

LE TRAITEMENT D'IMAGES « POUR LES NULS »

D. Legland

12 juin 2019

Résumé *Ce document présente une introduction à l'analyse et au traitement d'images dans un cadre scientifique, en particulier pour les sciences du vivant. Le public visé est le scientifique (chercheur, ingénieur ou technicien) qui a besoin d'outils pour analyser des images acquises dans le cadre de ses recherches. Le niveau mathématique requis est volontairement gardé faible, et dans la mesure du possible les exemples seront donnés en utilisant des images issues de recherches concrètes menées à l'INRA.*

Table des matières

1. Introduction	8
I. Manipuler et explorer les images numériques	10
2. Qu'est ce qu'une image numérique ?	11
2.1. Discrétisation de l'espace image	12
2.2. Images en niveaux de gris	13
2.3. Images couleurs	14
2.4. Formats d'image	16
3. Explorer et visualiser des images	19
3.1. Visualisation des niveaux de gris	19
3.2. Profils d'intensité	21
3.3. Histogramme	22
3.4. Histogramme d'une image couleur	23
3.5. Calibration spatiale	25
3.6. Images 3D, 4D, 5D...	25
II. Les outils du traitement d'images	27
4. Les opérateurs du traitement d'image	28
4.1. Les différents types de filtres	28
4.2. Tables de correspondances (LUT)	28
5. Le filtrage linéaire	30
5.1. Principe des filtres linéaires	30
5.2. Autres filtres classiques	33
5.3. Traitement des images couleur	34
6. Morphologie Mathématique	37
6.1. Filtrage morphologique	37
6.2. Reconstruction morphologique	38
6.3. Minima et maxima	40

III. Le traitement des images	43
7. Filtrage et restauration d'images	44
7.1. Modification d'histogramme	44
7.2. Corrections géométriques	45
7.3. Soustraction du fond	45
8. Segmentation	48
8.1. Seuillage	48
8.2. Classification de pixels	50
8.3. Détection de contours (<i>edge detection</i>)	50
8.4. Segmentation de régions par ligne de partage des eaux	50
8.5. Détection de structures (feature detection)	53
9. Traitement des images binaires	54
9.1. Carte des distances	54
9.2. Nettoyage des résultats de segmentation	54
9.3. Codage des contours	56
9.4. Séparation de particules se recouvrant	56
9.5. Squeletisation	57
IV. L'analyse et l'exploitation des images	59
10. Analyse	60
10.1. Morphologie de particules	60
10.2. Analyse de particules en 3D	66
10.3. Microstructure	69
10.4. Analyse de texture	69
10.5. Graphe d'adjacence des régions	69
Bibliographie	70

Table des figures

2.1.	Représentation d'une image matricielle	11
2.2.	Positions des pixels dans le repère image	12
2.3.	Représentation d'une image en niveaux de gris	13
2.4.	Représentation d'une image binaire	14
2.5.	Décomposition d'une image couleur	15
2.6.	Représentation d'une image multi-canaux	16
2.7.	Artefacts de compression JPG	18
3.1.	Visualiser les niveaux de gris d'une image	19
3.2.	Image codée sur 4096 niveaux de gris	20
3.3.	Agrandissement de l'affichage	21
3.4.	Palettes de couleurs	22
3.5.	Profil d'intensité	23
3.6.	Histogramme des niveaux de gris	23
3.7.	Histogramme d'une image couleur	24
3.8.	Histogrammes 2D et 3D d'une image couleur	24
3.9.	Calibration spatiale d'une image	25
3.10.	Représentations par coupe et surfacique d'une image 3D	26
4.1.	Tables de correspondances (LUT)	29
5.1.	Lissage par filtre plat	31
5.2.	Gradients horizontal et vertical par filtre de Prewitt	32
5.3.	Détection des contours par Prewitt et Sobel	33
5.4.	Lissage par filtre gaussien	34
5.5.	Filtres min et max	35
5.6.	Filtre de dynamique locale	36
6.1.	Érosion et dilatation	38
6.2.	Ouverture et fermeture morphologiques	39
6.3.	Dilatation et fermeture sur une image de maïs	40
6.4.	Reconstruction morphologique en niveaux de gris	40
6.5.	Détection des maxima régionaux et étendus	41
7.1.	Extension de dynamique	44
7.2.	Égalisation d'histogramme	45
7.3.	Distorsions géométriques	46
7.4.	Correction du fond avec rolling-ball	47

7.5. Correction de fond par modélisation polynomiale	47
8.1. Segmentation par la méthode d'Otsu	48
8.2. Segmentation d'image couleur par seuillage HSV	50
8.3. Segmentation d'image couleur par k-means	51
8.4. Segmentation de cellules par ligne de partage des eaux	52
8.5. Détection de contours par ligne de partage des eaux	52
9.1. Carte des distances d'une image binaire	54
9.2. Ouverture par attribut sur une image étiquetée.	55
9.3. Suppression des particules touchant les bords.	56
9.4. Séparation de particules	57
9.5. Squeletisation d'une image binaire.	57
9.6. Différents algorithmes de squeletisation.	58
10.1. Mesure du périmètre	61
10.2. Diamètre de Féret	62
10.3. Ellipse d'inertie d'une particule	63
10.4. Diamètre géodésique	65
10.5. Compacité d'une particule	66
10.6. Caractéristique d'Euler-Poincaré 3D	68
10.7. Graphe d'adjacence des régions	70

1. Introduction

L'acquisition d'images numériques est une pratique très répandue dans le milieu scientifique pour observer des échantillons et illustrer des résultats. Les récents développements des techniques d'imagerie permettent maintenant d'acquérir des images à des échelles très variées (microscopie à force atomique par exemple), d'explorer des structures tridimensionnelles (IRM, Tomographie, microscopie confocale...), et même d'obtenir des informations localisées sur la composition interne des échantillons via l'imagerie hyperspectrale (Raman, micro-spectroscopie infrarouge...).

Une question qui se pose de plus en plus est d'utiliser ces images pour pouvoir mesurer de manière quantitative l'information observée, et ainsi utiliser l'image comme instrument de mesure. Dans un premier temps, il est possible de mesurer certains paramètres simples « à la main », tels que le nombre de cellules, le diamètre apparent des particules... Cette approche est vite limitée pour plusieurs raisons : pour un grand nombre d'images le temps nécessaire devient vite prohibitif, la répétabilité des mesures n'est pas assurée, et la manipulation de certaines images (3D, multispectrales...) est complexe. De plus, on observe parfois des différences (textures, couleur...) que l'on ne sait pas toujours quantifier « à l'œil ».

Le but de l'analyse d'images est de fournir des méthodes pour décrire de manière quantitative l'information fournie par les images numériques. Ses avantages sont nombreux :

- automatisation des traitements (gain de temps)
- répétabilité des résultats
- large choix d'outils

La mise en place d'une chaîne automatisée de traitements nécessite cependant de résoudre un certain nombre de problèmes. Par exemple, si on veut mesurer la taille et la forme de particules (cellules, tâches sur des feuilles...), il faut au préalable détecter ces particules, en tenant compte de la variabilité des conditions d'acquisition et des structures parasites.

Une chaîne de traitement classique comprend donc les étapes suivantes :

- **pré-traitement**, pour corriger les défauts dus à l'acquisition et re-normaliser les images
- **amélioration** de l'image, ou encore filtrage, pour mettre en valeur les informations utiles et réduire les informations parasites (le bruit)
- **segmentation** de l'image, pour identifier les structures d'intérêt (des particules, des régions homogènes...)
- **analyse** des particules ou des régions, pour quantifier la taille, la forme, ou la texture des objets d'intérêt

Le traitement d'images s'inscrit dans une chaîne de traitement plus large, qui comprend en amont la mise au point de l'acquisition (échantillonnage et préparation du matériel étudié, réglage des conditions d'acquisition, contrôle de l'éclairage...), et en aval une

analyse statistique des résultats obtenus.

Première partie

Manipuler et explorer les images numériques

2. Qu'est ce qu'une image numérique ?

On rencontre couramment deux types d'images : les images matricielles (ou bitmap, ou plus rarement raster), et les images vectorielles.

Les **images matricielles** consistent en un tableau à deux (ou plus) dimensions, dont chaque case contient la valeur d'un pixel (voir la figure 2.1). Le terme pixel est la contraction de « *picture element* ». Les images fournies par les systèmes d'acquisition (microscope, appareil photo, scanner à plat...) sont de ce type. Elles sont stockées typiquement dans des fichiers aux formats .tif, .bmp, .png, .jpg...

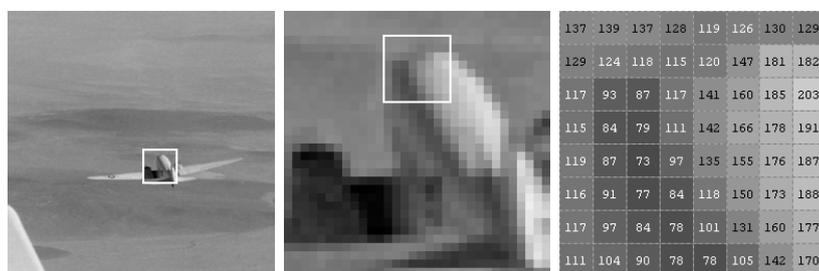


Figure 2.1. – Une image matricielle, agrandissement d'une zone de l'image, et affichage des valeurs constituant la matrice image.

Les **images vectorielles** représentent une image sous la forme d'une série de primitives géométriques : segment, point, cercle, polygone... Leur gros avantage est que contrairement aux images matricielles, on peut agrandir une image vectorielle autant que l'on veut sans perte de qualité. Elles sont souvent utilisées pour sauvegarder le résultat de graphiques ou de dessins techniques. Elles sont sauvées dans des formats postscript (ps, eps), .fig, .svg...

Dans le cadre du traitement d'images, on travaille exclusivement sur des images matricielles. Les images vectorielles ne sont utilisées que pour sauvegarder des graphiques de résultats. Par contre, on ne se limite pas forcément aux images à deux dimensions.

De manière un peu plus formelle, on peut voir une **image comme une application** (au sens mathématique du terme) qui à une position donnée p fait correspondre une valeur v . Pour des images planes, la position est un point du plan, repéré par deux coordonnées x et y . La valeur fournie est en général une intensité (lumineuse pour une caméra, de rayonnement pour un tomographe...), ou une information plus complexe (couleur, spectre...). La valeur d'un pixel peut être représentée par un vecteur de q valeurs, avec $q = 1$ pour une image d'intensité ou $q = 3$ pour une image couleur. Une image

2 Qu'est ce qu'une image numérique ?

idéalisée peut donc être vue comme une fonction de la forme :

$$F : \begin{cases} \mathbb{R}^n & \rightarrow & \mathbb{R}^q \\ p & \rightarrow & v = F(p) \end{cases} \quad (2.1)$$

En pratique, une image matricielle est soumise à deux types de **perte d'information** :

- une **discrétisation** de l'espace, qui ne mesure la valeur des pixels que pour un nombre fini de positions
- une **quantification** de la valeur des pixels en un nombre fini de valeurs, qui résulte de la conversion d'un signal analogique vers un signal numérique

2.1. Discrétisation de l'espace image

Dans les images matricielles 2D, les valeurs sont définies uniquement pour chaque élément de la matrice image. De manière traditionnelle, l'origine de l'image se situe en haut à gauche, en comptant les colonnes vers la droite et les lignes vers le bas, tous les deux en commençant à 0. Les valeurs disponibles sont donc de la forme :

$$F(x, y) \simeq I(u, v), u \in [0; M - 1], v \in [0; N - 1] \quad (2.2)$$

où u et v sont des valeurs entières représentant les indices des colonnes et des lignes, et M et N représentent la largeur et la hauteur de l'image (voir la figure 2.2).

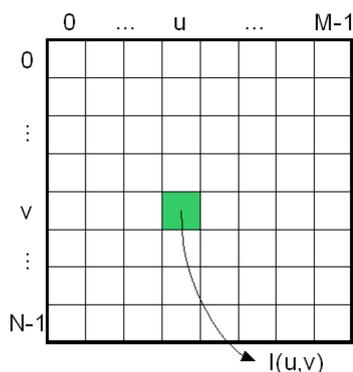


Figure 2.2. – Positions des pixels dans le repère image. Chaque pixel est repéré par l'indice de sa ligne (entre 0 et $N - 1$) et l'indice de sa colonne (entre 0 et $M - 1$)

La **résolution** d'une image spécifie la distance, en unités métriques, entre deux pixels. Pour les images destinées à l'impression, elle est souvent exprimée en points par pouce (*dots per inch*, ou DPI). En général, les pixels sont carrés et la résolution est la même pour les directions horizontales et verticales.

Les informations que l'on cherche à extraire des images étant souvent des paramètres de tailles ou de distances entre objets, il est important de disposer de la résolution pour l'analyse. Certains appareils d'acquisition enregistrent la résolution dans les meta-données au

sein des fichiers images, ou dans une base de données propre. Pour la visualisation, des barres d'échelle sont parfois incrustées de manière automatique dans l'image, mais leur exploitation automatique n'est pas facile. Enfin, si aucune information de résolution n'est disponible (par exemple pour les caméras), il est nécessaire d'utiliser des mires ou des points de repère de dimension connue.

2.2. Images en niveaux de gris

On peut classer les images en fonction de l'information contenue par chaque pixel.

2.2.1. Quantification des niveaux de gris

Les images les plus simples contiennent une valeur d'intensité, codée sur un nombre fini de **niveaux de gris** (on abrège parfois niveaux de gris en NDG). Le plus souvent, le nombre de niveaux de gris est codé sur 256 valeurs, le noir correspondant à la valeur 0 et le blanc à la valeur 255.

