'échantillonnage est grand), plus l'erreur aura tendance à s'accentuer. Pour l'aire, les conséquences de la digitalisation et des transformations géométriques sont minimes; par exemple, la surface d'un objet qui subit une rotation peut changer légèrement. En revanche, les conséquences de la discréétisation du périmètre sont beaucoup plus importantes, car l'erreur entre le périmètre original et le périmètre digitalisé peut atteindre plus de 40 %.

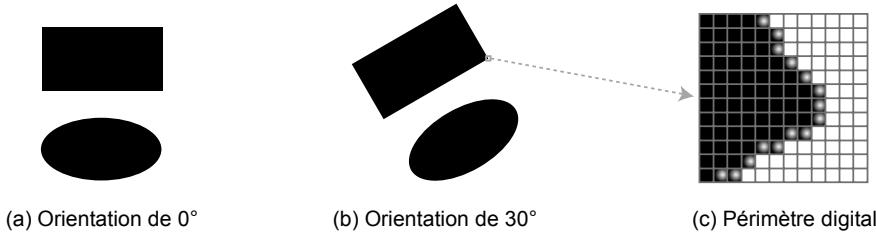


Figure 4.23 – Variation de l'aire et du périmètre selon l'orientation

Dans la Figure 4.23, l'orientation à 30 degrés d'un rectangle sur une grille orthogonale ne permet plus de représenter les côtés de façon rectiligne. En dehors des orientations selon les directions de la grille, la représentation d'une droite doit s'effectuer en marches d'escalier. Cette représentation introduit un biais dans la mesure du périmètre.

Plusieurs auteurs (Koplowitz & Bruckstein, 1989; Yang *et al.*, 1994) ont comparé des méthodes de mesure du périmètre en termes d'efficacité et de précision. Dans le présent travail, notre choix s'est porté sur la formule empirique proposée par Vossepoel (Vossepoel & Smeulders, 1982), car elle s'applique directement aux codes de Freeman et elle tend à minimiser l'erreur introduite par le contexte discret. La Figure 4.24 compare trois méthodes de mesure du périmètre basées sur les codes de Freeman par rapport au périmètre réel (géométrie continue). Il existe cependant d'autres approches pour mesurer le périmètre, comme l'approximation polygonale ou par les courbes de Bézier. Malheureusement, ces méthodes s'avèrent efficaces uniquement avec des formes simples.

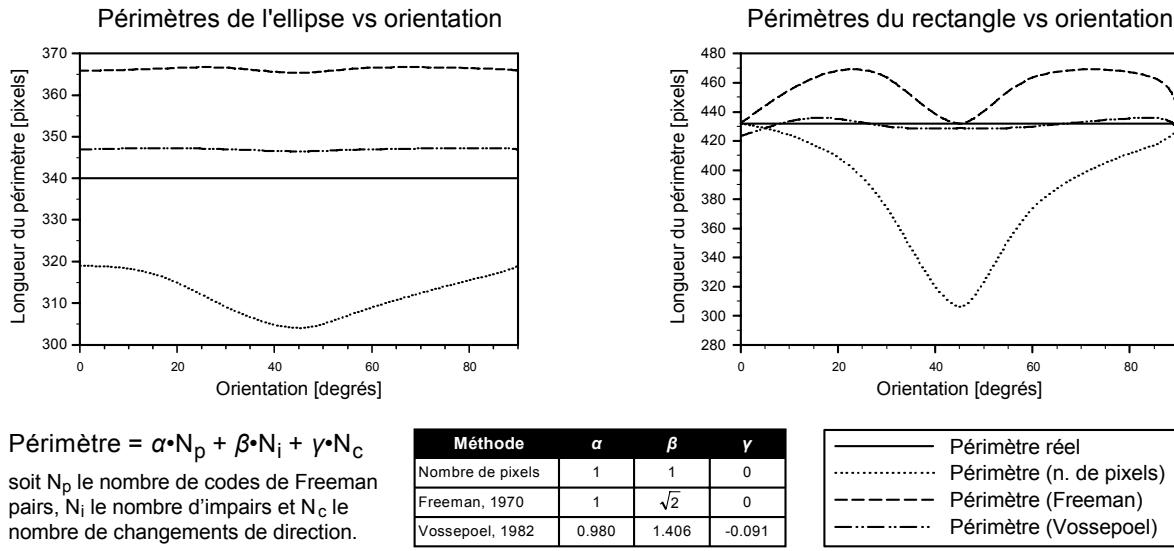


Figure 4.24 – Variation du périmètre selon l'orientation et la technique utilisée

La compacité est le paramètre de forme le plus communément utilisé. Toutefois, il manque de robustesse dans certains cas, comme dans la Figure 4.25, où il donne une valeur identique pour les trois objets, pourtant très différents. Il est peu sensible aux notions de rugosité et d'allongement. Pour révéler les caractères discriminants des formes, il est donc préférable d'utiliser la compacité en combinaison avec d'autres descripteurs.

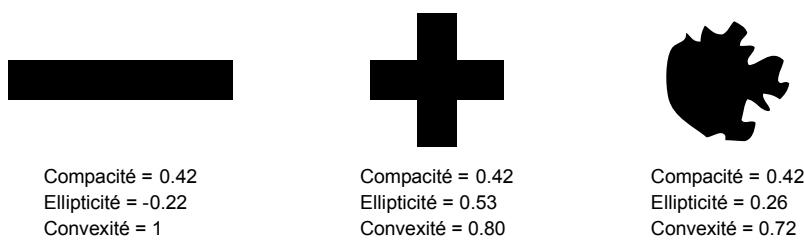


Figure 4.25 – Descripteurs de forme

La précision des descripteurs est directement liée au langage de programmation et plus particulièrement aux types de variable utilisés. Toutes les valeurs des descripteurs et les calculs intermédiaires sont stockés dans des variables réelles à double précision (type double codé sur 64 bits avec une étendue de 4.9406545841246544 E-324 à 1.79769313486231570 E+308). Ce type de variable a été choisi parce qu'il offre une étendue et une précision suffisantes pour la grande majorité des calculs, contrairement aux variables réelles de simple précision (type float codé sur 32 bits) où les résultats peuvent parfois contenir des erreurs non

négligeables suite à des répercussions d'arrondis (erreurs néanmoins plus faibles que celles engendrées par la digitalisation).

D'autres descripteurs qui n'ont pas été implémentés dans JMicroVision présentent un intérêt pour la caractérisation des objets, comme ceux de Fourier-Mellin, ceux basés sur la signature (signal 1D de la forme), ceux adaptés à des formes en fibre, les descripteurs texturaux tenant compte du voisinage et les moments invariants de Zernike.

## 4.8 Classification

Dans JMicroVision, chaque objet vectoriel a pour attribut une classe. Cette dernière, assignée par l'opérateur lors de l'extraction des objets (voir Section 4.4.1), est définie par un nom et une couleur. Par la suite, la classe attribuée à un objet peut être modifiée selon quatre méthodes. La première méthode consiste simplement pour l'opérateur à choisir une classe qu'il assigne en cliquant les objets un à un. Dans la seconde méthode, l'opérateur dessine une zone polygonale et attribue une classe à tous les objets se trouvant à l'intérieur de cette zone. Dans la troisième méthode, les objets sont classés d'après leurs caractères distinctifs en fixant un seuil minimal et un seuil maximal sur l'étendue des valeurs d'un descripteur. Ainsi, les objets les plus arrondis peuvent être associés à une classe, en choisissant le paramètre compacité (voir Section 4.7.4) et en fixant le seuil min à 0,85 et le seuil max à 1. La dernière méthode est celle des plus proches voisins, qui permet une classification supervisée à partir d'éléments d'apprentissage définis par l'opérateur.

La méthode des plus proches voisins se base sur des notions de proximité dans un espace des caractéristiques où, dans notre contexte, chaque dimension est représentée par un descripteur. Cet hyperespace est orthonormé, c'est-à-dire construit selon l'hypothèse d'une indépendance statistique entre les descripteurs (qui ne se vérifie souvent pas dans la réalité). La méthode comprend deux étapes: l'apprentissage, puis la classification (Roduit, 2001). Au cours de la première étape, l'opérateur doit choisir un certain nombre d'objets représentatifs des différentes classes en jeu. La deuxième étape l'amène à sélectionner les descripteurs les plus discriminants afin de construire l'espace décisionnel à partir duquel de nouveaux objets

pourront être classés. Le critère de classification est la plus petite distance entre un nouvel élément et les centroïdes de classe, où la distance s'exprime de la façon suivante:

$$d_z = \sqrt{\sum_{i=1}^n w_i (x_i - \bar{y}_i^z)^2} \quad (4.21)$$

où  $d_z$  est la distance euclidienne d'un élément  $x$  au centroïde d'une classe  $z$ ,  $n$  le nombre de descripteurs,  $w_i$  la pondération spécifique à un descripteur  $i$ ,  $x_i$  la valeur du descripteur  $i$  d'un objet à classer, et  $\bar{y}_i^z$  le centroïde (soit la moyenne des valeurs, spécifique à un descripteur  $i$ , des éléments d'apprentissage  $y$ , appartenant à une classe  $z$ ).

La pondération est définie par une valeur entre 0 et 1, qui traduit l'importance accordée à un descripteur. Dans JMicroVision, l'opérateur peut ajuster les pondérations, qui, par défaut, ont toutes une valeur de 1. C'est aussi à lui qu'il appartient de choisir les descripteurs qui seront les plus discriminants. En principe, ces derniers sont déterminés de façon empirique au vu du résultat de la classification.

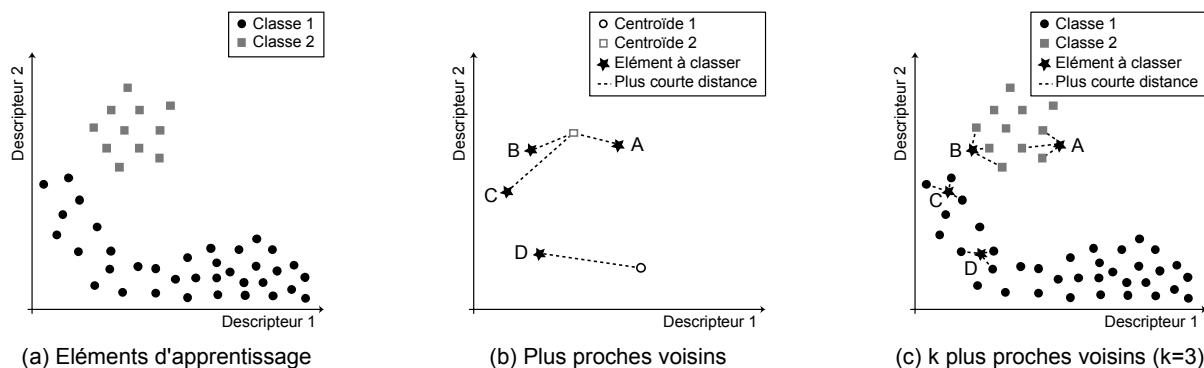


Figure 4.26 – Classification par les plus proches voisins

Une variante de la méthode des plus proches voisins est celle des  $k$  plus proches voisins (KNN: k-nearest neighbor), où la décision ne se fait plus sur la plus petite distance au centroïde, mais sur la proximité d'un nombre  $k$  d'éléments d'apprentissage. Cette variante se montre plus robuste car elle tient compte de la forme du nuage des éléments d'apprentissage. Dans la Figure 4.26, elle classe correctement l'objet C, contrairement à la méthode des plus proches voisins qui l'attribue à la classe 2. Toutefois, son exécution est plus lente car, pour

chaque décision, la distance avec tous les éléments d'apprentissage doit être calculée. Un autre inconvénient provient du fait que l'opérateur doit choisir de façon empirique la valeur de k.

## **II. Exemples d'applications en pétrographie**

### **Introduction**

La quantification et la caractérisation des composants pétrographiques renseignent sur les propriétés physiques ou chimiques ainsi que sur la texture des roches. Ces paramètres sont fondamentaux pour la détermination des roches et pour la compréhension de leur genèse. JMicroVision fournit des outils d'analyse quantitative pétrographique pouvant être utilisés tant dans la recherche académique que dans l'industrie (géotechnique, industrie pétrolière et minière).

Dans cette deuxième partie, plusieurs outils informatiques de mesure et de quantification sont présentés au travers d'une série d'exemples. Le Chapitre 5 traite des aspects généraux

concernant ces outils, tandis que les deux chapitres suivants sont consacrés à des méthodes d'analyse spécifiques à la granulométrie et à la porosité.

Différentes précautions doivent être prises de la préparation de l'échantillon jusqu'à l'interprétation des valeurs produites par JMicrоЩision afin de garantir des résultats aussi exacts et précis que possible. Selon Francus (2004), l'approche quantitative en analyse d'image peut être entachée de trois types d'erreurs:

- a) les erreurs d'acquisition (sursaturation, ombrage...).
- b) les erreurs d'échantillonnage (la résolution d'acquisition doit être adaptée aux objets à mesurer).
- c) les erreurs d'analyse (choix inapproprié de paramètres lors d'une opération, algorithme manquant de robustesse (voir 4.7.6)).

L'abondance des éléments pétrographiques en termes de volume (ou la composition modale) est estimée habituellement à partir de chartes visuelles (voir Section 5.1) ou à l'aide d'un "compteur de points" sous le microscope (voir Section 5.2). Elle peut aussi être obtenue de manière informatique (voir Section 5.2 et 5.3). Quelle que soit la façon de la traiter, la composition modale (y compris la porosité) reste une notion en trois dimensions, alors que toutes les méthodes énoncées précédemment collectent l'information uniquement sur la surface d'un échantillon. Pour que les résultats soient représentatifs de l'échantillon, il faut que celui-ci présente une certaine homogénéité dans le sens perpendiculaire à la coupe, ou qu'une moyenne de plusieurs sections soit calculée.

Que ce soit en utilisant les méthodes traditionnelles ou par analyse d'image, les proportions des composants perçus peuvent parfois être biaisées par l'observation des lames minces en lumière transmise (voir Figure 2.2). En effet, comme le faisceau lumineux traverse l'épaisseur de la lamelle de roche, les éléments observés sont la projection des éléments et non une vraie section plane de l'échantillon. De ce fait, certains éléments ont tendance à être surestimés lorsque leurs bords sont courbes, irréguliers ou orientés dans un sens autre que celui de l'épaisseur de la lame mince (voir Figure 2.3). Ce phénomène, aussi connu sous le nom d'effet de Holmes (1921), peut néanmoins être négligé pour des éléments subsphériques ayant un diamètre supérieur à l'épaisseur de la lamelle de roche. Lorsque les diamètres sont inférieurs,

un facteur de correction peut permettre de rectifier les diamètres obtenus (Weibel, 1979). Toutefois, il ne peut être calculé que si l'on connaît l'épaisseur de la lame et les diamètres des objets. L'effet de Holmes est discuté aussi dans d'autres disciplines comme la biologie (Mandarim-de-Lacerda, 2003). En minéralogie, il a tendance à élargir les grains ayant un fort indice de réfraction (Chayes, 1956). Ainsi, la proportion des minéraux mafiques (FeMg), qui ont généralement un indice élevé, aura tendance à être supérieure à la réalité. Pour éviter cet effet indésirable, il faut employer une méthode d'acquisition collectant l'information dans une épaisseur qui tend à être infiniment petite. C'est le cas pour les systèmes d'acquisition en lumière réfléchie et pour la microscopie électronique à balayage (MEB) utilisant les électrons rétrodiffusés (voir Section 2.1.5). Dans la majorité des situations, l'effet de Holmes reste toutefois une erreur acceptable.

Dans le processus d'analyse d'image, l'étape la plus difficile est sans nul doute la séparation des composants pétrographiques d'intérêt. Différentes techniques peuvent contribuer à une meilleure différenciation par des procédés de coloration de certaines phases minérales (Bailey & Stevens, 1960; Dickson, 1966; Hitzman, 1999; Houghton, 1980; Laniz *et al.*, 1964) ou par l'imprégnation de l'échantillon avec une résine colorée ou fluorescente (Gies, 1987) pour mettre en évidence la porosité. D'autres méthodes comme la combinaison de plusieurs images acquises selon différents types d'éclairage ou l'acquisition multispectrale en appliquant des filtres (Pirard, 2004) sont également utiles pour distinguer les différentes phases minérales.



# Chapitre 5 : Mesures et quantification

## 5.1 Introduction

Depuis des décennies, l'estimation visuelle (Folk, 1951; Terry & Chilingar, 1955) à partir de chartes proposant des valeurs soit quantitatives soit qualitatives (voir Figure 5.2) a souvent été utilisée pour caractériser les roches. Cette approche peut être considérée comme approximative en raison du caractère non continu des valeurs de référence et parce que, de façon systématique, l'œil humain a tendance à surestimer les composants les plus lumineux ou les plus colorés ainsi que ceux qui ressortent le plus (voir Figure 5.1). Contrairement à la perception humaine, qui se laisse facilement influencer par le contexte et qui varie dans le temps et selon l'individu, l'analyse d'image se présente comme une méthode quantitative plus rigoureuse.



Figure 5.1 – Illusions d'optique

Dans certains cas, lorsque les méthodes de segmentation (voir Chapitre 3) ne sont pas suffisamment robustes pour distinguer les différents composants pétrographiques, la composition modale d'une roche peut être déterminée par le comptage de points informatisé (voir Section 5.2). Cette méthode quantitative exige une intervention plus active de l'opérateur que les méthodes automatisées (voir Section 5.3), mais permet de déterminer la composition d'une roche dans toutes les situations (sauf évidemment quand les composants ne sont plus différenciables visuellement).

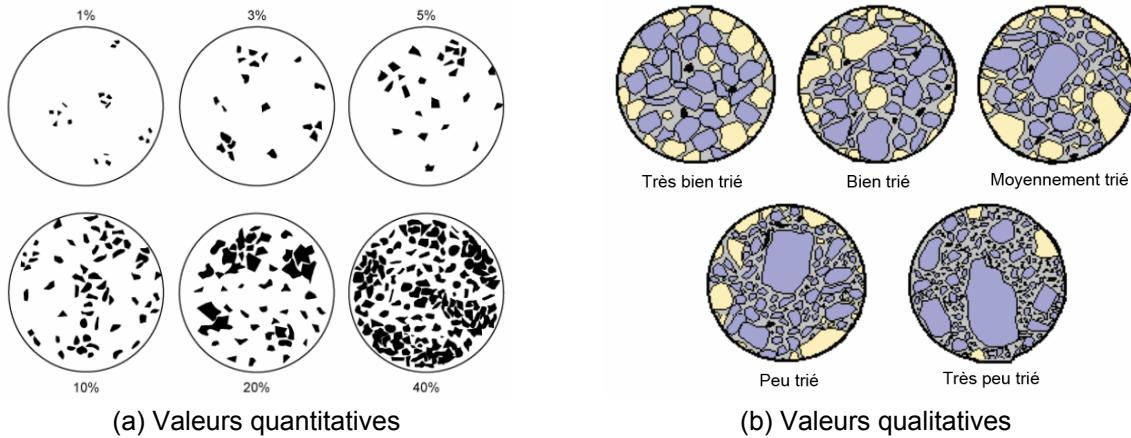


Figure 5.2 – Chartes visuelles

## 5.2 Comptage de point informatisé

Le comptage de points (Galehouse, 1971) est une technique qui permet de déterminer les proportions des différents éléments contenus dans une roche. Habituellement, il s'effectue sous le microscope au moyen d'un dispositif mécanique qui déplace la lame mince dans les deux dimensions du plan horizontal. Le dispositif est relié à un compteur, qui affiche le nombre de points comptabilisés par chacun des boutons ainsi que le nombre total de points recensés. Chaque bouton correspond à un composant pétrographique assigné par l'opérateur. Ce dernier identifie ce qu'il observe à la croisée des réticules et le comptabilise en appuyant sur le bouton correspondant. Chaque fois qu'un bouton est pressé, il entraîne le déplacement du chariot (selon X) à une distance choisie préalablement en fonction de la taille des éléments. Lorsqu'une ligne est terminée, le déplacement selon Y s'effectue manuellement. La lame mince est ainsi balayée de façon à échantillonner entre 300 et 500 points. Le nombre de points nécessaire à une représentation statistique satisfaisante peut varier selon le nombre de classes en jeu et selon l'arrangement et la taille des éléments (Neilson & Brockman, 1977).

Le comptage de points informatisé (Ndiaye, 2002) inclus dans JMicroVision est basé sur les mêmes principes que le comptage traditionnel, mais offre plusieurs avantages. Il ne requiert pas l'utilisation d'un microscope et, par conséquent, permet une observation bien plus confortable que celle au travers d'oculaires. Les vues simultanées et les différentes possibilités d'aménagement de l'interface utilisateur peuvent aussi atténuer la fatigue des yeux. Par exemple, l'utilisation des raccourcis clavier (au lieu de presser sur le bouton des classes)

permet à l'opérateur de garder son regard fixé sur le pointeur, ce qui a pour effet de réduire le mouvement des yeux et d'accélérer le processus de comptage.

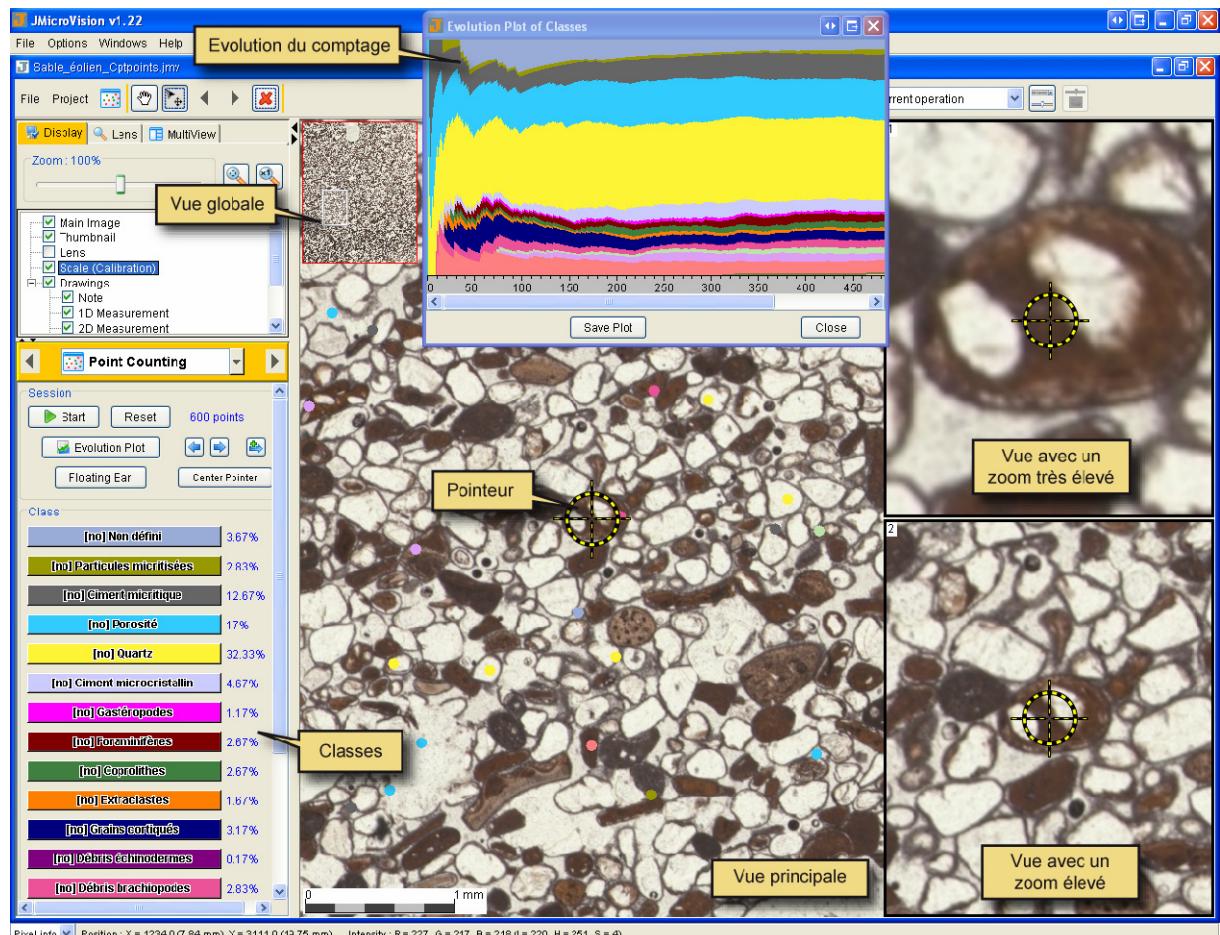


Figure 5.3 – Comptage de points sur un sable éolien

Pour effectuer un comptage de points, il faut d'abord créer une liste de classes représentant les différents composants pétrographiques. Chaque classe est définie par un nom, par une couleur et éventuellement par un raccourci clavier. Ensuite, une grille<sup>23</sup> ayant pour fonction de déterminer les positions successives du comptage doit être choisie. Plusieurs modèles sont proposés, notamment une grille aléatoire, où chaque nouveau point est choisi aléatoirement dans la surface de l'image, et une grille récursive, où les points sont localisés aux intersections d'une grille orthogonale. Celle-ci est au départ très grossière avec deux traits verticaux et horizontaux. Une fois les quatre intersections parcourues, de nouveaux traits sont insérés à

<sup>23</sup> Il existe aussi un mode sans grille, qui permet de dénombrer des objets en déposant des points manuellement.

mi-chemin entre les précédents. Ce processus se répète dès que toutes les positions ont été visitées. Lorsqu'un même composant pétrographique est réparti préférentiellement selon une direction horizontale ou verticale (faciès stratifié ou rubané), une grille récursive se montre moins adaptée qu'une grille aléatoire, car elle nécessite plus de points pour arriver à un résultat représentatif du fait que les points situés sur une ligne pourront tous appartenir à une même classe. Après avoir choisi la grille appropriée, l'opérateur peut encore appliquer des restrictions au comptage, comme l'arrêt après un certain nombre de points, la limitation à une zone personnalisée ou la restriction à des régions résultant d'une segmentation ou représentant un certain type d'objets vectoriels.