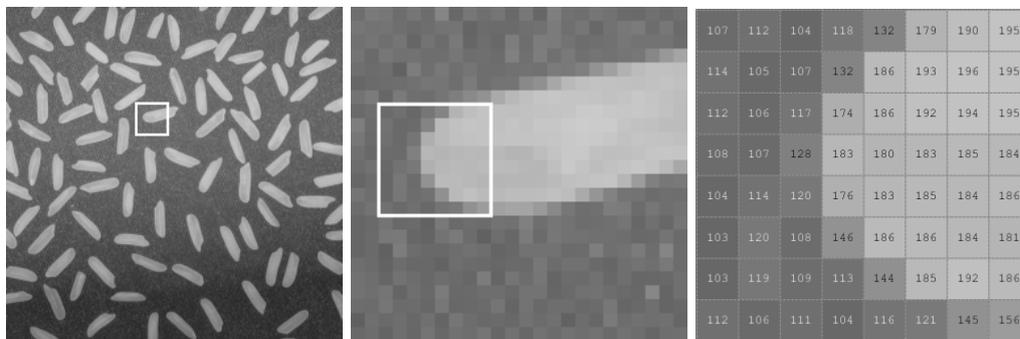


Figure 2.3. – Une image en niveaux de gris, agrandissement d'une zone de l'image, et affichage des valeurs constituant la matrice image.

Le nombre 256 vient de la représentation informatique des nombres entiers. L'unité de base manipulée par les ordinateurs est le **bit**, dont la valeur vaut soit 0 soit 1. Ces bits sont le plus souvent regroupés en **octets**, contenant 8 bits. Le nombre d'octets différents est égal à $2 \times 2 \times \dots \times 2 = 2^8 = 256$. En quantifiant les niveaux de gris sur un octet, on peut ainsi représenter 256 niveaux de gris différents.

Certains systèmes d'acquisition permettent d'acquérir des images avec une intensité codée sur 12, 14 ou 16 bits, permettant de représenter jusqu'à 4096, 16384 ou 65536 valeurs différentes. L'intérêt est de pouvoir détecter des variations d'intensité plus subtiles, mais leur visualisation nécessite des logiciels adaptés.

2.2.2. Images binaires

Les **images binaires** sont des images en niveau de gris particulières, dont les valeurs valent soit 0 (noir), soit 1 (blanc). On cherche souvent à se ramener les images de départ à des images binaires, car il est très facile ensuite d'identifier les objets présents (Figure 2.4).

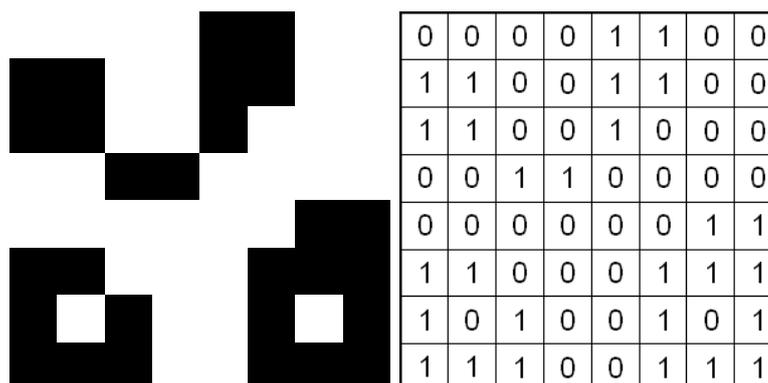


Figure 2.4. – Une image binaire, avec le tableau de valeurs correspondant. L'image est représentée en négatif (les valeurs 1 en noir, les valeurs 0 en blanc), afin de mieux visualiser les structures.

2.2.3. Images d'intensité

Il est possible de représenter la valeur des pixels avec des nombres à virgule flottante. Cela peut être utile si l'information stockée a une signification physique que l'on souhaite analyser (carte d'altitude, de profondeur, de densité...), ou pour stocker le résultat d'opérations arithmétiques sur plusieurs images. Par contre, les images en virgule flottante occupent plus de place en mémoire que les images en niveaux de gris. De plus, les valeurs minimales et maximales ne sont pas connues a priori, ce qui peut poser des problèmes d'affichage. On préfère donc en général stocker les résultats finaux en niveaux de gris.

2.3. Images couleurs

On rencontre plusieurs types d'images couleur, qui sont affichées de manière très similaire, mais dont la représentation interne est très différente.

2.3.1. Images en vraies couleurs

Les **images en vraies couleurs** stockent pour chaque pixel 3 composantes : rouge, vert, bleu (ou RGB, pour *Red-Green-Blue*), voir la figure 2.5. Ces trois composantes permettent de représenter toutes les autres couleurs, y compris les nuances de gris.

Si chaque composante est codée sur 8 bits (soit 256 valeurs possibles), on peut ainsi représenter jusqu'à $256 \times 256 \times 256 \simeq 16$ millions de couleurs différentes. On parle aussi d'images couleurs 24 bits. Pour le stockage, on rajoute parfois 8 bits supplémentaires. Cet ajout permet de coder chaque pixel sur un entier de 32 bits, qui est l'unité de calcul de base de nombreux ordinateurs, et d'extraire les composantes par des opérations opérant au niveau du bit.

Le traitement d'images sur des images couleurs est difficile. On utilise parfois des

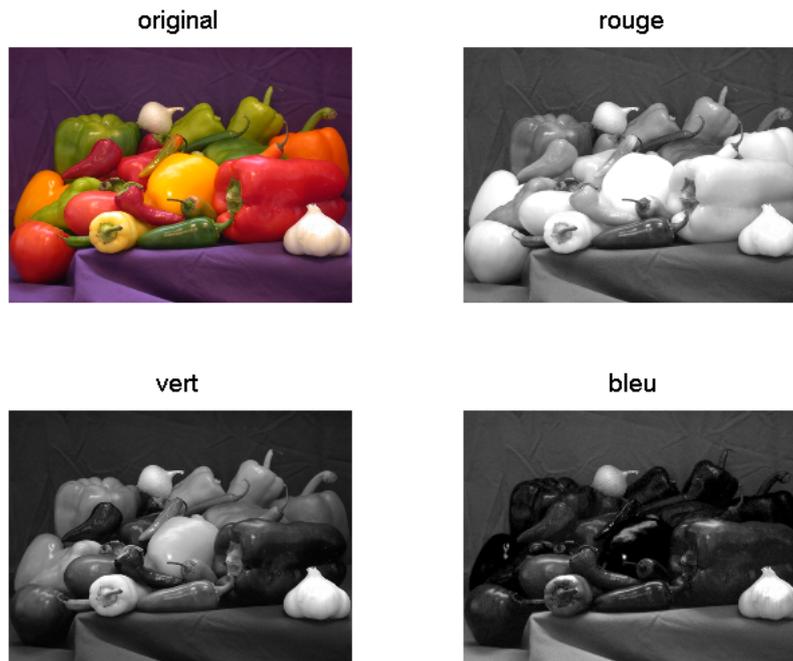


Figure 2.5. – Décomposition d'une image couleur en ses trois composantes rouge, verte et bleue.

espaces couleurs différents, qui sont plus adaptés que les composantes rouge-vert-bleu. L'espace HSV décompose chaque pixel en trois composantes de teinte (*hue*, qui représente la couleur principale), de saturation (dosage entre « coloré » et « gris »), et de valeur (qui correspond à l'intensité globale du pixel).

2.3.2. Images couleurs indexées

Les **images couleurs indexées** stockent pour chaque pixel un numéro de couleur (son index), qui fait référence à une couleur stockée séparément dans une palette.

L'intérêt de ces images est de réduire l'espace de stockage nécessaire, au prix d'une perte de qualité par rapport aux images en vraies couleurs. Par exemple, une image en 256 couleurs indexées occupera sensiblement la même place qu'une image en 256 niveaux de gris (la place occupée par la palette est négligeable par rapport à la taille de l'image).

2.3.3. Imagerie multi-canaux

En **microscopie multi-canaux**, on a l'habitude de représenter chaque canal avec une couleur différente. La représentation finale apparaît donc en couleur du fait de la combinaison des canaux associés aux couleurs rouge, verte et bleue (Figure 2.6).

Il faut bien garder à l'esprit que l'image ainsi obtenue est une image synthétique

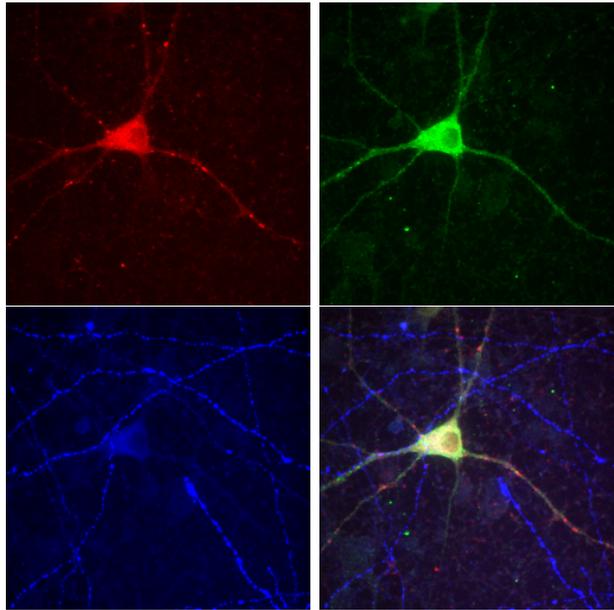


Figure 2.6. – Représentation d'une image multi-canaux en montrant soit chaque canal séparément, soit en combinant les canaux pour former une image en fausses couleurs.

obtenue en combinant plusieurs images d'intensité, mais que les couleurs ainsi obtenues sont totalement arbitraires, et dépendent entre autres choses de l'ordre des composantes.

2.3.4. Autres types de données

On distingue souvent les **images monomodales** et les **images multimodales**. Les pixels d'une image monomodale sont représentés par une valeur unique. C'est le cas des images en niveau de gris.

Les images multimodales peuvent être décomposées en plusieurs composantes monomodales. Par exemple, une image couleur peut être décomposée en 3 composantes rouge, verte, et bleu, tandis qu'une image complexe peut être décomposé en 2 composantes réelle et imaginaire. Chaque composante peut être étudiée ou travaillée séparément. On parle dans ce cas d'approche marginale. Il est souvent plus judicieux (et plus compliqué) de concevoir des traitements qui tiennent compte de l'ensemble des composantes.

2.4. Formats d'image

La sauvegarde des images sous forme de fichier nécessite de stocker non seulement la liste des valeurs des pixels, mais aussi la taille de l'image et la manière dont sont codés les pixels.

Les images étant des données volumineuses, on peut être tenté de les compresser. Cependant, les algorithmes de compression efficaces pour les images courantes en milieu

scientifique induisent une perte de données. Il vaut donc mieux stocker les images sous une forme non compressée, de préférence au format TIF.

2.4.1. Structure des fichiers

Un fichier image est en général composé de deux parties : un en-tête, et des données. L'exception concerne les fichiers raw, qui ne contiennent que les données brutes, et qui ne peuvent pas être lus sans connaître les conditions dans lesquelles a été acquise l'image.

L'**en-tête** contient la taille de l'image, le type de donnée (binaire, couleur, niveau de gris...), le nombre de bits utilisés pour le codage, l'ordre dans lequel sont stockés les valeurs, le type de compression utilisé... On peut parfois y trouver des informations utilisateurs (meta-données), telles que les données EXIF. Suivent ensuite les **données**, éventuellement compressées, qui ne peuvent être lues qu'une fois l'en-tête décodé.

2.4.2. Compression des images

La **compression** permet de transformer un ensemble de données de départ, correspondant aux valeurs des pixels, en un autre ensemble de données, de taille plus réduite, mais qui contient l'information de l'ensemble de départ. La **décompression** est l'opération de reconstruction des valeurs de départ à partir des données compressées. On distingue deux modes de compression : avec perte (ou destructif) et sans perte (ou non destructif).

Dans un mode de **compression sans perte**, l'image obtenue après décompression correspond exactement à l'image initiale. Plusieurs algorithmes existent, tels que le RLE (*Run-Length-Encoding*), qui détecte les plages de pixels de même valeur, ou le LZW, utilisé par les images GIF.

En pratique, ces types de compression sont plutôt efficaces pour des images avec beaucoup de **redondances** (beaucoup de zones avec des couleurs uniforme). Les images scientifiques, au même titre que les photographies, contiennent très peu de redondances, et les algorithmes de compression sans perte sont donc assez peu efficaces.

Dans un mode de **compression avec perte**, on accepte que l'image obtenue après décompression présente quelques différences avec l'image initiale. Les algorithmes utilisent pour cela une décomposition de l'image selon les composantes fréquentielles (compression JPEG) ou basée sur des ondelettes (JPEG2000), et ne conservent que les composantes principales de la décomposition. Il en résulte une détérioration de l'image reconstruite (voir la figure 2.7).

Il va de soi que si l'on désire exploiter l'information stockée dans les images, il faut privilégier les modes de compression sans perte.

2.4.3. Quelques formats

Il existe une grande quantité de formats de fichier d'image, mais seuls quelques-uns sont réellement intéressants en pratique.

- Tagged Image File Format (TIFF) c'est le format de prédilection pour sauvegarder des images scientifiques. Il permet de stocker des images en niveaux de gris, en

2 Qu'est ce qu'une image numérique ?



Figure 2.7. – Une image sauvée selon différents modes de compression : sans compression, jpg 40% et jpg 20%

- couleur, binaire, compressée (sans perte) ou non, ainsi que des données utilisateurs (les tags). En général, les microscopes sauvent les conditions d'acquisition.
- Bitmap (BMP) ce format est très répandu sous Windows, moins sous Linux. Il permet de compresser sans pertes des images en couleurs ou en niveaux de gris.
 - Portable Network Graphics (PNG) il s'agit d'un format plus standard que le bitmap, qui autorise plus de possibilité de compressions, mais qui reste plus adapté pour les images de type dessin
 - Jpeg (JPG) ce format est beaucoup utilisé pour stocker les photos, mais il compresse les images avec pertes. Il est à proscrire pour l'analyse d'images
 - Portable bit map (PBM), portable gray map (PGM) et portable pixel map (PPM) : ces 3 trois formats stockent les valeurs des pixels dans un fichier texte. L'avantage est qu'un être humain peut lire directement le fichier, mais la taille de ces fichiers est beaucoup plus importante que les autres formats.

Attention enfin aux fausses images en niveaux de gris, qui stockent une valeur d'intensité identique pour les trois canaux rouge vert et bleu. L'image s'affiche en niveaux de gris, mais occupe trois à quatre fois plus de place qu'une vraie image en niveaux de gris...

3. Explorer et visualiser des images

Bien que la plupart des images s'affichent correctement avec le premier logiciel venu, il peut arriver que le résultat ne soit pas celui attendu. De plus, pour certaines images particulières (images 3D, vidéos, images multispectrales...), ou bien si on veut examiner de très près les images (connaître la valeur en niveau de gris moyenne dans une région), les applications classiques de visualisation ne suffisent plus. Cette section a donc pour but de donner quelques clés pour éviter les mauvaises surprises.

3.1. Visualisation des niveaux de gris

3.1.1. Images en niveaux de gris

À l'échelle 1, les images en niveau de gris sont affichées en faisant correspondre chaque pixel de l'image à un pixel de l'écran. La convention générale est de représenter une faible valeur d'intensité par un pixel sombre, et une valeur élevée par un pixel clair (Figure 3.1).

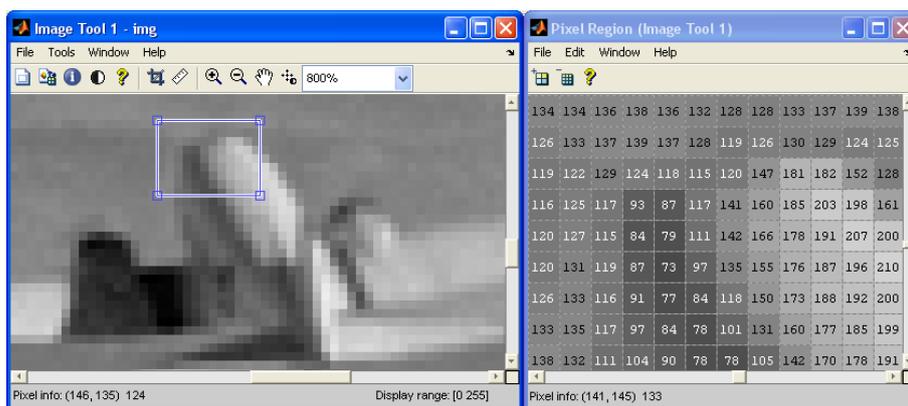


Figure 3.1. – Visualisation d'une image en niveaux de gris, ainsi que des valeurs d'intensité correspondantes, sous Matlab.

3.1.2. Normalisation des intensités

Les images en niveaux de gris sont la plupart du temps stockées sur 256 niveaux de gris, allant de 0 (représenté par du noir) à 255 (représenté par du blanc). Certains logiciels permettent d'inspecter directement la valeurs d'intensité des niveaux de gris, comme c'est le cas sur la figure 3.1.

Dans le cas d'images en niveaux de gris codées sur plus de niveaux, les niveaux sont ramenés entre 0 et 255 pour l'affichage. Si l'affichage n'est pas celui attendu, il se peut que le problème vienne de cette normalisation. Une solution est de convertir soi-même les niveaux de gris à l'aide d'opérations arithmétiques élémentaires (figure 3.2). Quelques méthodes de normalisation sont décrites dans la section 7.1.

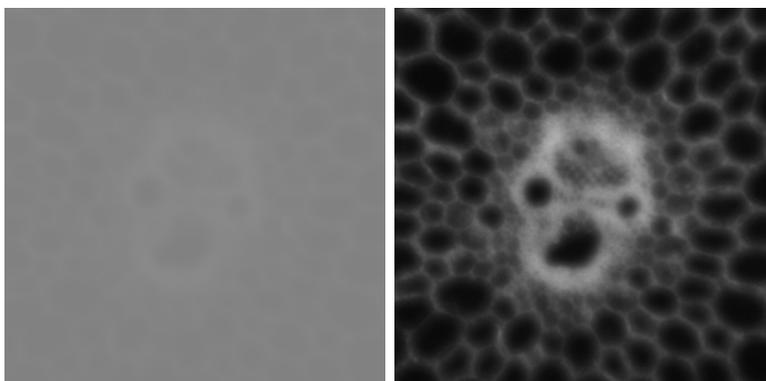


Figure 3.2. – Une image codée sur $2^{12} = 4096$ niveaux de gris, et sa représentation sous Matlab avant et après normalisation des valeurs entre 0 et 255.

3.1.3. Blanc sur noir ou noir sur blanc ?

Pour les images d'intensité codées en nombres à virgule flottante, le comportement par défaut des logiciels est de calculer les valeurs minimum et maximum, et de normaliser les valeurs par rapport à ces extrêmes. Parfois (par exemple sous Matlab), la normalisation se fait par défaut entre 0 et 1, ce qui peut ne pas correspondre au résultat attendu.

Pour les images binaires, on suit en général la même convention que pour les images en niveau de gris : la valeur 0 est représenté par du noir, et la valeur 1 par du blanc. Ce n'est pas toujours le cas : le logiciel ImageJ, par exemple, utilise par défaut la convention inverse.

Pour les documents imprimés, ou pour la présentation des résultats, on préfère souvent inverser la convention. Cela a plusieurs avantages : les structures fines sont plus visibles quand elles sont affichées en noir sur fond blanc, le résultat est (souvent) plus esthétique, et enfin cela évite de vider la cartouche d'encre de l'imprimante... Il convient donc d'être particulièrement attentif à la convention utilisée lorsque l'on travaille sur des images binaires ou d'intensité.

3.1.4. Agrandissement de l'affichage

Quand on agrandit l'affichage de l'image, on doit voir apparaître les pixels individuels. Ce n'est pas le cas de tous les logiciels de visualisation. Par exemple, en utilisant la fonction d'aperçu fournie avec Windows, un lissage est appliqué automatiquement, ce qui peut tromper l'utilisateur sur la résolution effective de l'image (voir la figure 3.3).

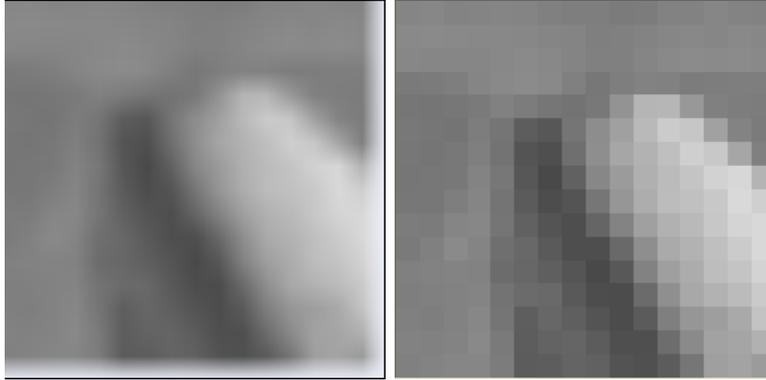


Figure 3.3. – La même portion d'image agrandie avec l'aperçu des photos de Windows (à gauche), et IrfanView (à droite). Malgré l'aspect lisse de l'image de gauche, elle n'est pas de meilleure qualité que l'image de droite.

3.1.5. Palettes de couleurs (tables de correspondance)

Les **palettes de couleurs**, ou **tables de correspondance** (en anglais : *Look-Up Table*, abrégé en LUT) consistent à représenter chaque valeur de l'image non pas avec la valeur de gris correspondante, mais avec une couleur spécifique stockée dans une table. L'avantage est que certaines structures peuvent être mieux mises en évidence, du fait du nombre plus important de couleurs utilisé (Figure 3.4).

Il faut cependant garder à l'esprit que l'image d'origine n'est pas modifiée : c'est uniquement sa représentation qui change.

Quelques palettes régulièrement utilisées :

hsv dégradé passant par le rouge, le vert, le bleu, puis de retour au rouge

jet dégradé allant du bleu foncé (couleur froide) au rouge foncé (couleur chaude), en passant par le vert (couleur neutre)

hot dégradé du noir au jaune

inversion dégradé de niveaux de gris allant du blanc au noir, ce qui permet de représenter l'image « en négatif »

bleu+gris+rouge dégradé de niveaux de gris, mais la valeur 0 est affichée en bleu, et la valeur 255 est affichée en rouge, afin de mettre en évidence les zones trop ou trop peu exposées

3.2. Profils d'intensité

Un outil interactif très intuitif pour visualiser les variations d'intensité dans une image est le profil des intensités. Il faut tout d'abord choisir un outil de sélection linéaire (segment, droite polygonale...), et tracer la région d'intérêt désirée. Le profil d'intensité permet de visualiser les variations d'intensité le long de la sélection (Figure 3.5).

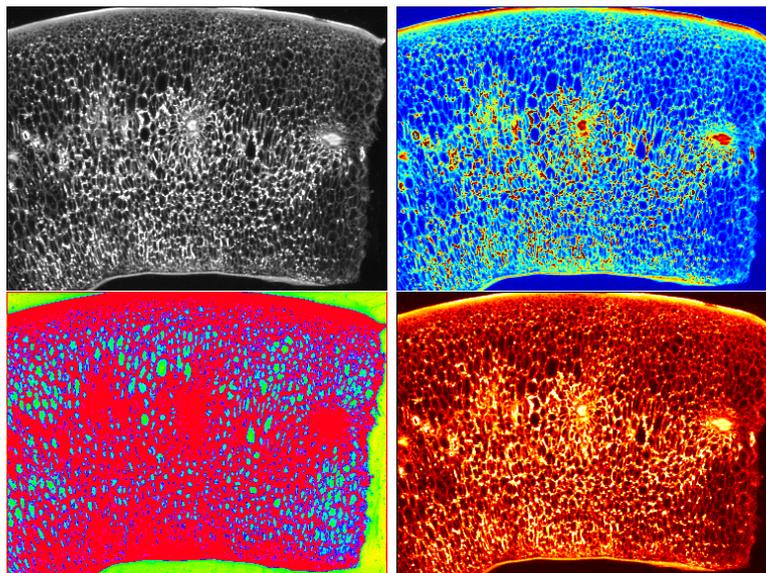


Figure 3.4. – Représentation d’une image en niveaux de gris en utilisant différentes palettes de couleur. De gauche à droite et de haut en bas : niveaux de gris naturels, palettes ‘jet’, ‘hsv’ et ‘hot’ de Matlab.

Une petite précaution est à prendre pour l’interprétation des valeurs : le profil est généralement calculé en interpolant les valeurs. On peut donc avoir des valeurs flottantes à partir d’images en niveaux de gris codés sur des entiers, ce qui peut être trompeur.

Pour des images couleurs, le comportement par défaut de certains logiciels est souvent d’afficher le profil d’intensité après conversion des couleurs. Pour ImageJ, le plugin « RGB Profile Plot » permet de pallier ce manque (Figure 3.5).

3.3. Histogramme

L’histogramme d’une image est une application qui à chaque valeur de niveau de gris dans l’image associe le nombre de pixels ayant cette valeur. Pour une image en 256 niveaux de gris, le résultat de l’histogramme sera une liste de 256 valeurs.

La figure 3.6 représente une image en niveaux de gris de coupe de tige de maïs (comportant des cellules et des faisceaux de xylème et phloèmes), et l’histogramme qui lui correspond. Le pic le plus à gauche de l’histogramme correspond aux pixels du fond (de couleur noire). Les valeurs de pixels 200 à 250 correspondent aux structures claires, tandis que les valeurs centrales (autour de 50 à 100) représentent les parois des cellules.

L’histogramme est un outil rapide permettant de vérifier la qualité de l’image :

- image globalement surexposée (trop claire) ou sous-exposée (trop sombre)
- histogramme idéalement centré, et le plus étalé possible
- présence ou absence de zones ayant un niveau de gris homogène

Quelques statistiques de base permettent de synthétiser les informations sur l’image :

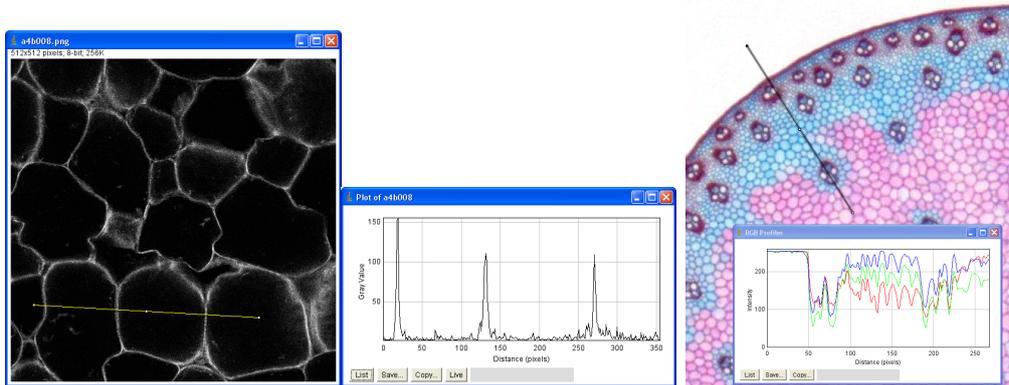


Figure 3.5. – Profil d'intensité sur une image en niveaux de gris, et sur une image couleur.