Une fois tous ces paramètres définis, le comptage peut démarrer. A chaque nouvelle position, le composant au centre du pointeur est identifié et comptabilisé par une pression sur le bouton correspondant ou à l'aide du raccourci clavier. Dans le comptage traditionnel, il est difficile de savoir combien de points sont nécessaires afin d'obtenir un échantillonnage satisfaisant du point de vue statistique. Pour qu'une décision concernant l'arrêt du comptage puisse être prise de façon méthodique, un graphique dans JMicrоЩision montre l'évolution des pourcentages des classes et permet à l'opérateur de voir quand les valeurs tendent à se stabiliser.

Les erreurs liées au comptage de points sont discutées par certains auteurs (Neilson & Brockman, 1977). Toutefois, elles sont pour la plupart liées au pas d'échantillonnage choisi et ne concernent donc pas l'utilisation d'une grille aléatoire. Une charte servant à estimer la précision ou l'erreur liée au comptage de points est proposée par Plas & Tobi (1965). C'est par l'hypothèse d'une roche homogène que cette erreur est calculée:

$$\varepsilon = \sqrt{\frac{f(100-f)}{n}} \quad (5.1)$$

où  $f$  représente le pourcentage d'une classe (ou la fréquence relative) et  $n$  le nombre total de points comptés. Les auteurs donnent un intervalle de confiance de 68% entre  $f-\varepsilon$  et  $f+\varepsilon$  et de 95% entre  $f-2\varepsilon$  et  $f+2\varepsilon$ .

Le comptage informatisé présente encore d'autres avantages. Il peut être sauvegardé et poursuivi plus tard. Des classes peuvent être ajoutées (sans limite), modifiées, supprimées et fusionnées même lorsque le comptage est en cours d'exécution. La classe attribuée à un point peut être changée en double-cliquant sur celui-ci ou en utilisant le système de navigation chronologique des points. Finalement, les résultats du comptage peuvent être exportés vers un tableur.

### 5.3 Estimation de la composition modale par segmentation

Lorsqu'il est possible de séparer les éléments par la couleur ou la luminosité, la composition modale peut être estimée par une méthode de segmentation. Dans la Figure 5.4, c'est par la segmentation par seuillage (voir Section 3.3) que les phases minérales sont extraites. Par défaut, la composition modale correspond à toute l'image. Cependant, elle peut par la suite être calculée à partir de zones définies par l'opérateur.

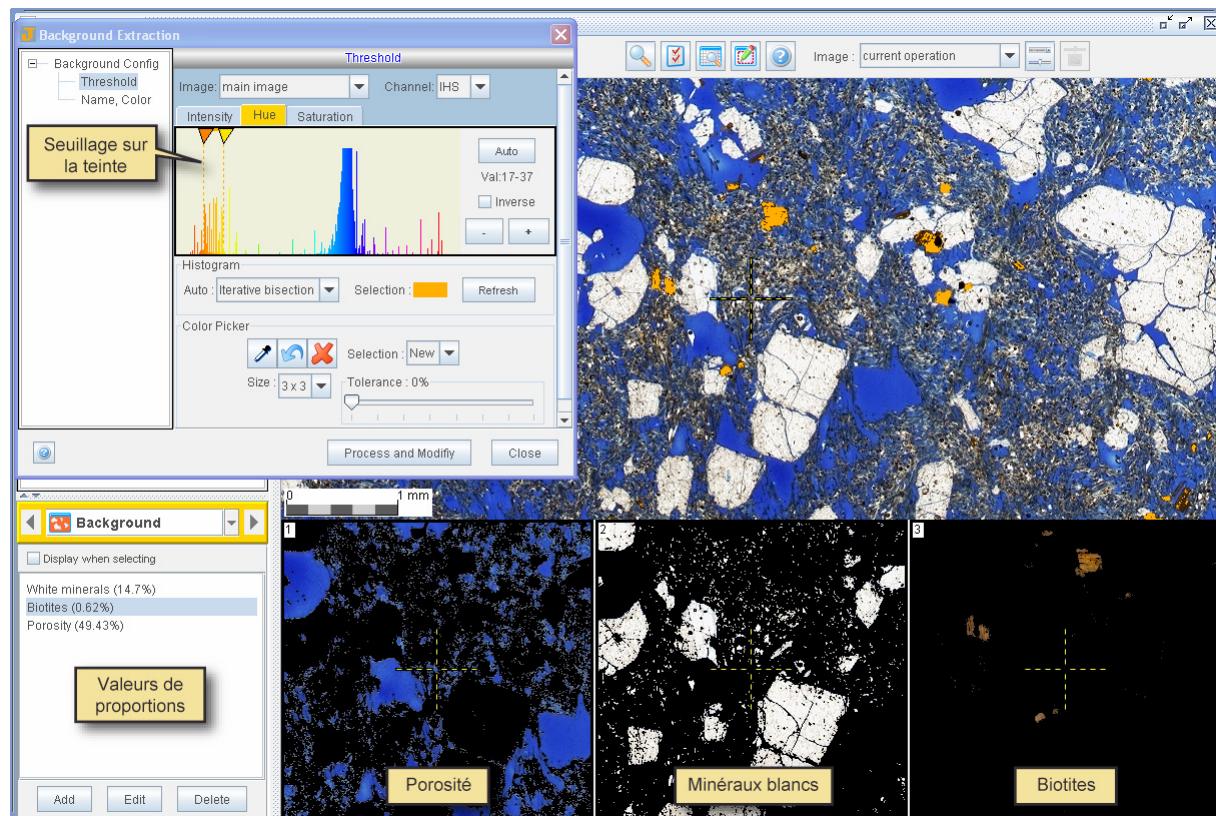


Figure 5.4 – Composition modale d'une roche volcanique

## 5.4 Caractérisation d'un faciès par un critère d'orientation

La Figure 5.5 met en évidence l'orientation des grains pour deux processus de dépôt d'un sable éolien: le grainflow (en haut) et le grainfall (en bas). Dans une première étape, les particules sont séparées à l'aide de l'algorithme SRC (voir Section 4.5.2). Plus d'une quarantaine de descripteurs (voir Section 4.7) sont ensuite calculés pour chaque particule.

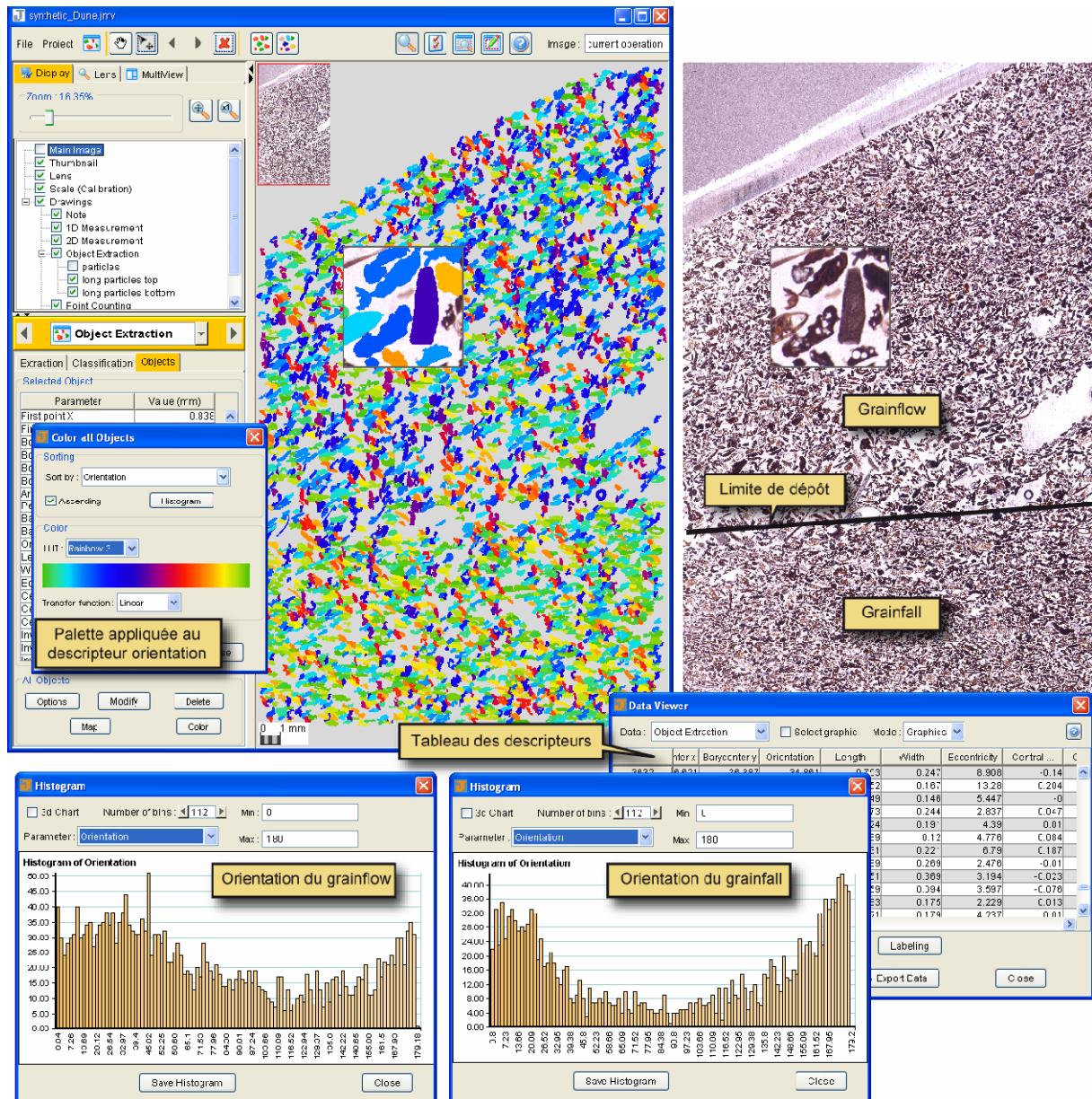


Figure 5.5 – Caractérisation de deux types dépôts selon l'orientation des grains

Les histogrammes des deux dépôts représentent la distribution des valeurs d'orientation des grains. Ils montrent des distributions unimodales (pour des données angulaires) avec un pic se

situant entre 30 et 35 degrés (angle critique de l'avalanche) pour le grainflow et un pic vers 0 degré pour le grainfall. Le paramètre d'orientation peut aussi être exprimé de façon visuelle directement sur les grains à l'aide d'une palette de couleurs. Cette dernière représente l'étendue de l'orientation de 0 à 180 degrés (sens trigonométrique) et la couleur de début et de fin est identique afin de respecter le caractère angulaire. Dans l'exemple en question, on peut constater que le vert, le turquoise et le jaune sont majoritaires dans la partie du bas et que dans celle du haut, c'est le bleu qui domine.



# Chapitre 6 : Granulométrie

## 6.1 Introduction

La granulométrie a pour objet de mesurer la dimension des grains et de représenter statistiquement leurs variations. L'aspect textural d'une roche ou d'un sédiment peut ainsi être caractérisé de manière quantitative. En sédimentologie, l'indice de tri et la taille des grains sont de bons indicateurs pour déterminer l'environnement de dépôt (Folk, 1974). Par ailleurs, la taille des grains détermine en partie les propriétés rhéologiques des roches (Fredrich & Evans, 1990; Prikryl, 2001) et influence la circulation des fluides dans les sables ou les roches poreuses. Dans le génie civil, ces deux aspects sont importants pour la réalisation d'ouvrages, pour la fabrication des bétons et pour la fragmentation des roches.

Les méthodes traditionnellement utilisées pour mesurer la dimension des grains sont le tamisage (Krumbein & Pettijohn, 1938) et la diffraction laser (Agrawal *et al.*, 1991). Dans JMicroVision, la granulométrie peut être traitée selon plusieurs approches. La taille des grains peut être déterminée par une analyse individuelle des particules (voir Section 6.2). Elle peut aussi être représentée par un profil granulométrique, qui exprime la variation de la taille des grains selon une direction (voir Section 6.3), ou par la méthode des ouvertures successives<sup>24</sup> (Russ, 1992; Vincent & LizardTech, 2000). L'avantage des deux dernières solutions est qu'elles ne nécessitent pas de séparation des grains au préalable.