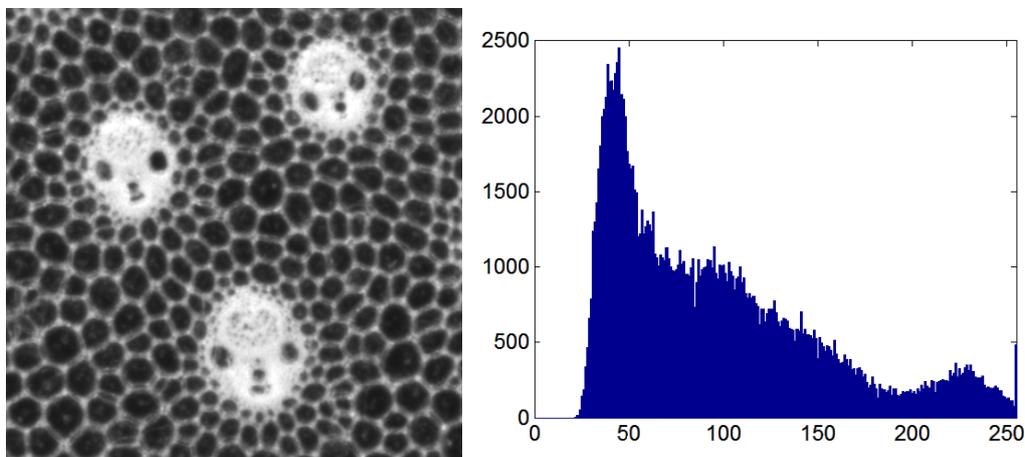


Figure 3.6. – Exemple d'image en niveau de gris représentant des cellules végétales, et l'histogramme des niveaux de gris correspondant.

valeur moyenne de l'histogramme, position du mode (la valeur maximale), écart-type...

3.4. Histogramme d'une image couleur

Du fait de la nature multi-canal des images couleur, la représentation sous forme d'histogramme est plus compliquée. Une solution simple pour explorer la distribution des couleurs est de représenter l'histogramme des valeurs de chaque canal séparément (traitement marginal) (Fig. 3.7-B). L'inconvénient de ce type de représentation est que l'on perd l'information sur la corrélation des composantes.

Afin de représenter les dépendances entre deux composantes colorimétriques, une solution est de passer par un histogramme conjoint, qui représente le nombre ou la fréquence des pixels pour chaque couple de valeurs dans deux composantes. La figure 3.8A repré-

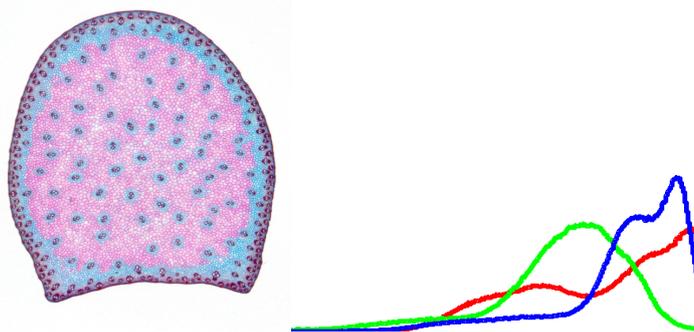


Figure 3.7. – Exemple d’image couleur représentant une coupe de tige de maïs colorée au FASGA, et histogramme des valeurs de chaque canal.

sente ainsi l’histogramme conjoint des canaux rouge et bleu pour l’image représentée sur la figure 3.7-A. On distingue plusieurs grappes de pixels de couleur proches, qui seraient plus difficiles à détecter sur un histogramme simple :

- la partie centrale de l’histogramme (valeurs moyennes de rouge et de bleu) correspond aux pixels magenta sombre, situés principalement dans l’écorce,
- le coin supérieur gauche de l’histogramme (valeurs élevées à la fois de bleu et de rouge) correspond aux pixels magenta clair, prépondérants dans le parenchyme,
- la partie supérieure centrale de l’histogramme (valeurs élevées de bleu, moyennes de rouge) correspond aux pixels situés dans la zone bleue (en périphérie du parenchyme et autour des faisceaux).

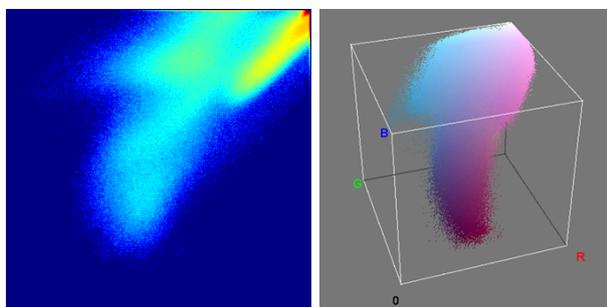


Figure 3.8. – Représentations multidimensionnelles de la distribution des couleurs dans l’image 3.7. A) histogramme conjoint des canaux rouge et bleu, B) représentation 3D de la distribution des couleurs dans l’espace RGB.

La démarche peut être poussée plus loin en calculant un histogramme 3D dans un espace colorimétrique donné, typiquement l’espace RGB (3.8). Avec une coloration appropriée, on retrouve intuitivement les colorations observables sur l’image d’origine. Par contre, l’exploitation de ce type de représentation nécessite d’utiliser des outils statistiques plus évolués (clustering, réduction de dimension...).

3.5. Calibration spatiale

À partir du moment où l'on veut mesurer des paramètres morphologiques (longueur, distance, surface...) dans des images, il est nécessaire de connaître la résolution de l'image. Heureusement, la plupart des logiciels de traitement permettent de calibrer les images en spécifiant leur résolution.

La méthode la plus simple est de fournir cette résolution manuellement, si elle est connue. L'autre solution est de choisir dans l'image un point de repère de longueur connue, de mesurer sa longueur en nombre de pixels, et d'effectuer la mise à l'échelle (figure 3.9).

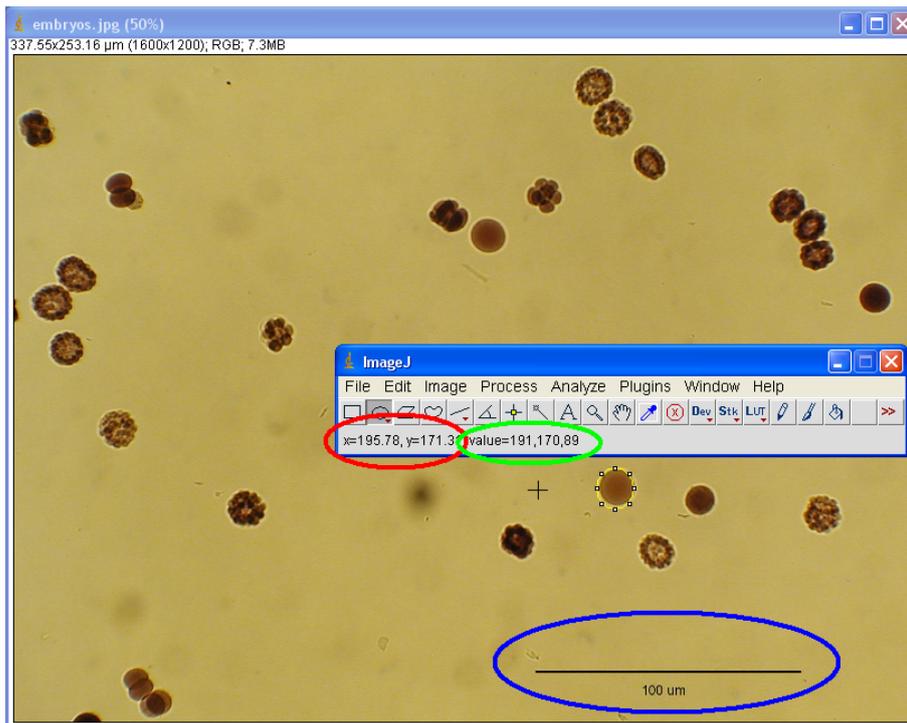


Figure 3.9. – Affichage d'une image après sa calibration sous ImageJ. La calibration s'est faite à l'aide de la barre d'échelle en bas à droite. La position du pixel est donnée en microns, et les valeurs de rouge, vert bleu correspondantes sont données dans la fenêtre principale.

3.6. Images 3D, 4D, 5D...

De plus en plus d'appareils d'acquisition produisent des images en 3 dimensions : IRM, tomographie, microscopie confocale... Les valeurs de l'image sont stockées dans des tableaux à 3 dimensions, et les éléments sont appelés des *voxels* (pour *volume element*).

Si les images sont acquises à des temps différents, on obtient des films, où le temps

joue le rôle d'une troisième ou quatrième dimension. Pour les images hyperspectrales, on peut encore rajouter une dimension supplémentaire. On arrive donc à des images allant jusqu'à 5 dimensions, ce qui rend parfois leur exploration problématique.

3.6.1. Visualisation plan par plan

Le mode de visualisation le plus simple est de fixer les dimensions supérieures à la deuxième, et de représenter une « coupe 2D » de l'image. Le logiciel ImageJ permet ainsi de représenter une image 2D choisie dans une pile en faisant varier le canal ou la dimension en z (Figure 3.10).



Figure 3.10. – Représentation d'une image 3D en visualisant chaque coupe individuelle, ou en appliquant une reconstruction surfacique.

3.6.2. Représentation volumique

La reconstruction volumique montre l'image en perspective, en remplissant les voxels de l'image de manière plus ou moins dense en fonction de leur intensité. Il est ainsi possible de localiser les zones de l'image regroupant les voxels les plus lumineux.

3.6.3. Reconstruction surfacique

Pour des images bien contrastées et relativement peu bruitées, il est possible de fixer une valeur seuil, et de représenter la surface équivalente qui entoure tous les pixels d'intensité inférieure (ou extérieure) à ce niveau. On obtient une représentation sous forme de maillage surfacique, que l'on peut visualiser plus facilement en 3D, éventuellement en superposant un ensemble de coupes de l'image d'origine (Figure 3.10).

L'algorithme le plus souvent employé pour cette tâche est celui des « Marching Cubes » [3], ou une de ses variantes.

Deuxième partie

Les outils du traitement d'images

4. Les opérateurs du traitement d'image

Le but du filtrage d'image est d'améliorer leur aspect, en rendant plus visibles les structures d'intérêt, tout en réduisant l'influence des informations parasites (bruit de l'image). On utilise pour cela différents **filtres**, qui vont transformer l'image de départ en une image filtrée, plus facile à analyser.

4.1. Les différents types de filtres

On peut classer les filtres en fonction de la quantité d'information nécessaire pour calculer la valeur d'un pixel de l'image résultat. Certains filtres appliquent simplement une fonction de conversion à la valeur de **chaque pixel**. C'est le cas des filtres qui modifient l'histogramme, ou des filtres de binarisation.

D'autres filtres opèrent sur un **voisinage local** de chaque pixel. Pour chaque pixel, le résultat du filtre ne dépendra que des valeurs des pixels voisins. Par exemple, une opération courante consiste à remplacer la valeur de chaque pixel par la valeur moyenne de ses voisins, ce qui a pour effet de réduire le bruit dans l'image. Une autre possibilité consiste à calculer la différence entre les valeurs des pixels du centre et de la périphérie du voisinage. L'effet sera alors de mettre en évidence les contours dans l'image.

Le résultat du filtrage dépendra en grande partie de la taille du voisinage : plus le voisinage est grand, et plus le résultat sera différent de l'image de départ. Il est aussi possible d'utiliser des formes de voisinage différentes. Les plus classiques sont les voisinages carrés de taille 3×3 , 5×5 ... On peut préférer des voisinages en forme de disque, qui traitent toutes les directions de manière identique, au contraire des formes carrées. À l'inverse, on peut utiliser des voisinages en forme de segment de droite, pour détecter ou supprimer des formes fines.

Enfin, d'autres filtres ont besoin de la **totalité de l'image** pour calculer le résultat en un point. C'est le cas des filtres de reconstruction morphologique, de seuillage par hystérésis, ou de la ligne de partage des eaux.

4.2. Tables de correspondances (LUT)

On peut modifier une image en calculant pour chaque pixel une nouvelle valeur $I'(u, v)$ qui ne dépend que de la valeur de départ $g = I(u, v)$ du pixel

$$\begin{aligned} g' &\leftarrow f(g), \text{ ou} \\ I'(u, v) &\leftarrow f(I(u, v)) \end{aligned}$$

Le nombre de valeurs de départ étant fixe (256 pour les images courantes), on peut précalculer les valeurs finales et stocker le résultat dans un tableau, appelé **table de correspondance**, ou *look-up table* (LUT).

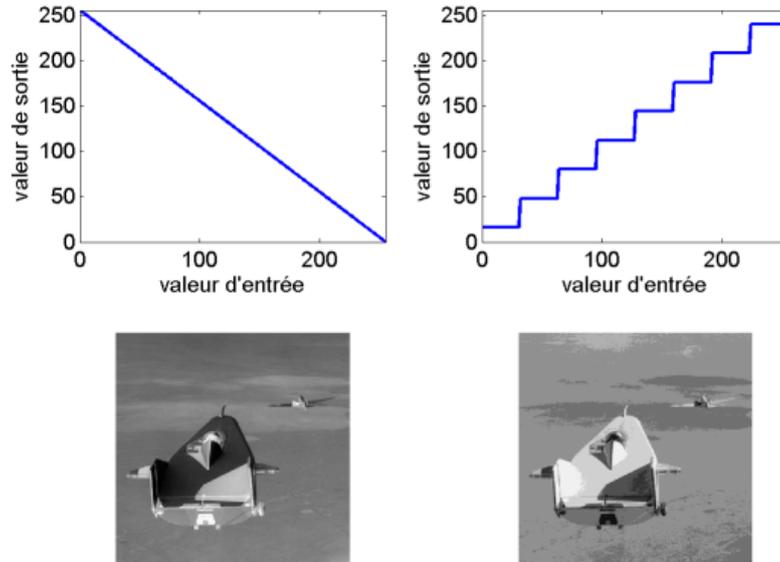


Figure 4.1. – Application de deux tables de correspondances (LUT) sur l'image de départ : une inversion, et une re-quantification sur 8 niveaux de gris.

La figure 4.1 montre deux LUT, et le résultat de leur application sur l'image de départ.

5. Le filtrage linéaire

5.1. Principe des filtres linéaires

Les **filtres linéaires** sont une classe particulière de filtres, dont les propriétés mathématiques les rendent intéressants à utiliser.