Il existe d'autres approches granulométriques qui n'ont pas été implémentées dans JMicroVision. On trouve notamment une méthode qui transforme l'image en fréquence

---

<sup>24</sup> Le concept d'érosion-dilatation appliqué à la granulométrie a été introduit par Matheron (1967). Cette méthode peut être réalisée avec JMicroVision, mais elle ne figure pas dans les exemples en raison de sa lourdeur sur les images de haute résolution.

(transformation de Fourier) et qui classe les différentes tailles de grain à l'aide d'un réseau de neurones (Barron *et al.*, 1994). Cette approche semble toutefois limitée du fait que les composantes fréquentielles contiennent aussi les variations d'éclairage, de forme, d'orientation et de texture locale. Ces informations supplémentaires tendent à brouter considérablement le signal granulométrique.

## 6.2 Analyse individuelle des grains

En analyse d'image, la difficulté réside plus dans la séparation des particules lorsqu'elles sont connexes (voir Section 4.5) que dans la mesure de leur taille. Quand la séparation est réalisée avec succès, la granulométrie peut être exprimée par les valeurs représentant la surface ou le diamètre des particules. Le diamètre sphérique équivalent (voir Section 4.7.4) est souvent utilisé car il présente une valeur indépendante de la forme de l'objet. Quant aux diamètres de Feret (voir Section 4.7.5.1), ils mesurent une particule selon différentes orientations et, de ce fait, se rapprochent plus de la méthode mécanique du tamisage.

Les principaux paramètres caractérisant la distribution des diamètres ou des surfaces sont la moyenne, la médiane, le mode, l'indice de tri, le coefficient de dissymétrie (skewness) et le coefficient d'aplatissement (Kurtosis). Les trois derniers paramètres peuvent être calculés selon la méthode de Folk (1974). La distribution peut aussi être représentée de manière graphique par un histogramme (voir Figure 6.2) ou par une courbe granulométrique des fréquences cumulées.

Pour obtenir des résultats valides, il est important de considérer différents effets perturbateurs. Il s'agit notamment de s'assurer que l'image possède une résolution suffisante pour mesurer les objets d'intérêt (voir Section 4.7). Pour les images représentant une section, il faut en outre tenir compte du fait que les surfaces ou les diamètres des particules ont tendance à être sous-estimés. En effet, pour un grain subsphérique, à moins que la section ne passe par son centre de gravité, le diamètre de la section sera forcément plus petit que celui du grain (en 3D). Des formules empiriques pour convertir la moyenne des valeurs observées ont été proposées par certains auteurs (Harrell & Eriksson, 1979; Johnson, 1994). Elles restent cependant discutables, car elles semblent être liées à un certain type de faciès. Des méthodes plus

sophistiquées, basées sur les principes de stéréologie<sup>25</sup>, permettent de corriger les distributions de taille des grains (Howard & Reed, 1998; Russ, 1990; Sahagian & Proussevitch, 1998; Saltykov, 1967; Underwood, 1970; Weibel, 1979). Higgins (2000) propose un logiciel<sup>26</sup> de correction stéréologique pour la distribution de la taille des grains ayant une forme spécifique. Toutefois, ces méthodes de correction ne sont applicables qu'à des faciès ayant une population de grains avec des formes similaires et relativement simples. En raison de ces restrictions, elles n'ont pas été implémentées dans JMicroVision.

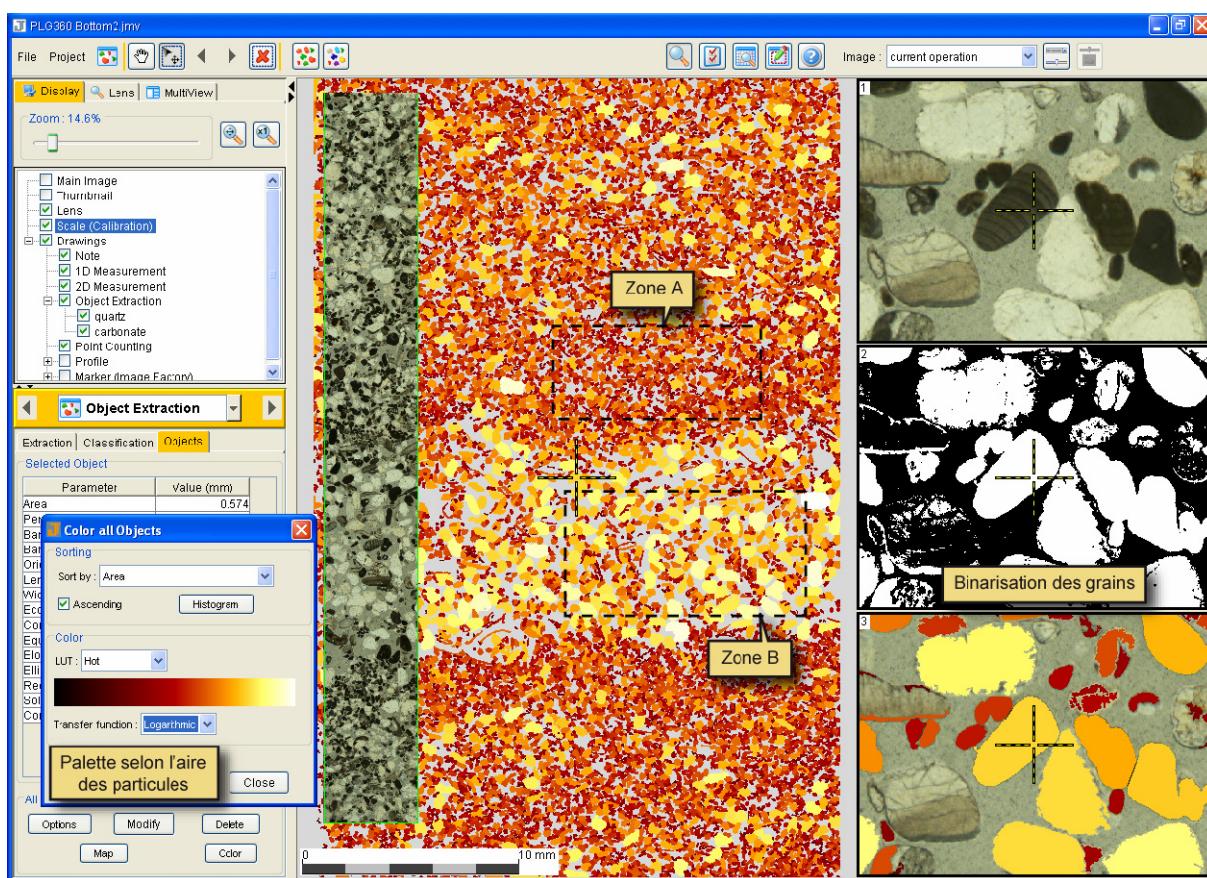


Figure 6.1 – Mesure de la surface des grains dans un faciès éolien

Dans la Figure 6.1, l'image est d'abord binarisée par seuillage (voir Section 3.3) et les grains sont ensuite séparés par l'algorithme SRC (voir Section 4.5). Une signature granulométrique locale est exprimée par deux histogrammes (sans correction stéréologique, voir Figure 6.2)

<sup>25</sup> La stéréologie est une branche de la science qui a pour but de déduire les propriétés en trois dimensions d'objets observés en deux dimensions.

<sup>26</sup> CSD Corrections peut être téléchargé à l'adresse <http://geologie.uqac.ca/~mhiggins/csdcorrections.html>.

représentant chacun une zone (en pointillé noir) et montrant une distribution différente. Ces variations sont aussi mises en évidence de manière visuelle en utilisant une palette qui donne une couleur aux grains par rapport à leur valeur de surface.

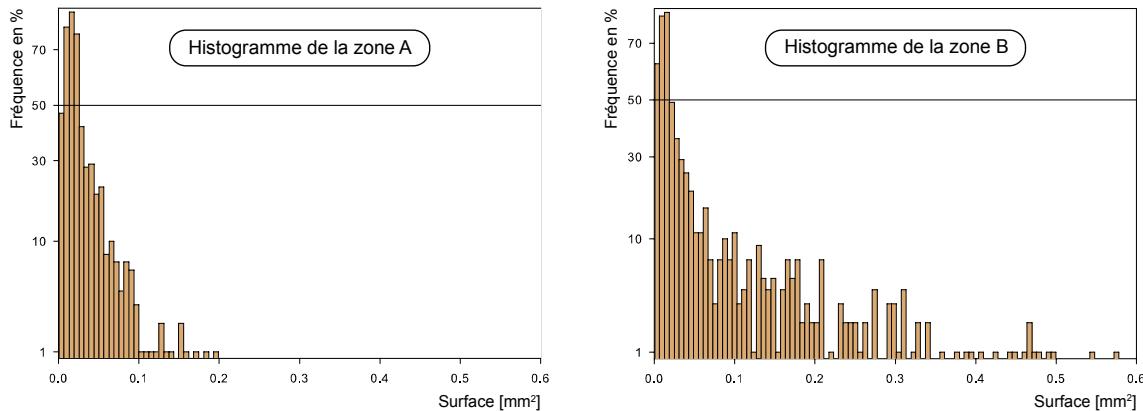


Figure 6.2 – Histogrammes de la surface des grains

Dans la Figure 6.3, l'image source<sup>27</sup> est d'abord rendue plus uniforme localement en appliquant un filtre médian (voir Section 2.2.2). Puis l'image du gradient est calculée à partir de l'opérateur de Sobel. Cette dernière image est seuillée afin d'en extraire les zones de fort contraste. Une érosion<sup>28</sup> des régions blanches est ensuite effectuée pour accentuer leurs séparations et éliminer les très petites régions. L'image érodée est finalement utilisée comme image des marqueurs pour la segmentation par la ligne de partage des eaux (voir Section 3.5.1). Une fois la séparation réalisée, les mesures individuelles (voir Section 4.7) peuvent être employées pour caractériser l'ensemble des grains.

<sup>27</sup> L'image a été acquise en lumière réfléchie et représente des grains très grossiers d'aluminium recristallisé (Philpotts, 1988).

<sup>28</sup> L'érosion est une opération de morphologie mathématique réalisée à l'aide d'un élément structurant.