Dans le cas d'un filtrage linéaire de l'image, les opérations impliquées ne sont que des additions (ou soustractions) et multiplications (ou divisions) par des valeurs constantes. En pratique, cela consiste à réaliser une moyenne pondérée de la valeur des pixels du voisinage. Par exemple, une opération courante consiste à remplacer la valeur de chaque pixel par la valeur moyenne de ses voisins. La nouvelle valeur du pixel $I(u, v)$ est déterminée par l'équation :

$$I'(u, v) \leftarrow 1/9[I(u-1, v-1) + I(u, v-1) + I(u+1, v-1) \\ + I(u-1, v) + I(u, v) + I(u+1, v) \\ + I(u-1, v+1) + I(u, v+1) + I(u+1, v+1)] \quad (5.1)$$

On peut classer les filtres linéaire en fonction de leur effet sur l'image de départ. Les **filtres moyenneurs** réduisent le bruit tout en lissant les structures. Les **filtres dérivatifs** permettent de mettre en évidence les bords des structures, et sont très utilisés pour détecter les contours. Les **filtres dérivatifs du second ordre** sont une alternative aux filtres dérivatifs simples, et permettent aussi de mettre en évidence les contours.

5.1.1. Filtres moyenneurs

Les **filtres moyenneurs**, comme leur nom l'indique, calculent la moyenne, éventuellement pondérée, des pixels situés dans le voisinage de chaque pixel. Cette famille de filtres permet de réduire le bruit dans l'image, ce qui rend les zones homogènes plus lisses. Par contre, les contours sont fortement dégradés, et les structures trop fines peuvent devenir moins visibles.

Filtres plats

Les filtres qui calculent une moyenne directe des pixels sont appelés **filtres plats** (*flat smoothing*). Ils utilisent en général des voisinages carrés (3×3 , 5×5 ...). Pour un filtre plat de taille 3×3 , la matrice de voisinage correspondante est la suivante :

$$H = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Pour des filtres plats de taille $n \times n$, la matrice est de la forme :

$$\frac{1}{n^2} \times \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

Le résultat d'un filtre plat avec trois tailles de filtre différentes est donné sur la figure 5.1.

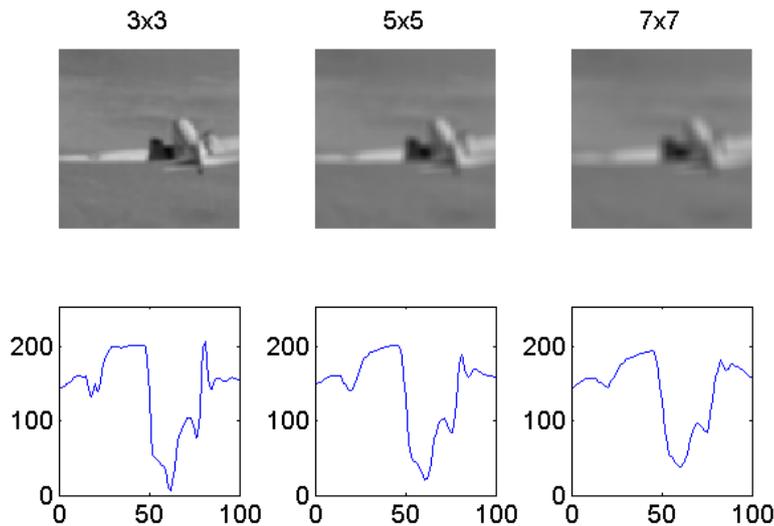


Figure 5.1. – Influence de la taille du voisinage sur le résultat d'un filtre plat. En dessous, on montre les profils d'intensité des niveaux de gris le long d'une ligne horizontale passant au niveau de l'aile de l'avion.

Moyennes pondérées

On peut vouloir pondérer le poids associé à la valeur de chaque voisin. En particulier, on peut supposer que l'influence d'un voisin proche est plus importante que celle d'un voisin éloigné. On utilise donc aussi des filtres pondérés selon la distance au pixel central :

$$H = \begin{bmatrix} 0.05 & 0.15 & 0.05 \\ 0.15 & 0.20 & 0.15 \\ 0.05 & 0.15 & 0.05 \end{bmatrix} = \frac{1}{20} \begin{bmatrix} 1 & 3 & 1 \\ 3 & 4 & 3 \\ 1 & 3 & 1 \end{bmatrix}$$

Les pondérations les plus classiques sont d'utiliser un noyau gaussien ou parabolique. Pour un noyau gaussien, les coefficients sont calculés à partir de la distance au pixel central du noyau du filtre :

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Un noyau discret 5×5 approchant un filtrage par un noyau gaussien avec $\sigma = 2$ est le suivant :

$$H = \frac{1}{101} \begin{bmatrix} 1 & 3 & 4 & 3 & 1 \\ 3 & 5 & 7 & 5 & 3 \\ 4 & 7 & 9 & 7 & 4 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 4 & 3 & 1 \end{bmatrix}$$

5.1.2. Filtres dérivatifs

Dans une image en niveaux de gris, les contours des structures d'intérêt peuvent être mis en évidence en utilisant des filtres linéaires qui calculent les variations de niveaux de gris. Pour cela on utilise des noyaux qui calculent la différence entre les niveaux de gris dans une direction, et les niveaux de gris dans la direction opposée. Les **filtres de Prewitt**, par exemple, sont de la forme :

$$P_{0^\circ} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, P_{90^\circ} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (5.2)$$

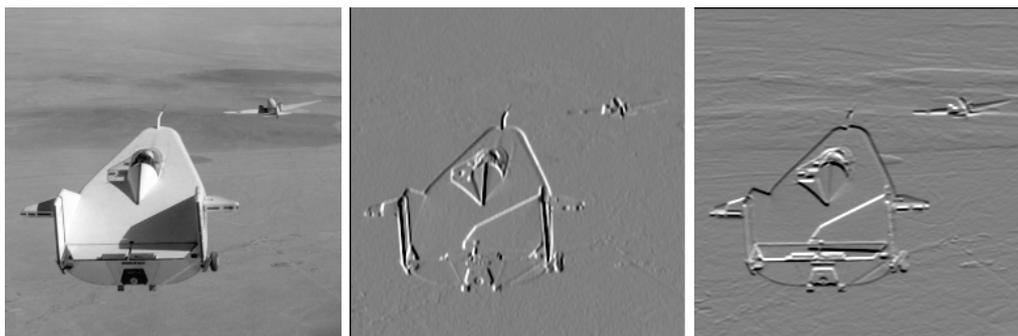


Figure 5.2. – Une image en niveaux de gris, et le résultat d'un filtre de Prewitt dans les directions horizontale et verticale. Les images filtrées ont été renormalisées entre 0 et 255.

Le résultat de leur application sur une image en niveaux de gris est donné figure 5.2. Un filtre assez similaire est le **filtre de Sobel**, qui donne un poids plus important aux pixels plus proches du pixel central :

$$S_{0^\circ} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_{90^\circ} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (5.3)$$

Il est aussi possible de considérer les **directions diagonales**, en orientant différemment les coefficients :

$$P_{45^\circ} = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, P_{135^\circ} = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} \quad (5.4)$$

L'image finale des contours est obtenue en combinant les résultats des filtres Prewitt ou Sobel dans plusieurs directions. Une combinaison pratique est de calculer le module des directions orthogonales (Figure 5.3), pour obtenir une **image gradient** :

$$I' = \sqrt{P_{0^\circ}(I)^2 + P_{90^\circ}(I)^2}$$

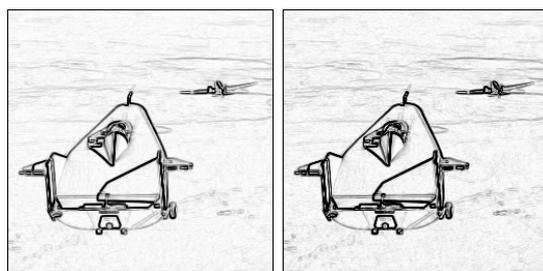


Figure 5.3. – Résultat de la détection des contours en utilisant les filtres de Prewitt (à gauche) ou de Sobel (à droite), et en calculant le module sur les directions horizontales et verticales.

5.2. Autres filtres classiques

5.2.1. Filtres d'ordre

Les **filtres d'ordre** considèrent l'ensemble des pixels du voisinage, classés par ordre croissant d'intensité, et sélectionnent le i -ième pixel de la liste pour l'affecter au résultat. Les filtres les plus utilisés de cette famille sont les filtres min, max, et médian.

Le **filtre médian**, comme son nom l'indique, sélectionne la valeur médiane des pixels du voisinage. Pour un voisinage carré 5×5 , par exemple, les valeurs des 25 voisins sont ordonnées par ordre croissant, et la 13-ième valeur est gardée comme résultat.

L'avantage du filtre médian est qu'il permet de réduire fortement le bruit dans l'image, et qu'il préserve beaucoup mieux les bords que les filtres moyennés.

Les **filtres minimum et maximum** ordonnent l'ensemble des pixels du voisinage, et sélectionnent soit la plus petite ou la plus élevée. Cette famille de filtre permet de supprimer des petits détails très lumineux ou très sombres, mais affecte fortement la taille des objets.

Le **filtre de dynamique locale** (« *range filter* ») permet de quantifier si un pixel est dans une zone homogène ou non. Il se calcule à partir de la différence entre les filtres minimum et maximum. Il donne des résultats souvent assez similaires à ceux du gradient. Un avantage est que les résultats peuvent être codés avec le même type de données que

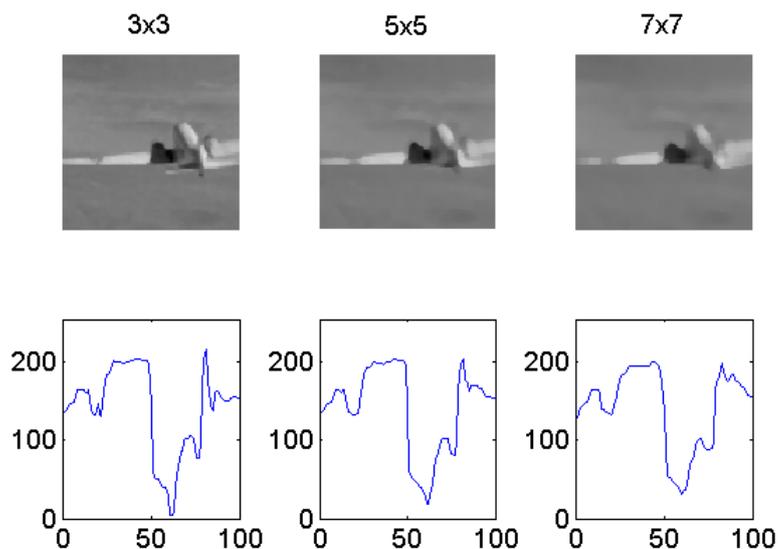


Figure 5.4. – Influence de la taille du voisinage sur le résultat d’un filtre médian. En dessous, on montre les profils d’intensité des niveaux de gris le long d’une ligne horizontale passant au niveau de l’aile de l’avion. À comparer avec la figure équivalente pour les filtres moyenneurs (Figure 5.1).

l’image d’origine, alors que le gradient nécessite en général de passer par des nombre à virgule flottante.

Il existe de nombreux autres filtres de réduction du bruit dans les images numériques. Parmi les plus connus, on peut citer les filtres de Nagao, de Kuwahara, ou encore le filtre bilatéral.

5.3. Traitement des images couleur

Le traitement des images couleurs pose parfois quelques difficultés, du fait que chaque pixel est représenté non pas par une valeur, mais par un ensemble de valeurs. Le même problème se rencontre pour le traitement des images multicanaux ou multispectrale.

Une solution radicale consiste à **convertir l’image en niveaux de gris**, et à ne travailler que sur l’image d’intensité. Cette approche présente l’inconvénient de perdre la richesse de l’information couleur, mais est parfois la plus rapide à mettre en œuvre.

Une approche plus satisfaisante est l’**approche marginale**, qui consiste à séparer les différents canaux, et à travailler séparément sur chacun des canaux qui peuvent être vus comme des images en niveaux de gris. Il est parfois préférable de d’abord **changer l’espace de représentation des couleurs** afin de mieux séparer les différentes informations. En particulier, les espaces HSV ou Lab se révèlent souvent très utiles. Les résultats des traitements sur les différents canaux peuvent ensuite être recombinaés, pour former une nouvelle image couleur ou en niveaux de gris.

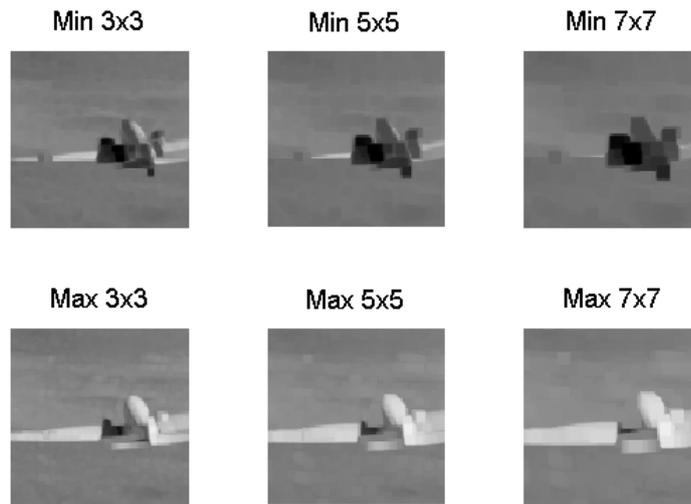


Figure 5.5. – Influence de la taille du voisinage sur le résultat des filtres min et max.

Le traitement des images couleur peut aussi se faire en tenant compte de la nature vectorielle de l'information fournie par chaque pixel. Cette **approche globale** nécessite de développer des algorithmes spécifiques, et est donc moins souvent représentée dans la palette des outils disponibles.

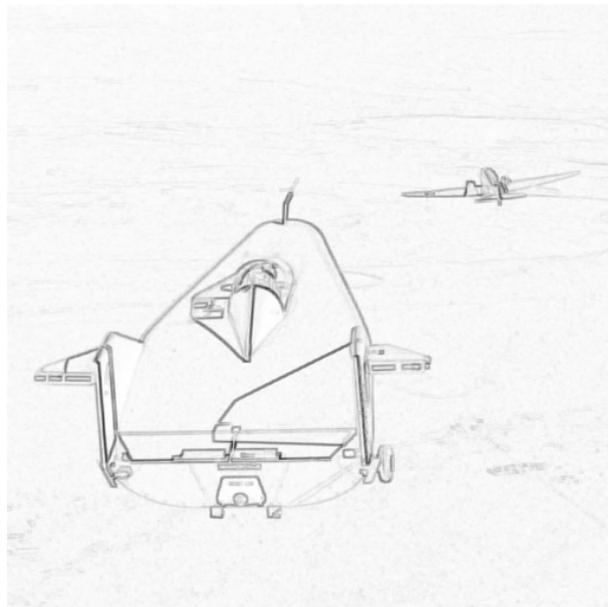


Figure 5.6. – Calcul du filtre de dynamique locale sur une image d'avion.

6. Morphologie Mathématique

La **morphologie mathématique** est une classe d'opérateur de traitements d'images qui a été développée à l'origine pour des images binaires, mais qui a depuis été étendue aux images en niveaux de gris [4, 5]. Les outils seront présentés ici sur des images binaires.

6.1. Filtrage morphologique

Les opérateurs morphologiques travaillent aussi sur le voisinage local de chaque pixel. La forme de ce voisinage est appelé **élément structurant**. Comme pour les filtres linéaires ou médian, on peut utiliser des éléments structurants de taille et de forme variées. Les éléments structurants couramment utilisés sont le disque (isotrope, mais qui nécessite souvent un temps de calcul plus long), et le carré, qui permet d'accélérer les calculs au prix de quelques artefacts.

6.1.1. Érosion et dilatation

Une **dilatation morphologique** consiste à déplacer l'élément structurant sur chaque pixel de l'image, et à regarder si l'élément structurant « touche » la structure d'intérêt. Plus formellement, on teste pour chaque position si l'intersection entre la structure et l'élément structurant après décalage est vide ou non. Le résultat est une structure plus grosse que la structure d'origine (Figure 6.1). En fonction de la taille de l'élément structurant, certaines particules peuvent se trouver connectées, et certains trous disparaître.

L'**érosion** est l'opération inverse, qui est définie comme une dilatation du complémentaire de la structure. Elle consiste à chercher tous les pixels pour lesquels l'élément structurant centré sur ce pixel touche l'extérieur de la structure. Le résultat est une structure rognée (Figure 6.1). On observe la disparition des particules plus petites que l'élément structurant utilisé, et la séparation éventuelle des grosses particules.

On peut remarquer qu'en niveaux de gris, l'érosion est équivalente à l'application d'un filtre minimum, tandis que la dilatation est équivalente à l'application d'un filtre maximum.

6.1.2. Ouverture et fermeture

L'érosion et la dilatation ont l'inconvénient de modifier fortement la taille des structures dans l'image. Pour réduire cet effet, on les utilise souvent en combinaison. On définit ainsi la **fermeture morphologique** comme une dilatation suivie d'une érosion, et l'**ouverture morphologique** comme une érosion suivie d'une dilatation (Figure 6.2).

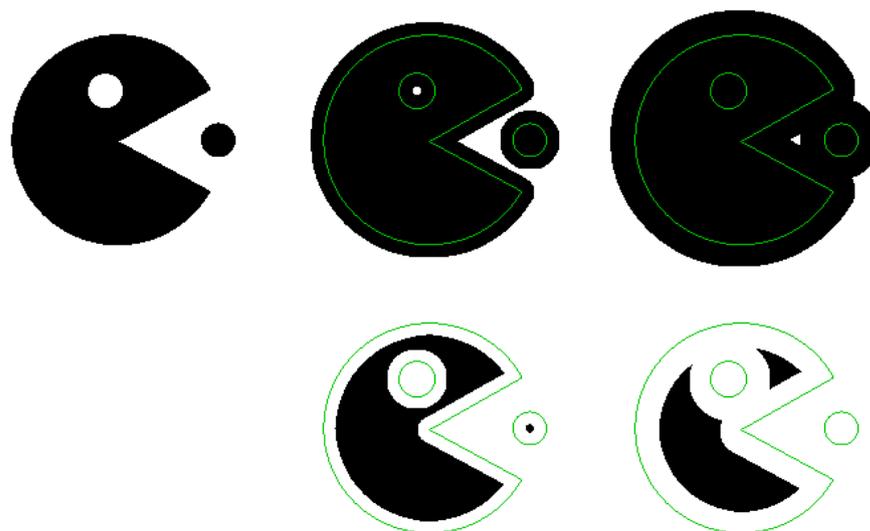


Figure 6.1. – Érosion et dilatation d’une image synthétique. En haut : image d’origine, et résultat d’une dilatation avec deux disques de taille croissante. En bas : résultat d’une érosion avec les mêmes éléments structurants. Le contour de la forme d’origine est affiché en surimpression.

L’ouverture et la fermeture morphologique ont une propriété d’idempotence : le résultat ne change pas si on applique plusieurs fois l’opérateur, il suffit de l’appliquer une seule fois.

L’ouverture et la fermeture morphologique changent relativement peu la forme des grosses structures. Par contre, elle permettent de faire disparaître facilement les petites particules isolées, ou les petits trous à l’intérieur des structures. On les utilise donc souvent pour nettoyer le résultat d’une binarisation. Le résultat d’une ouverture ou d’une fermeture peut aussi être comparé à l’image d’origine : c’est la base des opération de top-hat et de bottom-hat.

6.1.3. Exemple sur une image en niveaux de gris

La figure 6.3 présente l’application d’une fermeture morphologique sur une image de faisceau de maïs observé en microscopie confocale. La fermeture morphologique permet de faire disparaître les petites cellules, ce qui facilite la détection des cellules plus grosses sans changer leur forme de manière trop importante.

6.2. Reconstruction morphologique

La reconstruction morphologique est un des outils essentiels de la morphologie mathématique, car il est utilisé par de nombreux autres algorithmes, tels que le remplissage de

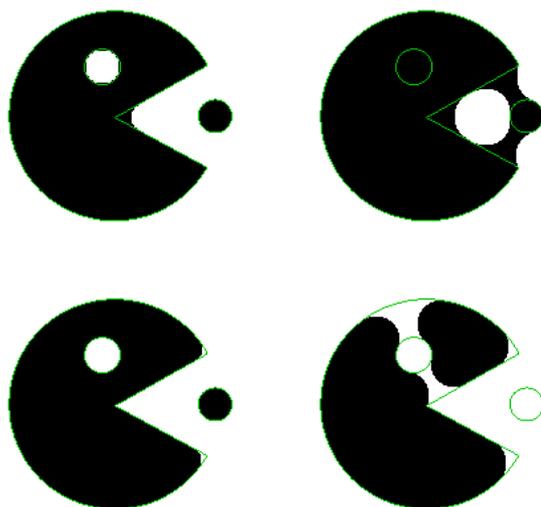


Figure 6.2. – Ouverture et fermeture morphologique appliquées sur une image synthétique. En haut : fermeture, en bas : ouverture. On utilise des éléments structurants en forme de disque, avec un rayon plus grand pour les images de droite.

trous, la suppression des objets touchant les bords de l'image, ou la détection minima et maxima étendus. L'idée est de propager le signal donné par une image de marqueurs initiaux, tout en tenant compte des limites imposées par une image de masque.

6.2.1. Principe de la reconstruction morphologique

La **dilatation conditionnelle** à un masque d'une image binaire consiste à lui appliquer une dilatation et à calculer l'intersection du résultat avec l'image de masque. Pour les images en niveaux de gris, on calcule le minimum des deux images. De manière duale, l'**érosion conditionnelle** à un masque d'une image en niveaux de gris consiste à appliquer une érosion, et à garder le maximum avec l'image de masque.

Une **reconstruction morphologique par dilatation** est obtenue en itérant des dilatations conditionnellement à une image de masque, jusqu'à stabilité. De manière symétrique, une reconstruction morphologique par érosion est obtenue en itérant des érosions morphologiques conditionnellement à une image de masque.

Dans l'exemple ci-dessus, les marqueurs sont sélectionnés manuellement à l'intérieur de grains d'intérêt, et une image binaire des marqueurs est créée. L'image des marqueurs est alors dilatée conditionnellement à l'image initiale, et l'opération est répétée jusqu'à stabilité. Les marqueurs vont progressivement épouser la forme des grains qui les contiennent, mais les grains ne contenant pas de marqueurs ne seront pas présents dans l'image résultat.

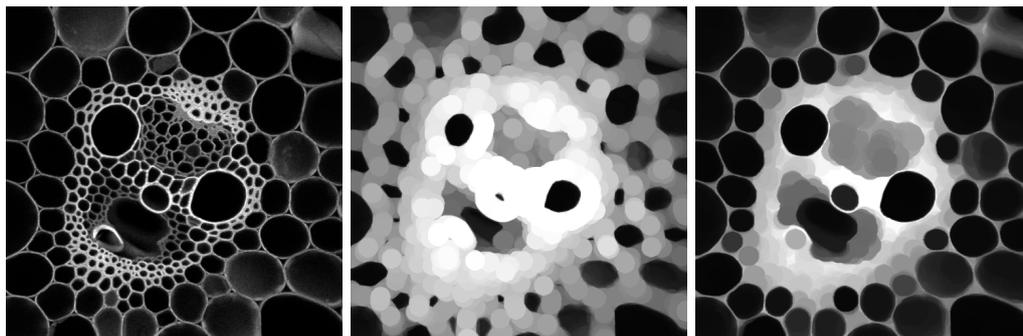


Figure 6.3. – Dilatation et fermeture morphologique sur une image de faisceau de maïs observée en microscopie confocale.

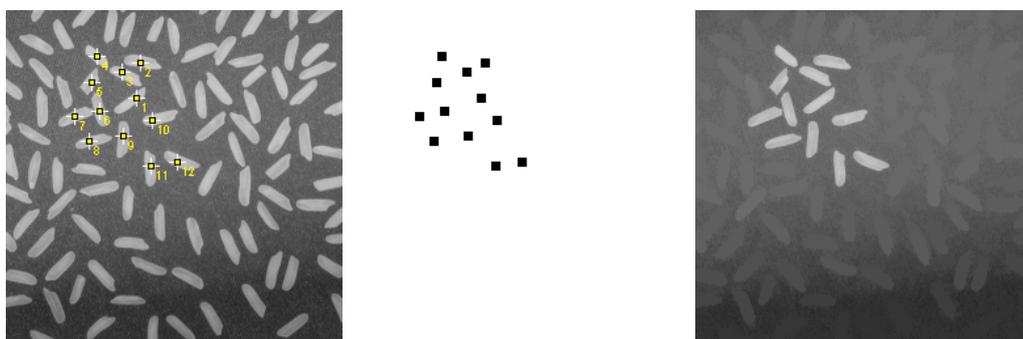


Figure 6.4. – Reconstruction morphologique par dilatation sur une image en niveaux de gris. Des marqueurs sont sélectionnés manuellement, et une image binaire des marqueurs est générée. Cette image de marqueur est utilisée pour reconstruire les grains sélectionnés.

6.3. Minima et maxima

6.3.1. Minima et maxima régionaux

On définit les **maxima régionaux** comme des régions composées d'un ensemble connecté de pixels ayant tous la même valeur, et dont les voisins ont des valeurs strictement inférieure. Les minima régionaux sont définis de manière symétrique comme des régions dont les voisins ont tous des valeurs strictement supérieure à celle de la région.

On peut calculer l'image des maxima régionaux à partir de la différence entre l'image et le résultat de la reconstruction morphologique par dilatation de $I - 1$ sous I :

$$\text{RMAX}(I) = I - R^\delta(I - 1, I) \quad (6.1)$$

ou, dans le cas où le type de données utilisé ne permet pas de représenter les valeurs négatives, par l'équation suivante :

$$\text{RMAX}(I) = (I + 1) - R^\delta(I, I + 1) \quad (6.2)$$

Un exemple de détection des maxima régionaux sur une image en niveaux de gris est présenté sur la figure 6.5.

De manière symétrique, les **minima régionaux** sont définis à partir d'une reconstruction géodésique par érosion de $I + 1$ au-dessus de I :

$$\text{RMIN}(I) = R^\varepsilon(I + 1, I) - I \quad (6.3)$$

6.3.2. Minima et maxima étendus

Les maxima étendus sont définis de manière assez similaire, mais de manière plus permissive. On se fixe une valeur de "dynamique", et on recherche les ensembles de pixels tels que la différence des intensités des pixels soit inférieure à la dynamique, et que tous les voisins aient une valeur inférieure au maximum de la région moins la dynamique. Cette nouvelle définition permet d'identifier des maxima plus étendus, et de limiter des effets du bruit dans l'image.

Les **minima et maxima étendus** sont définis à partir du résultat d'une reconstruction morphologique par érosion de $I + 1$ au-dessus de I :

$$\text{HMAX}(I, h) = R^\delta(I, I + h) \quad (6.4)$$

$$\text{EMAX}(I, h) = \text{RMAX}[\text{HMAX}(I, h)] \quad (6.5)$$

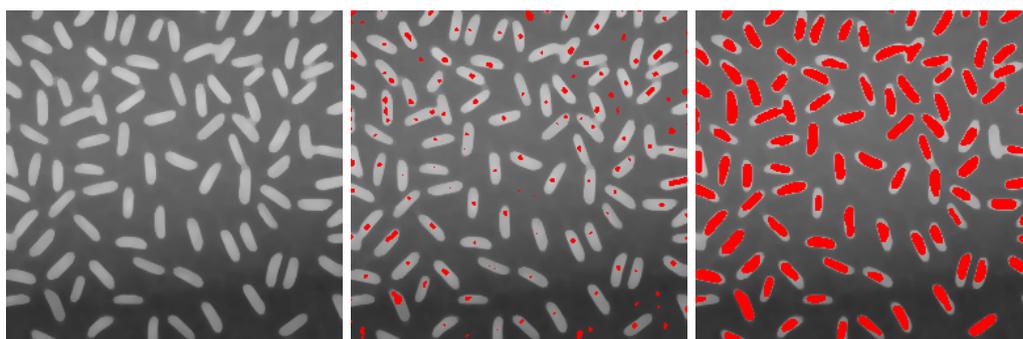


Figure 6.5. – Détection des maxima régionaux et étendus sur une image en niveaux de gris.