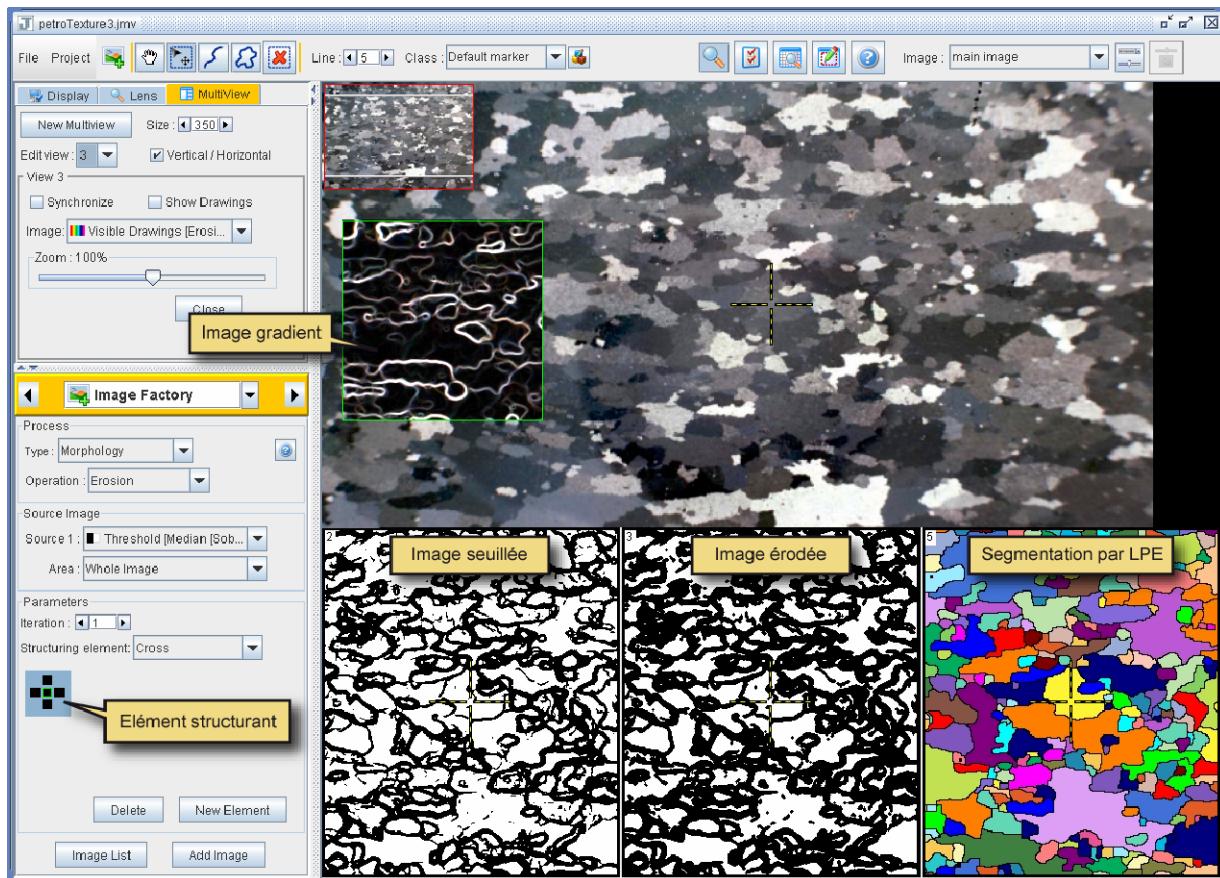


Figure 6.3 – Processus d'individualisation des grains par LPE

### 6.3 Profils granulométriques

L'outil "profil" permet d'établir des courbes granulométriques selon une direction. Il est utile lorsque la séparation des objets ne peut être réalisée de manière satisfaisante ou lorsque l'on désire obtenir rapidement des renseignements sur la variation de la taille des grains. L'algorithme est basé sur celui proposé par Rubin (2004), qui montre que l'autocorrélation spatiale varie en fonction de la taille des grains. C'est le facteur de corrélation entre une ligne de l'image et cette même ligne décalée de 4 pixels<sup>29</sup> qui détermine la valeur d'un point du profil. Ainsi pour un profil vertical, l'autocorrélation d'une ligne de l'image représente la valeur (entre 0 et 1) du point de la courbe granulométrique situé à la même hauteur. Les lignes contenant des gros grains produisent un facteur de corrélation élevé, alors que celles avec de petits grains donnent une valeur plus faible.

<sup>29</sup> La valeur de 4 a été choisie de manière empirique. Elle doit néanmoins être plus petite que la taille des grains.

Pour que les résultats soient valides, les interstices ne doivent pas être plus grands que les particules, sinon les valeurs de corrélation rendront compte de la taille des interstices et non plus de celle des grains. De plus, les valeurs d'intensité entre les particules et les interstices doivent être suffisamment contrastées.

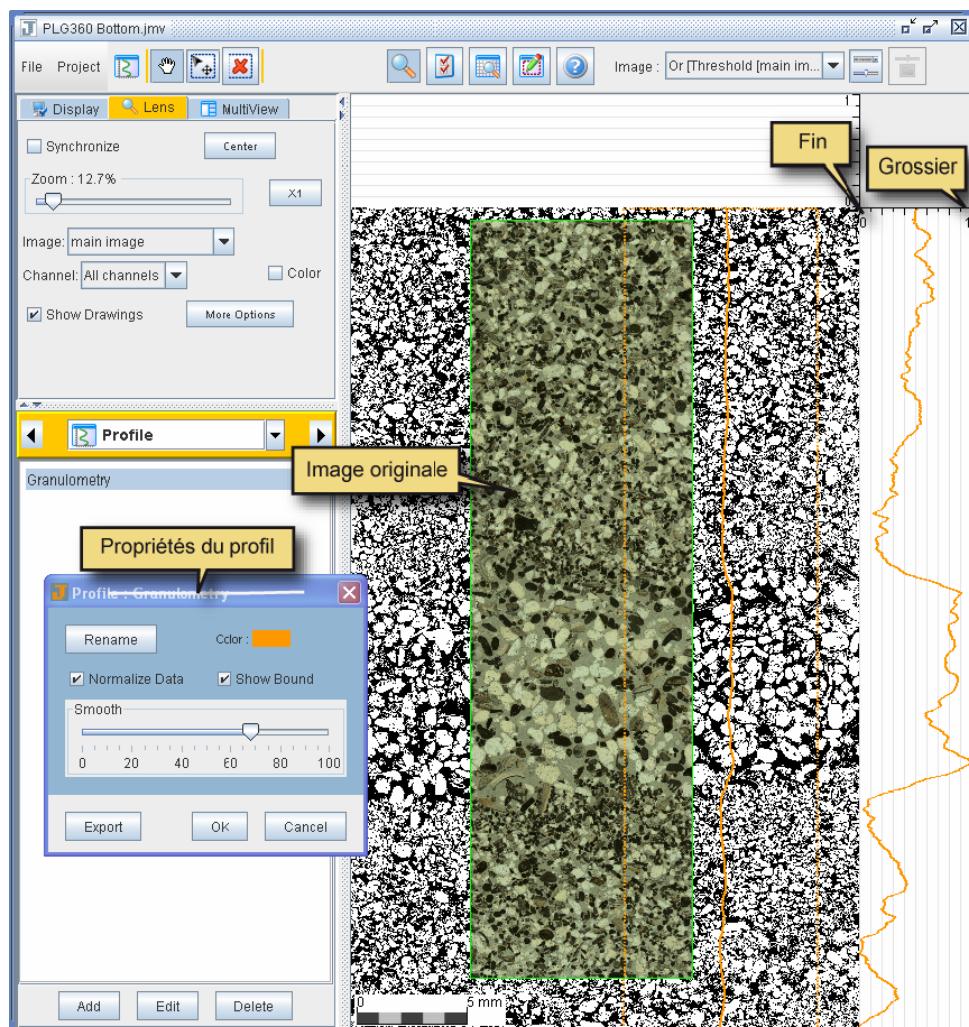


Figure 6.4 – Profil granulométrique sur un sable

Dans la Figure 6.4, le profil est calculé sur l'image binaire car celle-ci permet d'obtenir de meilleurs résultats que l'image originale, où les grains de quartz ne se diffèrentient que très peu des interstices. C'est l'opérateur qui définit l'étendue et la direction de la zone rectangulaire (en pointillé orange) dans laquelle le processus sera effectué. Une fois créés, les profils peuvent être lissés pour diminuer le bruit et normalisés pour accentuer les variations

faibles. Pour obtenir des valeurs granulométriques réelles, l'image doit être calibrée et des mesures de plusieurs diamètres de grain doivent être effectuées (avec l'outil de mesure 1D) dans les zones de variation maximale du profil et reportées sur ce dernier. Les profils sont sauvegardés dans le fichier projet de JMicrоЩision et sont exportable dans le format vectoriel SVG ou simplement dans un fichier texte, où les valeurs successives de la courbe sont écrites.



# Chapitre 7 : Porosité

## 7.1 Introduction

La porosité représente le pourcentage de vide par rapport au volume total d'une roche ou d'un sédiment. Elle se mesure notamment à l'aide d'un porosimètre ou est obtenue par les logs (densité et neutron) lors de forages. Associée à la porosité, la notion de perméabilité correspond à la capacité d'une roche à laisser passer un fluide (air, eau, huile ou gaz) au travers de son réseau poreux. Dans l'industrie pétrolière, ce paramètre contrôlera à la productivité d'un réservoir, alors que la porosité renseigne sur la capacité du réservoir à stocker les hydrocarbures fluides. Ce chapitre aborde la problématique de l'estimation de la porosité et de la perméabilité à partir d'images 2D. Des méthodes visant à quantifier les types de pores et à caractériser le réseau poreux sont présentées au travers d'exemples.

## 7.2 Estimation de la porosité et de la perméabilité

### 7.2.1 La porosité

Les techniques d'estimation de la porosité par analyse d'image ont déjà été présentées dans plusieurs travaux (Anselmetti *et al.*, 1998; Ehrlich *et al.*, 1991a; Gerard *et al.*, 1992; Gies, 1987). Il en ressort que la porosité doit être étudiée à différentes échelles pour obtenir une évaluation globale. L'échantillon peut être imprégné d'une résine colorée ou fluorescente afin de faciliter la segmentation des pores. De manière générale, la porosité observée sur une lame mince peut être exprimée ainsi:

$$\Phi_{total} = \Phi_{macro} + \Phi_{micro} F_{micro} \quad (7.1)$$

où  $\Phi_{macro}$  est la porosité observée à l'aide d'un microscope optique (voir Section 2.1.4) ou d'un scanner à diapositives (voir Section 2.1.3),  $\Phi_{micro}$  la microporosité ( $\sim < 300 \mu\text{m}^2$ ) observée au MEB, et  $F_{micro}$  la fraction microporeuse ou la proportion de surface contenant de la microporosité, qui est égale à  $(1 - \Phi_{macro})$  lorsque l'échantillon entier en contient.

Dans JMicronVision, la porosité est généralement extraite de l'image par la méthode du seuillage (voir Section 3.3). Lorsque les pores sont colorés en bleu, la sélection peut s'opérer sur les valeurs du canal de la teinte (voir Section 3.2.1.3). Sinon, les pores peuvent être distingués par l'observation en lumière polarisée<sup>30</sup> ou par une combinaison (par l'opérateur logique ET, voir opérateur dans le guide utilisateur) des images seuillées en lumière normale et en lumière polarisée.

### 7.2.2 La perméabilité

Si l'on exclut les mesures pétrophysiques, la perméabilité peut être estimée soit en caractérisant l'espace poreux soit en étudiant le milieu qui l'entoure. Cette dernière approche peut indiquer une tendance d'après la taille ainsi que le degré de tri et de compaction des grains. En effet, la porosité et la perméabilité tendent à être plus grandes lorsque le matériel est plus grossier et mieux classé. Cependant, les variations de perméabilité ne peuvent s'expliquer uniquement par ces critères, car les réseaux poreux sont parfois très complexes en raison des transformations des roches au cours des temps géologiques par les phénomènes de fracturation, de métamorphisme et de diagenèse (cimentation, dissolution, recristallisation...). De plus, la perméabilité (dans un milieu poreux multiphasique) est aussi influencée par des paramètres pétrophysiques tels que la pression capillaire ou la saturation irréductible.

En raisonnant en termes d'espace poreux, les facteurs les plus déterminants de la perméabilité spécifique<sup>31</sup> sont la connectivité et la morphologie du réseau poreux ainsi que la taille des seuils de pores<sup>32</sup> (Ehrlich et al., 1991b). Comme l'imagerie 2D permet uniquement de

<sup>30</sup> La porosité a tendance à être sous-estimée lors de la sélection des valeurs sombres en lumière polarisée, car seuls les espaces vides traversant toute l'épaisseur de la lame apparaissent très sombres.

<sup>31</sup> La perméabilité spécifique est la capacité d'une roche à laisser passer un seul fluide (air, eau ou huile) sans présence au préalable d'autres fluides.

<sup>32</sup> La taille d'un seuil de pore est le diamètre le plus étroit du conduit reliant deux pores.

caractériser la morphologie des pores en section, l'estimation de la perméabilité sur des images 2D semble donc irréalisable. Seule l'imagerie 3D pourrait traiter ces différents paramètres, mais elle se présente comme une technique lourde et difficile à mettre en œuvre (Sallier, 2005).

Plusieurs auteurs ont tout de même proposé des méthodes d'estimation de la perméabilité à partir des techniques d'analyse d'image 2D. Anselmetti et al. (1998) ont avancé l'idée que la perméabilité pouvait être évaluée à partir de la compacité des macropores (voir Section 4.7.4). Par une simplification des équations de Kozeny-Carman, qui modélisent la perméabilité, Solymar (1998) et Cerepi et al. (2001) ont proposé une formule où seuls la surface et le périmètre des pores sont requis. Toutes ces approches nous semblent être réductrices du fait qu'elles reposent uniquement sur une information en 2D, représentée par la surface et le périmètre des pores, pour rendre compte d'un réseau poreux complexe en 3D. En analysant quelques cas simples, on s'aperçoit que les hypothèses avancées par ces auteurs ne peuvent être vérifiées.