Dans l'exemple de la figure 6.5, on s'aperçoit que la détection des maxima régionaux (image du centre) donne beaucoup de faux positifs (des régions détectées comme des maxima, mais qui n'en sont pas). De plus, la taille des maxima est relativement petite, et certains grains sont associés à plusieurs maxima. Au contraire, la détection des maxima étendus (image de droite), ici avec un seuil de 10, montre une meilleure correspondance entre les maxima détectés et les grains recherchés.

De manière symétrique, les minima régionaux sont définis à partir d'une reconstruction morphologique par érosion de $I + 1$ au-dessus de I :

$$\text{HMIN}(I, h) = R^\varepsilon(I + h, I) \quad (6.6)$$

$$\text{EMIN}(I, h) = \text{RMIN}[\text{HMIN}(I, h)] \quad (6.7)$$

6 Morphologie Mathématique

Les minima étendus sont particulièrement efficace pour initialiser de manière automatique les marqueurs d'une ligne de partage des eaux.

Troisième partie

Le traitement des images

7. Filtrage et restauration d'images

7.1. Modification d'histogramme

Lorsque les conditions d'acquisition d'une image en niveaux de gris sont mal maîtrisées, il peut arriver que l'image apparaisse trop sombre ou trop foncée. On peut améliorer l'affichage final, en modifiant l'histogramme de l'image.

7.1.1. Extension de dynamique

Cette méthode consiste à utiliser au mieux les niveaux de gris situés dans une plage de valeur donnée, et à les étaler de manière régulière pour couvrir toute la dynamique possible des niveaux de gris.

En pratique, la relation entre les niveaux de gris d'entrée et les niveaux de gris de sortie est une fonction affine, arrondie à des valeurs entières entre 0 et 255, que l'on peut calculer à partir des valeurs minimum et maximum des plages de niveaux de gris en entrée :

$$f(g) = 255 \times \frac{g - g_{\min}}{g_{\max} - g_{\min}} \quad (7.1)$$

La figure 7.1 montre l'image de départ après avoir transformé les niveaux de gris situés entre les valeurs 95 et 170 en des niveaux de gris situés entre 0 et 255.

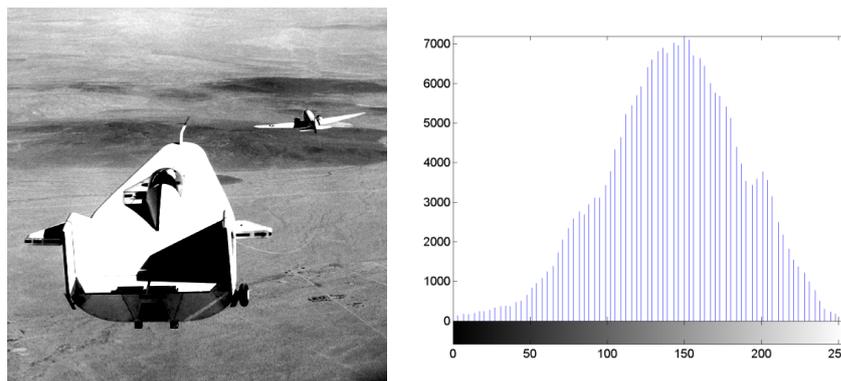


Figure 7.1. – Une image en niveau de gris après extension de sa dynamique, et son histogramme.

7.1.2. Égalisation d'histogramme

Cette opération a pour but de rendre un histogramme le plus plat possible. Elle calcule de manière automatique la fonction de transformation $f(g)$, en associant l'image à une modélisation probabiliste dont l'histogramme représente la densité de probabilité des niveaux de gris.

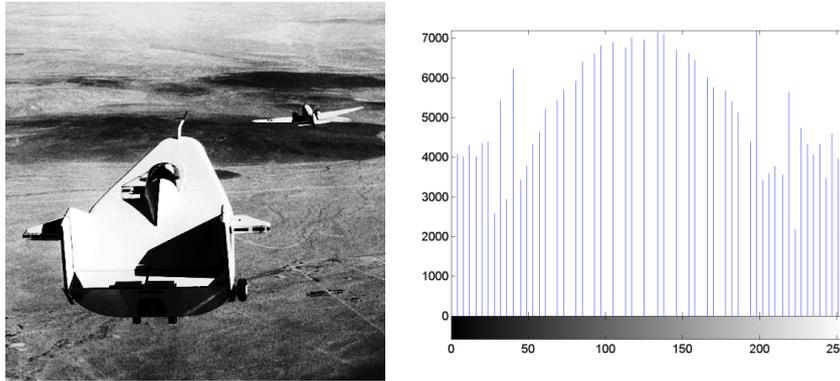


Figure 7.2. – Une image en niveau de gris après égalisation de son histogramme, et son histogramme.

7.2. Corrections géométriques

Les corrections géométriques ont pour but de compenser certaines déformations dues à l'optique des appareils (voir la figure 7.3), à des problèmes de perspective ou de rotation (pour des images obtenues avec un appareil photo classique ou une caméra), ou tout simplement pour créer une image mosaïque à partir de plusieurs images parcellaires.

La correction se déroule en deux étapes :

- choix du modèle de déformation et calcul de ses paramètres, par rapport à l'aide de points de repères
- création de l'image résultat à partir de l'image de départ et des paramètres de la déformation

Le calcul de l'image finale nécessite de connaître la valeur d'intensité pour des positions qui ne sont pas situées sur la grille image. Cette intensité est calculée à partir des pixels voisins, en utilisant une fonction d'interpolation (linéaire, cubique...). Le résultat est ensuite converti en niveaux de gris.

7.3. Soustraction du fond

Un artefact courant en microscopie optique est d'avoir le centre de l'image beaucoup plus lumineux que les coins. Les mesures d'intensité dépendent donc fortement de la position dans l'image, et il peut être nécessaire de corriger ce biais.

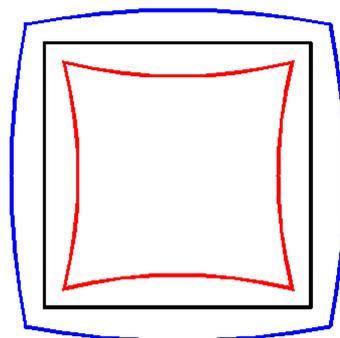


Figure 7.3. – Exemples de distorsions géométriques pouvant survenir dans le cas d'un défaut d'optique. Le carré noir au centre représente l'image sans distorsion, le contour bleu extérieur représente une déformation en coussinet, et le contour rouge au centre représente une distorsion de type barillet.

Les méthodes de correction classiques consistent à combiner une image de fond avec l'image observée. L'image de fond peut être une image de calibration acquise au préalable, avant d'observer l'échantillon. Si l'image de calibration est obtenue en observant une lame vide, les différences d'intensité observées sont alors dues uniquement aux différences d'illumination.

Si une image de calibration n'est pas disponible, il existe plusieurs algorithmes permettant d'estimer une image de fond à partir d'une image expérimentale. On fait en général l'hypothèse que la taille des structures observées est petite devant l'étendue des variations d'intensité du fond. Les algorithmes populaires de cette famille sont le « rolling ball », et le chapeau haut-de-forme (ou tophat).

Le **rolling ball** considère une boule qui se promène sous la surface formée par les niveaux de gris. On suppose que le fond est sombre et les structures blanches. Dans les zones homogènes, le haut de la boule correspond au niveau de gris local, et donc au fond. Dans les zones avec des pics, la boule « bute » sur le niveau le plus sombre trouvé dans le voisinage, qui est assimilé au fond. En calculant l'altitude du haut de la boule pour chaque pixel de l'image, on en déduit une image approximative du fond, que l'on peut ensuite soustraire de l'image observée.

Sous ImageJ, cette opération est disponible via le menu `Process > Subtract Background`. On peut régler le rayon de la boule, et spécifier la couleur du fond.

Si la taille des structures observées est grande par rapport à la portée des variations, les approches de type rolling-ball ou top-hat vont introduire plus d'artefacts qu'elles n'en corrigent. Si les variations sont assez régulières, comme c'est souvent le cas pour les halos d'illumination, on peut modéliser ces variations sur l'ensemble de l'image, de manière à inférer la correction à apporter à l'intérieur des structures (Figure 7.5). La figure 7.5 montre un exemple de calcul d'une image de fond en utilisant l'information de

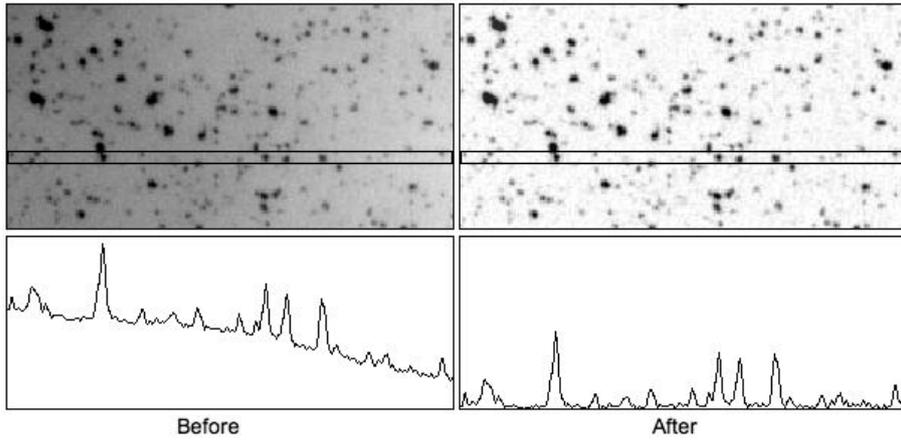


Figure 7.4. – Exemple de correction d'intensité du fond en utilisant l'outil « Rolling Ball » sous ImageJ

la périphérie de l'image, et en ajustant une surface polynomiale.

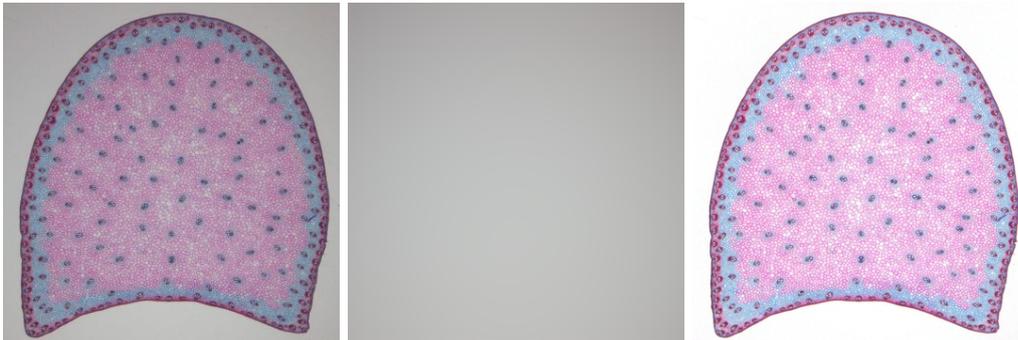


Figure 7.5. – Correction de fond par modélisation polynomiale des variations d'intensité. (a) Image originale. (b) Modélisation des variations d'intensité dans l'image. (c) Image corrigée.

8. Segmentation

On appelle **segmentation d'une image** l'opération consistant à identifier les structures d'intérêt dans une image. On distingue deux types d'approches pour la segmentation : contour ou région.

Dans l'**approche contour**, on cherche à isoler le ou les contours des objets d'intérêt. Le résultat se présente en général sous la forme d'un ensemble de chaînes de pixels, et des traitements additionnels sont souvent nécessaires pour associer les contours aux objets d'intérêt.

L'autre approche cherche à **identifier des régions** de pixels homogènes au sein de l'image. Le critère d'homogénéité peut être l'intensité, la couleur, ou même la texture locale. Le résultat se présente soit sous la forme d'une image binaire, soit d'une image étiquetée, chaque étiquette ou label correspondant à une région. Les méthodes de seuillage et la ligne de partage des eaux sont les représentantes emblématiques de cette approche.

8.1. Seuillage

Le **seuillage** (en anglais *threshold*) est sûrement la méthode de segmentation la plus simple. Elle consiste à tester pour chaque pixel de l'image si sa valeur est supérieure ou inférieure à un certain seuil, et produit une image binaire regroupant les résultats. Il est aussi possible d'utiliser plusieurs seuils, voire de sélectionner les pixels appartenant à une plage spécifique de niveaux de gris.

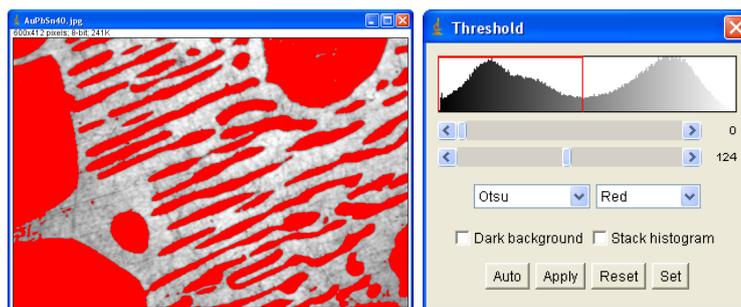


Figure 8.1. – Segmentation d'une image en niveaux de gris par la méthode d'Otsu sous ImageJ. A gauche, le résultat de la segmentation est affiché en surimpression. A droite, l'histogramme de l'image est affiché, avec la plage de niveaux de gris sélectionnée.

8.1.1. Détection automatique du seuil

Il existe plusieurs méthodes pour **détecter de manière automatique la valeur du seuil** à appliquer. L'une des plus répandue est la méthode d'Otsu. Elle consiste à faire l'hypothèse que l'image contient deux classes, décrites chacune par une partie de l'histogramme de l'image. La qualité du seuillage est quantifiée en mesurant la variance des niveaux de gris de chaque classe. On cherche ensuite la valeur de seuil qui minimise les variances des deux classes, et on utilise cette valeur pour binariser l'image (Figure 8.1).