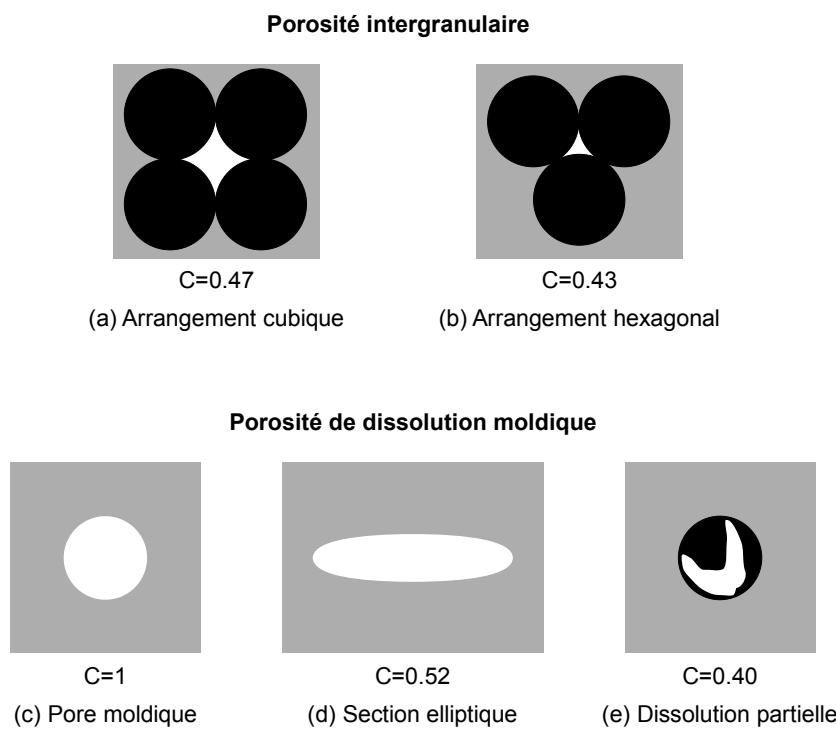


Figure 7.1 – Compacité de différents types de pore

Dans la Figure 7.1, différents types de pore (représentés en blanc) sont mesurés selon la compacité (voir Section 4.7.4), qui exprime la "dispersion" de la forme par une valeur comprise entre 0 et 1, où 1 correspond à la forme la plus ramassée, soit une sphère. De manière générale, le coefficient de perméabilité des roches renfermant des pores intergranulaires devrait être plutôt élevé, alors que celles contenant des pores de dissolution moldique devrait être faible (Lucia, 1995). Selon Anselmetti et al. (1998), plus la compacité est faible, plus le pore sera ramifié et, par conséquent, plus le réseau sera connecté et aura donc un coefficient de perméabilité élevé. Cependant, comme nous allons le voir, cette affirmation ne se confirme pas:

- L'illustration (b) révèle une valeur de compacité plus faible que celle de (a) alors même que le pore de (b) possède moins de branches et que l'arrangement hexagonal devrait avoir une perméabilité plus faible en raison de ses seuils de pores plus petits.
- Bien que le pore de l'illustration (d), représentant la section d'un grain ellipsoïdal dissous, ne possède pas de branches, sa valeur de compacité est néanmoins proche de celle de (a).
- Lorsque la dissolution est partielle, comme dans l'illustration (e), les pores peuvent aussi avoir des valeurs de compacité très faibles.

En conclusion, la détermination de la perméabilité par les méthodes proposées par Anselmetti et al. (1998), Solymar (1998) et Cerepi et al. (2001) semble peu fiable.

## 7.3 Classification des types de porosité

### 7.3.1 Introduction

L'estimation de la perméabilité par la proportion des types de pore du point de vue pétrophysique apparaît comme une approche plus raisonnable que les méthodes basées uniquement sur la surface et le périmètre des pores, discutées à la section précédente. Lucia (1995) a montré que la perméabilité peut en effet être évaluée selon la proportion de pores connectés, mal-connectés ou non-connectés ainsi que par le type de faciès selon la classification de Dunham (1962). Ainsi, la porosité interparticulaire ou intercristalline se présente comme celle ayant la plus grande capacité à laisser circuler les fluides, alors que la microporosité, la porosité intraparticulaire et la porosité de dissolution moldique ne contribuerait que dans une faible mesure à la perméabilité. Quant à la fracturation et à la

porosité de dissolution élargie, elles devraient être considérées comme un réseau poreux indépendant du réseau interparticulaire. Dans les sections suivantes, deux méthodes permettant d'estimer la proportion des types de pore sont présentées: la première se base sur une classification des pores selon un ou plusieurs descripteurs et la seconde sur un comptage de points restreint à la porosité.

### 7.3.2 Proportion des types de pore par analyse individuelle

Dans l'exemple de la Figure 7.2, les pores sont d'abord sélectionnés par seuillage, puis ils sont analysés individuellement de manière automatique. C'est par le paramètre additionnel "contexte" (voir Section 4.7.5.2), décrivant l'environnement autour du pore, que les pores inter- et intra-nummulitiques ont pu être différenciés. Comme le contraste entre les valeurs d'intensité de la matrice (sombres) et des nummulites (claires) est important, ce descripteur s'est révélé suffisamment discriminant pour permettre une classification automatique des pores (3<sup>ème</sup> méthode, voir Section 4.8).

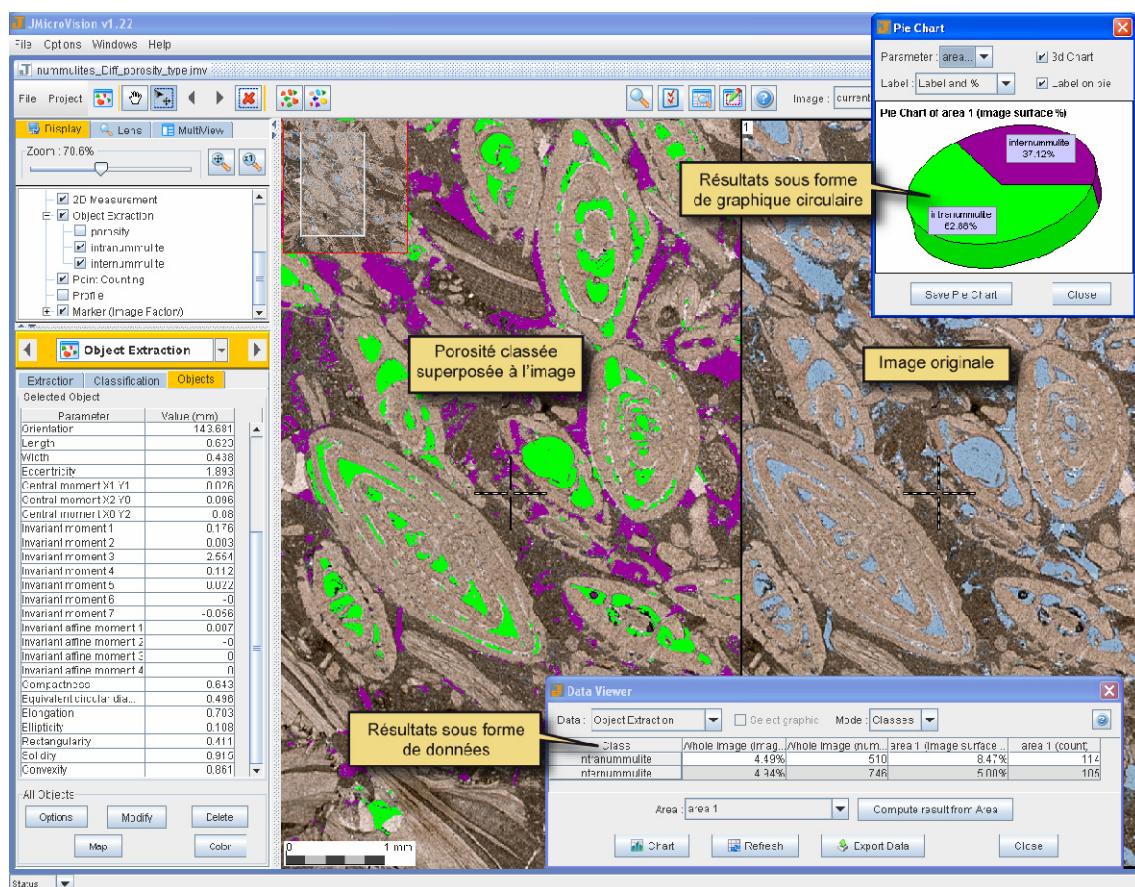


Figure 7.2 – Proportion des types de pore par analyse individuelle

### 7.3.3 Proportion des types de pore par comptage de points

Lorsque les descripteurs ne permettent pas de différencier les types de pore, il faut recourir à la méthode du comptage de points. Dans la Figure 7.3, le comptage de points (voir Section 5.2) a été effectué exclusivement dans la porosité pour déterminer la proportion des différents types de pore. Une grille aléatoire a été choisie pour le comptage et celui-ci est limité à la porosité colorée en bleu, qui a été segmentée au préalable par seuillage (voir Section 3.3).

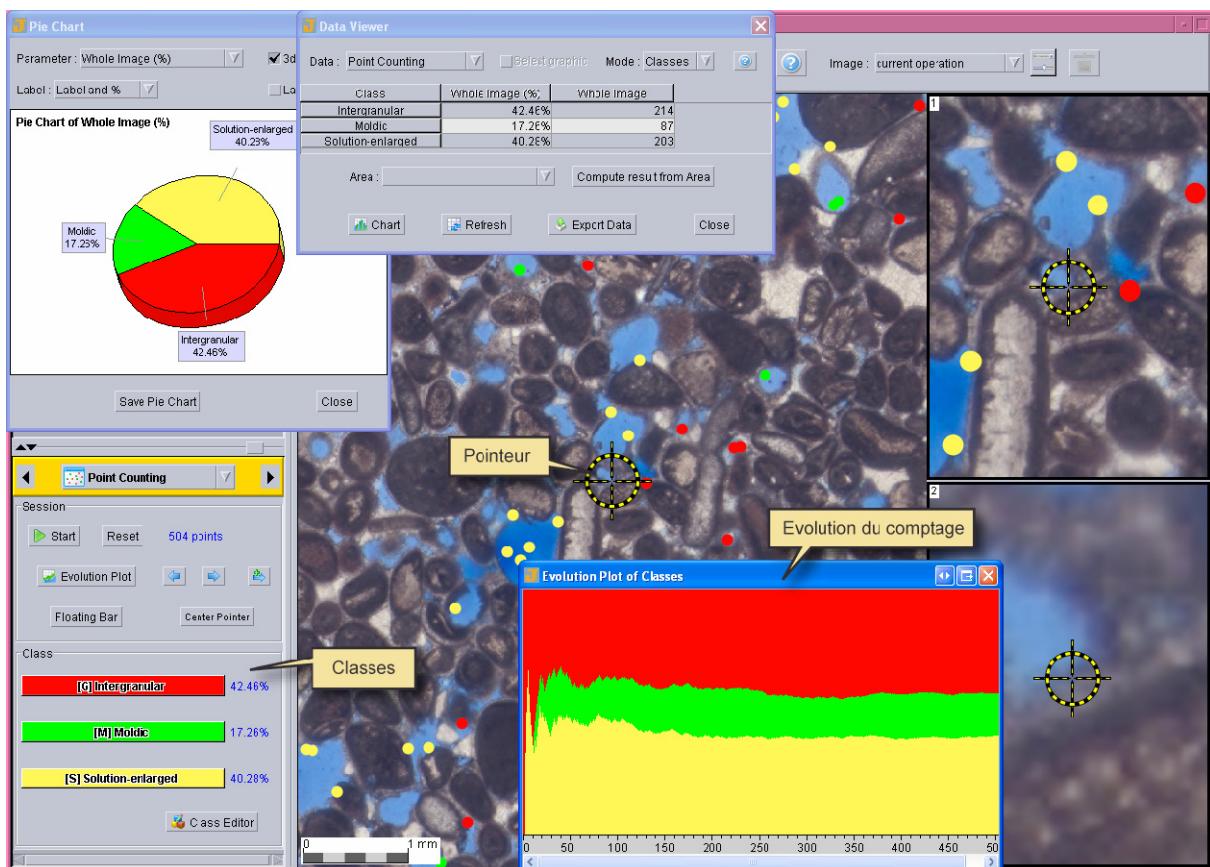


Figure 7.3 – Proportion des types de pore par comptage de points

Le diagramme d'évolution du comptage fait ressortir que la proportion des types de pore se stabilise à partir de 250 points. Sachant que la porosité dans l'image est de 6,1%, on peut estimer qu'il faudrait, sans limiter le comptage aux régions poreuses, au moins 4098 points ( $= 250 / 0,061$ ) pour obtenir un résultat similaire. Ce nombre de points est bien supérieur aux 300 ou 500 points comptés habituellement sous le microscope pour ce genre d'analyse. Si l'on se reporte au graphique d'évolution, seuls 36 points comptés exclusivement dans la porosité correspondent aux 500 points comptés sans restriction. Cela montre qu'un comptage de 500

points, effectué normalement, est largement insuffisant pour comptabiliser de manière précise la proportion des sous-classes d'un composant représenté à raison de moins de 10% dans l'image.

## 7.4 Caractérisation du réseau poreux

### 7.4.1 Introduction

Les méthodes présentées dans les deux sections suivantes permettent de caractériser un réseau poreux plutôt que de traiter les pores de manière individuelle. Elles sont utiles lorsque la porosité est élevée et que les pores ne peuvent plus être considérés séparément. Dans la Figure 7.4, les images A et B ont été acquises au MEB (voir Section 2.1.5). Elles montrent des configurations différentes de réseaux microporeux (représentés en noir).

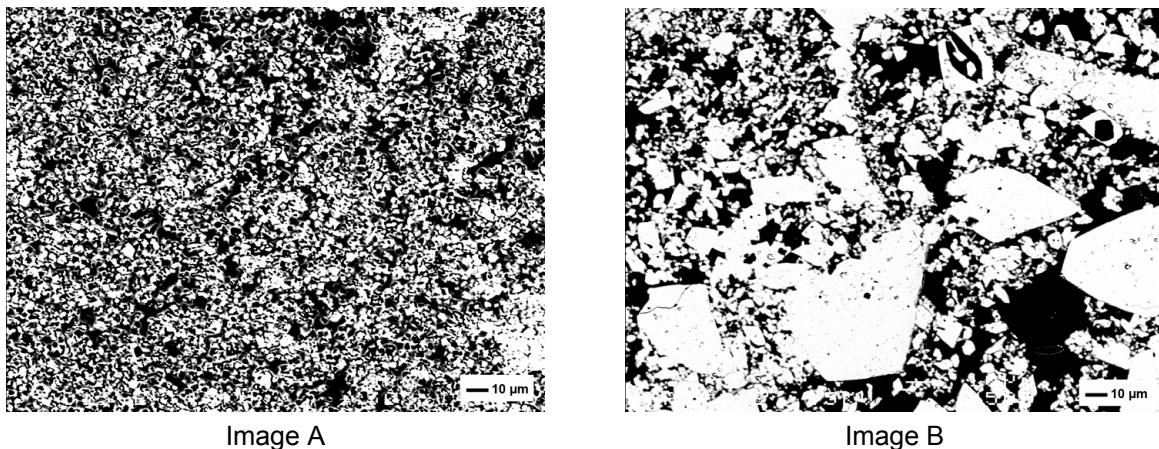


Figure 7.4 – Exemples de réseaux microporeux dans des roches carbonatées

### 7.4.2 Distribution de la taille des pores

Comme la taille des pores n'est pas directement mesurable en raison de la difficulté à les individualiser, c'est par une carte des distances que le réseau poreux va pouvoir être caractérisé. La porosité est d'abord seuillée (voir Section 3.3), puis l'image "distance", illustrée à la Figure 7.5, est obtenue par la fonction "Euclidian distance map", où chaque valeur représente la plus courte distance entre le pixel et le bord du pore. Les valeurs de distance sont calculées selon l'algorithme proposé par Russ (1992) et se rapprochent de vraies distances euclidiennes.

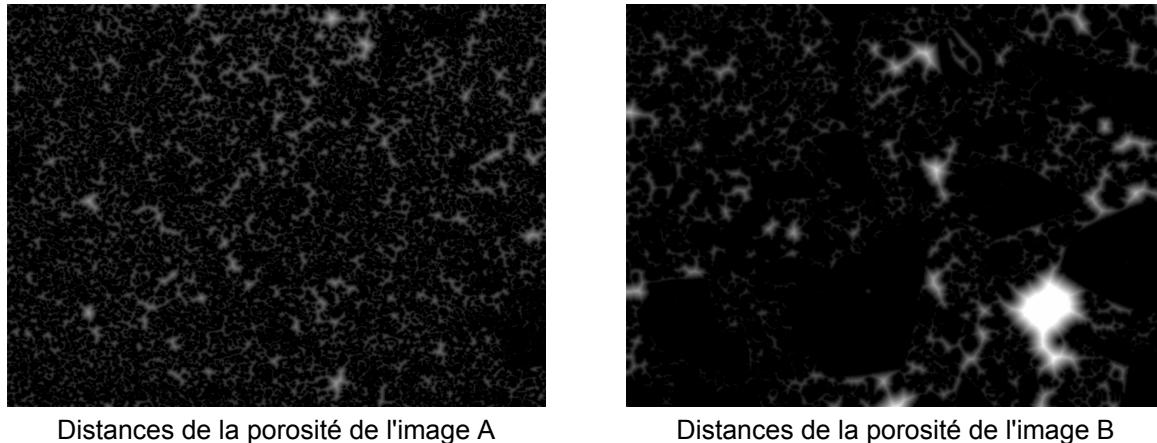


Figure 7.5 – Carte des distances du réseau microporeux

Une autre façon de caractériser le réseau poreux est de mesurer le pourcentage de son envahissement par des sphères de différentes grandeurs. Par exemple, pour des sphères d'un diamètre de 20 pixels, on comptabilise toutes celles qui peuvent être inscrites dans le réseau poreux et on établit la proportion de remplissage. Cette opération<sup>33</sup> est effectuée par un algorithme qui cherche à placer les sphères dans le réseau poreux à l'aide de l'image "distance".

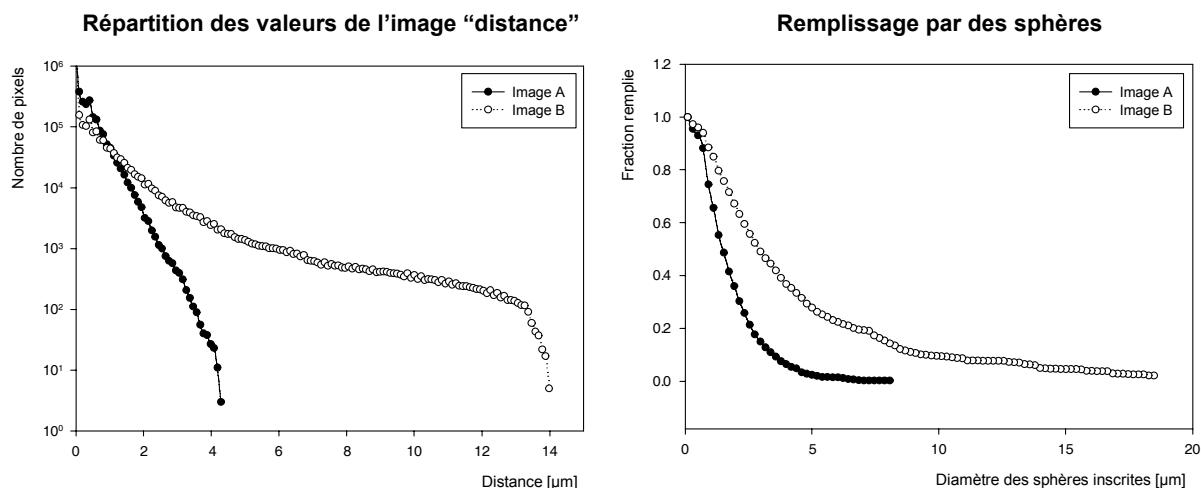


Figure 7.6 – Caractérisation du réseau microporeux

Dans la Figure 7.6, le premier graphique exprime les fréquences des valeurs de pixels (ici converties en microns). Ces dernières ont été récupérées à l'aide de l'outil de mesure 2D

<sup>33</sup> Cette opération peut être exécutée avec le plugin "Granulometry".

pouvant extraire les données statistiques d'une image. Quant au graphique de droite, il montre le degré de remplissage du réseau poreux en fonction du diamètre des sphères inscrites.

### 7.4.3 Homogénéité du réseau poreux

Le paramètre homogénéité<sup>34</sup> exprime la variation de la porosité à différentes échelles. L'image est découpée en une multitude de carrés de dimensions identiques et la porosité est calculée pour chaque carré. La déviation standard permet de mesurer le degré de fluctuation de la porosité pour l'ensemble des carrés. Dans la Figure 7.7, on peut donc observer la variabilité de la porosité en fonction de la dimension des carrés.

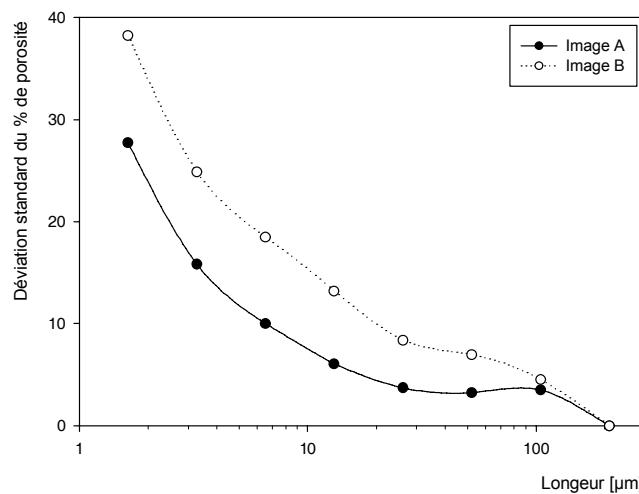


Figure 7.7 – Homogénéité du réseau poreux

<sup>34</sup> Paramètre additionnel de l'outil "Background" (voir le paramètre "homogeneity" dans le guide utilisateur).



# **Chapitre 8 : Conclusion**

Ce chapitre clôture la présentation des méthodes et des applications qui font l'objet de cette thèse. Dans un rapide survol, nous mettrons l'accent sur nos apports dans les différents outils du logiciel. Finalement, il sera question des perspectives qu'ouvre ce travail.

## **8.1 Apports de cette thèse**

L'objectif principal de cette thèse était de fournir une palette d'outils informatiques adaptés à la visualisation et à l'analyse d'images pétrographiques. Ces outils ont été regroupés dans le logiciel JMicrоЩision. Ce dernier a été conçu de manière à pouvoir afficher et traiter des images haute résolution dans des conditions optimales. A l'instar d'un microscope, il permet une observation dynamique d'une lame mince, même en combinant différents modes d'éclairage (lumière polarisée, cathodoluminescence...) ou focus. De plus, la loupe et l'outil multivue permettent une vision simultanée de plusieurs images, chacune pouvant avoir un coefficient de zoom propre tout en conservant une position commune au centre. Même avec des images de très grandes tailles et un ordinateur doté d'une faible capacité mémoire, les outils de visualisation se révèlent performants grâce à la technique du tuilage et à l'exécution différée.

Les caractéristiques des objets extraits de l'image peuvent être mises en évidence de manière visuelle soit par divers diagrammes soit par une coloration des objets en fonction de leur valeur (selon le descripteur sélectionné). Le rendu visuel d'un projet ou d'une portion de celui-ci peut être exporté sous forme d'images dans plusieurs formats ou sous forme de graphiques vectoriels dans le format SVG. De même, toutes les données sont exportables dans le format CSV.

Nous retiendrons que le choix du dispositif d'acquisition dépend principalement de ce que l'on veut observer et à quelle résolution. Le scanner à diapositive reste toutefois un moyen simple et efficace, qui permet de numériser entièrement les lames minces avec une résolution généralement suffisante pour l'observation des composants pétrographiques.

Comme les images pétrographiques sont variées et complexes, nous avons vu qu'il était difficile d'établir des règles précises concernant l'utilisation des méthodes de prétraitement et de segmentation. L'intuition de l'opérateur reste primordiale pour envisager la suite des opérations la plus adaptée et pour le réglage des paramètres. La deuxième partie de cette thèse propose néanmoins plusieurs marches à suivre pour des analyses classiques en pétrographie. Nous avons pu constater que la segmentation par seuillage se montre efficace dans de nombreux cas. Quant à la segmentation par la ligne de partage des eaux à l'aide de marqueurs, elle représente un outil puissant, mais limité à cause de la difficulté à obtenir automatiquement des images de marqueurs.

Nous avons proposé une nouvelle méthode de séparation des objets connexes à la Section 4.5 et nous avons pu constater que celle-ci était nettement plus robuste sur les particules allongées ou celles contenant des trous que les méthodes traditionnelles. Afin d'optimiser la gestion de la mémoire de plusieurs opérations (voir Section 4.4), nous avons adapté l'algorithme de croissance de régions pour qu'il soit conforme au tuilage et à l'exécution différée.

Lorsque les différentes méthodes de segmentation échouent, il reste le détourage manuel des contours à l'aide du dispositif de pointage, qui s'avère une tâche fastidieuse. Cependant, nous avons vu que le comptage de points informatisé pouvait représenter une alternative à la segmentation en ce qui concerne l'estimation de la composition modale et que, lorsque le comptage est limité à un type de région (après segmentation), l'estimation des sous-classes peut être effectuée de manière rapide et efficace.

En ce qui concerne l'estimation de la perméabilité à partir des images 2D, nous retiendrons que les méthodes proposées dans la littérature s'avèrent peu concluantes et que les deux méthodes de quantification des types de pores de la Section 7.3 permettent une estimation

plus acceptable. Finalement, nous avons constaté que le descripteur "contexte", proposé à la Section 4.7.5.2, pouvait être utile pour la reconnaissance des types de pore. Ces derniers peuvent être classés selon plusieurs méthodes, dont la classification supervisée précédée d'une phase d'apprentissage.

## 8.2 Perspectives

D'autres disciplines, notamment la biologie, la physique, la médecine, l'agronomie, l'agro-alimentaire, l'environnement et l'archéologie pourraient tirer profit de JMicrоЩision du fait que celui-ci contient de nombreux outils de mesure et de quantification, et n'intègre pas d'éléments spécifiques à la pétrographie dans l'interface utilisateur. Ces éléments sont soit abstraits soit ajoutés sous forme de plugins.

Dans l'agro-alimentaire, par exemple, l'algorithme SRC de la Section 4.5 pourrait s'avérer très approprié pour la séparation des grains de céréales (Zhang *et al.*, 2005). Quant à l'algorithme de croissance de régions adapté au tuilage, il pourrait être fort utile pour toutes les disciplines qui doivent traiter des images de très grandes tailles (notamment l'imagerie satellitaire). Il présente l'avantage d'être exécutable même sur un matériel informatique peu performant.

A titre de recommandation pour les applications pétrographiques, il serait judicieux de répertorier les différentes méthodes de coloration des phases minérales et d'imprégnation, voire d'en expérimenter de nouvelles. Cela pourrait grandement faciliter l'étape de la segmentation.

L'intégration d'un langage script dans le logiciel pourrait amener un plus grand degré d'automatisation et de souplesse. Toutefois, il est déjà possible dans JMicrоЩision d'appliquer à un nouveau projet différentes opérations (dont toutes celles sur les images) issues d'un projet antérieur. Des méthodes de segmentation et de classification supplémentaires ainsi que de nouveaux descripteurs (voir Section 4.7.6) pourraient être ajoutés au logiciel afin de le rendre

plus performant. JMicrоЩision reste néanmoins ouvert au développement grâce à la possibilité d'intégrer de nouvelles fonctionnalités par le biais de plugins<sup>35</sup>.

---

<sup>35</sup> La marche à suivre pour développer de nouveaux plugins et la documentation des classes se trouvent à l'adresse <http://www.jmicrovision.com/help/buildPlugins.html>.

# Bibliographie

- Agrawal, Y. C., McCave, I. N., & Riley, J. B.** (1991). Laser diffraction size analysis. *Principles, Methods and Application of Particle Size Analysis*, 119-128.
- Anselmetti, F. S., Luthi, S. M., & Eberli, G. P.** (1998). Quantitative characterization of carbonate pore systems by digital image analysis. *AAPG Bulletin*, 82(10), 1815-1836.
- Bailey, E. H., & Stevens, R. E.** (1960). Selective staining of k-feldspar and plagioclase on rock slabs and thin sections. *American Mineralogist*, 45, 1020–1025.
- Barber, C. B., Dobkin, D. P., & Huhdanpaa, H.** (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4), 469-483.
- Barron, L., Smith, M. L., & Prisbrey, K.** (1994). Neural network pattern recognition of blast fragment size distributions. *Particulate science and technology*, 12(3), 235-242.
- Beucher, S., & Lantuejoul, C.** (1979). *Use of watersheds in contour detection*. Paper presented at the Workshop on Image Processing, Rennes.
- Cerepi, A., Humbert, L., & Burlot, R.** (2001). Petrophysical properties of porous medium from petrographic image analysis data. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 187-188, 233-256.
- Chayes, F.** (1956). *Petrographic modal analysis*. New York: Wiley.
- De Keyser, T. L.** (1999). Digital scanning of thin sections and peels. *Journal of Sedimentary Research*, 69(4), 962-964.
- Deutsch, L. P., & Schiffman, A. M.** (1984). Efficient implementation of the smalltalk-80 system. *Proceedings of the 11th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, 297-302.
- Dickson, J. A. D.** (1966). Carbonate identification and genesis as revealed by staining. *Journal of Sedimentary Research*, 36(2), 491-505.
- Dunham, R. J.** (1962). Classification of carbonate rocks according to depositional texture. *Classification of Carbonate Rocks. AAPG Mem*, 1, 108-121.
- Ehrlich, R., Crabtree, S. J., Horkowitz, K. O., & Horkowitz, J. P.** (1991a). Petrography and reservoir physics i: Objective classification of reservoir porosity. *AAPG Bulletin*, 75(10), 1547-1562.
- Ehrlich, R., Etris, E. L., Brumfield, D., Yuan, L. P., & Crabtree, S. J.** (1991b). Petrography and reservoir physics, iii, physical models for permeability and formation factor. *AAPG Bull*, 75(10), 1579–1592.
- Flusser, J., & Suk, T. a. u.** (1993). Pattern recognition by affine moment invariants. *Pattern Recognition*, 26(1), 167-174.
- Folk, R. L.** (1951). A comparison chart for visual percentage estimation. *Journal of Sedimentary Research*, 21(1), 32-33.
- Folk, R. L.** (1974). *Petrology of sedimentary rocks*. Austin, Texas: Hemphill Publishing Company.
- Francus, P.** (2004). *Image analysis, sediments and paleoenvironments*. Dordrecht: Springer.
- Fredrich, J. T., & Evans, B.** (1990). Effect of grain size on brittle and semibrittle strength: Implications for micromechanical modelling of failure in compression. *Journal of Geophysical Research*, 95(B7), 10907-10920.

- Freeman, H.** (1961). On the encoding of arbitrary geometric configurations. *Transactions on electronic computers*, 10, 260-268.
- Freeman, H.** (1970). Boundary encoding and processing. *Picture Processing and Psychopictorics*, 241-266.
- Galehouse, J. S.** (1971). Point counting. *Procedures in Sedimentary Petrology*, 385–407.
- Gerard, R. E., Philipson, C. A., Manni, F. M., & Marschall, D. M.** (1992). Petrographic image analysis: An alternate method for determining petrophysical properties. *Automated pattern analysis in petroleum exploration table of contents*, 249-263.
- Gies, R. M.** (1987). An improved method for viewing micropore systems in rocks with the polarizing microscope. *Society of Petroleum Engineers Formation Evaluation*, 2, 209-214.
- Graham, R. L., & Yao, F. F.** (1983). Finding the convex hull of a simple polygon. *J. Algorithms*, 4(4), 324-331.
- Harrell, J. A., & Eriksson, K. A.** (1979). Empirical conversion equations for thin-section and sieve derived size distribution parameters. *Journal of Sedimentary Research*, 49(1), 273-280.
- Higgins, M. D.** (2000). Measurement of crystal size distributions. *American Mineralogist*, 85(9), 1105-1116.
- Hitzman, M. W.** (1999). Routine staining of drill core to determine carbonate mineralogy and distinguish carbonate alteration textures. *Mineralium Deposita*, 34(8), 794-798.
- Holmes, A.** (1921). *Petrographic methods and calculations*. London: Murby.
- Houghton, H. F.** (1980). Refined techniques for staining plagioclase and alkali feldspars in thin section. *Journal of Sedimentary Research*, 50(2), 629-631.
- Howard, C. V., & Reed, M. G.** (1998). *Unbiased stereology: Three-dimensional measurement in microscopy*. Oxford: BIOS Scientific Publishers.
- Hu, M. K.** (1962). Visual pattern recognition by moment invariants. *IEEE Trans. Infor. Theory*, 8, 179-187.
- Johnson, M. R.** (1994). Thin section grain size analysis revisited. *Sedimentology*, 41, 985-999.
- Kong, T. Y., & Rosenfeld, A.** (1989). Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3), 357-393.
- Koplowitz, J., & Bruckstein, A. M.** (1989). Design of perimeter estimators for digitized planar shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-11*(6), 611-622.
- Krumbein, W. C., & Pettijohn, S. J.** (1938). *Manual of sediment petrography*. New York: Appleton-Century Crofts.
- Laniz, R. V., Stevens, R. E., & Norman, M. B.** (1964). Staining of plagioclase feldspar and other minerals with fd and c. Red no. 2. *US Geol. Surv. Pap.* 501-B, B152-B153.
- Lowe, D. G.** (1989). Organization of smooth image curves at multiple scales. *International Journal of Computer Vision*, 3(2), 119-130.
- Lucia, F. J.** (1995). Rock-fabric/ petrophysical classification of carbonate pore space for reservoir characterization. *AAPG Bulletin*, 79(9), 1275-1300.
- MacQueen, J.** (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281-297.
- Mandarim-de-Lacerda, C. A.** (2003). Stereological tools in biomedical research. *Anais da Academia Brasileira de Ciencias*, 75, 469-486.
- Matheron, G.** (1967). *Elements pour une théorie des milieux poreux*. Paris: Masson.

- Meyer, F., & Beucher, S.** (1990). Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1), 21-46.
- Ndiaye, M.** (2002). Pétrographie quantitative: Un logiciel java facilitant le comptage par point: Université de Genève, travail de diplôme (non publié).
- Neilson, M. J., & Brockman, G. F.** (1977). The error associated with point-counting. *American Mineralogist*, 62(11-12), 1238-1244.
- Philpotts, A. R.** (1988). *Petrography of igneous and metamorphic rocks*. New York: Prentice Hall Englewood Cliffs.
- Pirard, E.** (2004). Multispectral imaging of ore minerals in optical microscopy. *Mineralogical Magazine*, 68(2), 323-333.
- Plas, L., & Tobi, A. C.** (1965). A chart for judging the reliability of point counting results. *American Journal of Science*, 263(1), 87-90.
- Prikryl, R.** (2001). Some microstructural aspects of strength variation in rocks. *International Journal of Rock Mechanics and Mining Sciences*, 38(5), 671-682.
- Ramer, U.** (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3), 244-256.
- Reed, S. J. B.** (1996). *Electron microprobe analysis and scanning electron microscopy in geology*. Cambridge: Cambridge University Press.
- Roduit, N.** (2001). Typification de la porosité des roches carbonatées à l'aide de l'analyse d'images et des réseaux neuronaux: Université de Genève, travail de diplôme (non publié).
- Rubin, D. M.** (2004). A simple autocorrelation algorithm for determining grain size from digital images of sediment. *Journal of Sedimentary Research*, 74(1), 160-165.
- Russ, J. C.** (1990). *Computer-assisted microscopy: The measurement and analysis of images*. New York: Plenum Press.
- Russ, J. C.** (1992). *The image processing handbook*. Boca Raton: CRC Press.
- Sahagian, D. L., & Proussevitch, A. A.** (1998). 3d particle size distributions from 2d observations: Stereology for natural applications. *Journal of Volcanology and Geothermal Research*, 84(3), 173-196.
- Sallier, B.** (2005). *Carbonates microporeux : Influence de l'architecture du milieu poreux et la mouillabilité sur les écoulements diphasiques dans les réservoirs pétroliers*. Université de Genève, Terre et Environnement, vol. 57, pp. 230.
- Saltykov, S. A.** (1967). The determination of the size distribution of particles in an opaque matrix from a measurement of the size distributions of their sections. *Stereology*: New York, Springer-Verlag, 163-173.
- Seul, M., O'Gorman, L., & Sammon, M. J.** (2000). *Practical algorithms for image analysis*. Cambridge: Cambridge University Press.
- Shannon, C. E.** (1948). A mathematical theory of communication, bell syst. *Tech. J*, 27(3), 379-423.
- Solymar, M.** (1998). Image analysis, and estimation of porosity and permeability of arnager greensand, upper cretaceous, denmark. In *23rd general assembly of the european geophysical society; part 1, society symposia, solid earth geophysics and geodesy*. (Vol. 1, pp. 256). Kathlenburg-Lindau: European Geophysical Society.
- Stokes, M., Anderson, M., Chandrasekar, S., & Motta, R.** (1996). A standard default color space for the internet—srgb. *Microsoft and Hewlett-Packard Joint Report, Version, 1*.
- Terry, R. D., & Chilingar, G. V.** (1955). Comparison charts for visual estimation of percentage composition. *J. Sedim. Petrol.*, 28(4), 229-234.
- Underwood, E. E.** (1970). *Quantitative stereology*. Massachusetts: Addison-Wesley.

- Van den Berg, E. H., Meesters, A. G. C. A., Kenter, J. A. M., & Schlager, W.** (2002). Automated separation of touching grains in digital images of thin sections. *Computers and Geosciences*, 28(2), 179-190.
- Vincent, L., & LizardTech, I.** (2000). *Fast granulometric methods for the extraction of global image information*. Paper presented at the 11th Annual Symposium of the South African Pattern Recognition Association, Johannesburg.
- Vossepoel, A. M., & Smeulders, A. W. M.** (1982). Vector code probability and metrification error in the representation of straight lines of finite length. *Computer Graphics Image Processing*, 20(4), 347-368.
- Weibel, E. R.** (1979). *Stereological methods: Practical methods for morphometry*. New York: Academic Press.
- Yang, L., Albregtsen, F., Llnnestad, T., & Grlttum, P.** (1994). Methods to estimate areas and perimeters of blob-like objects: A comparison. *Proc. IAPR Workshop on Machine Vision Applications*, 272-276.
- Zhang, G., Jayas, D. S., Karunakaran, C., & White, N.** (2005). Separation of touching grain kernels in an image by ellipse fitting algorithm. *Biosystems engineering*, 92(2), 135-142.