8.1.2. Seuillage à hystéresis

Le **seuillage à hystéresis** est une méthode un peu plus perfectionnée, qui permet d'améliorer la détection des contours des zones contrastées, tout en évitant de détecter des zones du fond.

Le principe est d'utiliser deux seuils : un seuil « haut » et un seuil « bas ». On sélectionne d'abord l'ensemble des pixels au dessus du seuil haut, puis l'ensemble des pixels au-dessus du seuil bas. On ne garde ensuite que les composantes connexes du seuil bas qui contiennent au moins un pixel au dessus du seuil haut.

En pratique, cette détection peut se faire soit par reconstruction géodésique, soit par étiquetage des composantes connexes de l'image des seuils bas et détection des étiquettes présentes dans l'image des seuils hauts.

8.1.3. Seuillage d'image couleurs

Dans le cas d'images couleurs, chaque pixel est défini par un triplet (R, G, B) de valeurs, et on ne peut plus se baser sur un seuil simple. Une première possibilité pour segmenter une image couleur est de la **convertir en niveaux de gris**, et de travailler sur l'image d'intensité. L'inconvénient est que l'on perd de l'information, et que certaines couleurs différentes présentent un niveau de gris après conversion très proche.

Pour tirer parti de l'ensemble des canaux, une solution est de **segmenter chaque canal** séparément, et de combiner les résultats des différents seuillages. On remarque cependant que les informations portées par les différents canaux sont souvent fortement corrélées, et qu'il est plus judicieux de travailler sur des images dans un **espace de couleur transformé**. L'espace HSV, par exemple, permet de séparer les notions de teinte, de saturation, et d'intensité, ce qui simplifie la recherche des seuils. Sous ImageJ, le plugin « Color Threshold » se révèle très pratique pour la segmentation d'images couleur.

Chaque pixel couleur pouvant être vu comme un point dans un espace tridimensionnel, il est aussi envisageable d'utiliser des **méthodes de classification**, supervisées ou non supervisées. Pour une classification non supervisée (*clustering*), le principe est d'identifier les pixels formant des groupes homogènes. La figure 8.3 présente le cas d'une segmentation par les k-means appliquée à une image de coupe de tige maïs colorée au Fasga, et préalablement filtrée. Dans le cas d'une classification supervisée, les groupes seront définis à partir d'un ensemble de pixels d'apprentissage, utilisés pour déterminer les caractéristiques colorimétriques des différentes classes. L'avantage de ces méthodes est que

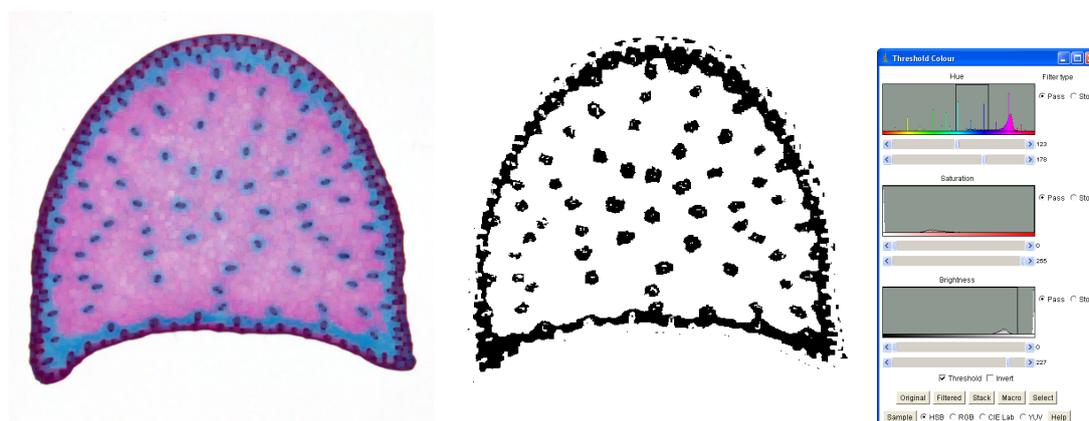


Figure 8.2. – Segmentation d’une image couleur par seuillage des composantes de l’image après conversion dans l’espace HSV. Le canal de teinte permet de sélectionner les pixels de couleur bleue, et le canal d’intensité permet d’éviter la sursegmentation.

les frontières entre classes peuvent être beaucoup plus libres que dans le cas d’un simple seuillage.

8.2. Classification de pixels

Les méthodes de classification, utiles pour la segmentation d’images couleur, peuvent aussi s’employer pour segmenter des images complexes en niveaux de gris. Pour discriminer efficacement les différentes classes de pixel, on cherche à décrire chaque pixel par un ensemble de caractéristiques. En appliquant une large variété de filtres de différente nature à l’image d’origine, on obtient pour chaque pixel autant de valeurs que le nombre de filtres appliqués. En choisissant des régions représentatives de chaque classe ou catégorie dans l’image, on peut initialiser un classifieur, qui va calculer les caractéristiques de chaque classe. En appliquant le résultat sur l’image d’origine, et si la famille de filtres a été correctement choisie, il est possible d’obtenir une segmentation d’image qu’il aurait été difficile d’obtenir autrement.

8.3. Détection de contours (*edge detection*)

8.4. Segmentation de régions par ligne de partage des eaux

La **ligne de partage des eaux** est un outil de segmentation issu de la morphologie mathématique, qui trouve de nombreuses applications aussi bien en segmentation que comme outil de post-traitement. Une application est la segmentation de cellules séparées par leurs parois. En appliquant la ligne de partage des eaux sur une image gradient, on peut détecter les contours des objets. On l’utilise ainsi couramment pour séparer des particules binarisées qui se touchent (section 9.4).

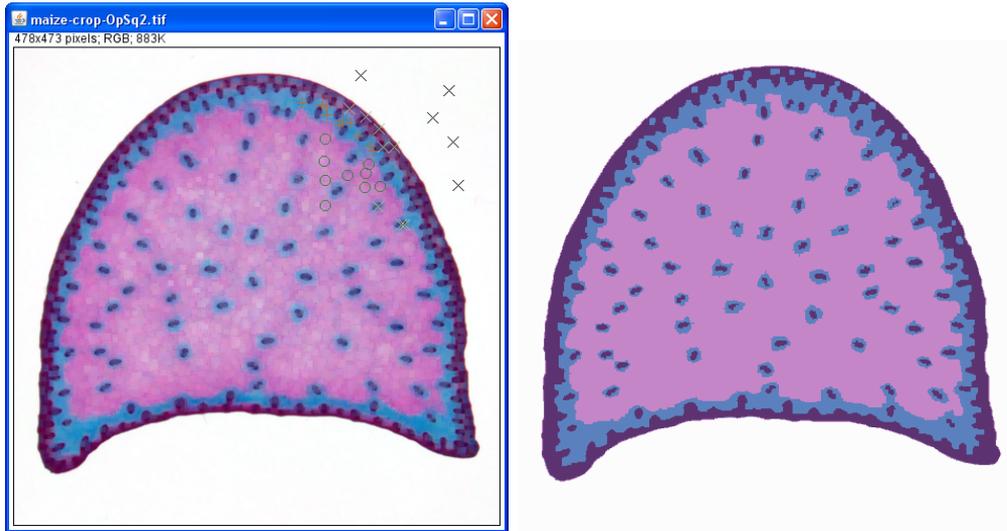


Figure 8.3. – Segmentation d’une image couleur par la méthode des k-means : sélection des marqueurs pour les différentes classes, et résultat de la classification appliquée à l’ensemble de l’image.

8.4.1. Principe

L’algorithme s’applique à des images en niveaux de gris. L’image est assimilée à un paysage dont l’altitude de chaque point est donnée par l’intensité du pixel correspondant. Les pixels de forte intensité (blancs) correspondent à des sommets, tandis que les pixels de faible intensité (noir) correspondent à des fonds de vallées.

Le principe de l’algorithme est de partitionner l’image selon une approche topographique : on recherche les bassins versants, séparés par des lignes de crêtes.

8.4.2. Application sur une image de tissu végétal

La ligne de partage des eaux est particulièrement bien adaptée pour la segmentation d’images de tissus végétaux, dans lesquelles les parois des cellules sont bien contrastées par rapport au fond. L’utilisation d’un seuillage n’est pas toujours le plus efficace : les variations d’intensité du fond rendent difficile l’identification du seuil, et le résultat de la segmentation laisse beaucoup de bruit à l’intérieur des cellules sous la forme de pixels isolés (8.4).

L’application de la ligne de partage des eaux permet de segmenter efficacement les cellules dans l’image. Les fonds des cellules représentent les bassins versants, tandis que les parois sont les lignes de crête de l’image. Le résultat est indépendant des variations d’intensité globales dans l’image, et l’image binaire résultante produit des structures dont l’épaisseur est régulière (1 ou 2 pixels). Le résultat est soit l’image binaire de la ligne de partage des eaux (8.4-b), soit l’image étiquetée correspondant aux différentes régions, ici les cellules.

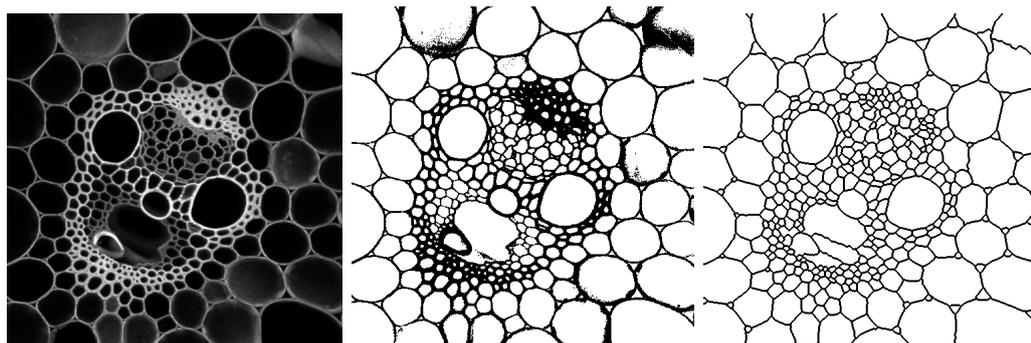


Figure 8.4. – Une image de cellules végétales (à gauche), et le résultat de la segmentation par deux méthodes : seuillage (au centre), et ligne de partage des eaux (à droite).

Il y a cependant quelques précautions à prendre avant d'appliquer la ligne de partage des eaux :

- Les images gagnent à être **filtrées** au préalable, sous peine de voir apparaître de trop nombreuses régions (phénomène de sur-segmentation)
- Il est possible d'**imposer les minima** à appliquer à l'image, soit manuellement, soit sur la base de la dynamique des niveaux de gris des régions
- les régions obtenues peuvent éventuellement faire l'objet de **post-traitements** : division de particules non convexes, fusion de régions correspondant à une même cellule...

8.4.3. Application sur une image gradient

La ligne de partage des eaux est souvent utilisée sur des images de gradient pour identifier le contour d'objets qui apparaissent contrastés par rapport au fond. On commence par calculer le gradient de l'image, par exemple à l'aide de filtres de Prewitt ou de Sobel. On calcule ensuite la norme du gradient (8.5-b).

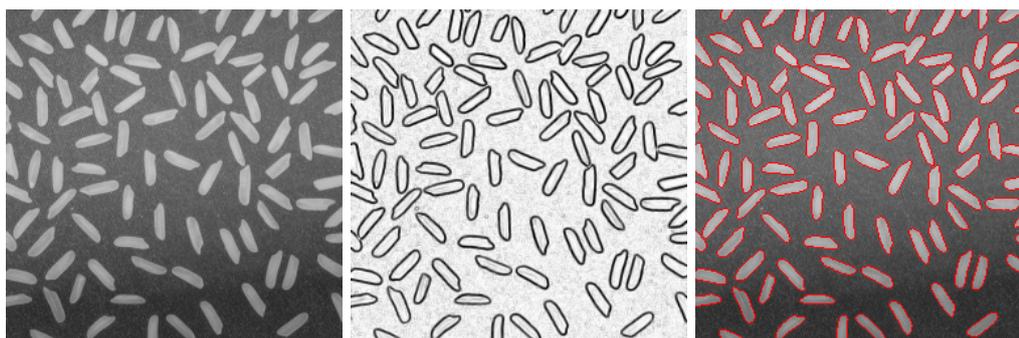


Figure 8.5. – Application de la ligne de partage des eaux sur le gradient d'une image pour détecter le contour des grains. (a) image d'origine (b) norme du gradient (représenté en inversé) (c) résultat de la ligne de partage des eaux superposée sur l'image d'origine.

En filtrant un peu (ici : détection et imposition des minima étendus), on obtient une ligne de partage des eaux sur l'image gradient qui correspond aux contours des particules (ici des grains) dans l'image (8.5-c).

On note sur la figure exemple que la segmentation s'affranchit très bien de la variation d'intensité du fond.

8.5. Détection de structures (feature detection)

8.5.1. Détection des coins

Plutôt que des bords, on cherche parfois à extraire les extrémités, ou les coins des structures observées. Pour cela, le **détecteur de coins de Harris** est une des méthodes les plus populaires.

8.5.2. Détection de lignes ou de cercles

Il est possible d'identifier des structures linéaires ou circulaires dans les images, en utilisant la **transformée de Hough**.

9. Traitement des images binaires

Ce chapitre regroupe plusieurs techniques spécifiques aux images binaires, couramment employées suite à des outils de segmentation, soit pour améliorer le résultat de la segmentation, ou pour simplifier la structures des objets visibles dans l'image.

9.1. Carte des distances

La carte des distances (*distance map*), ou fonction distance, est un outil très employé en traitement d'images. Elle consiste à calculer, pour chaque pixel, quelle est la distance la plus proche à une structure donnée. Par exemple, on peut calculer la distance au bord le plus proche pour les pixels à l'intérieur d'une particule (Figure 9.1).

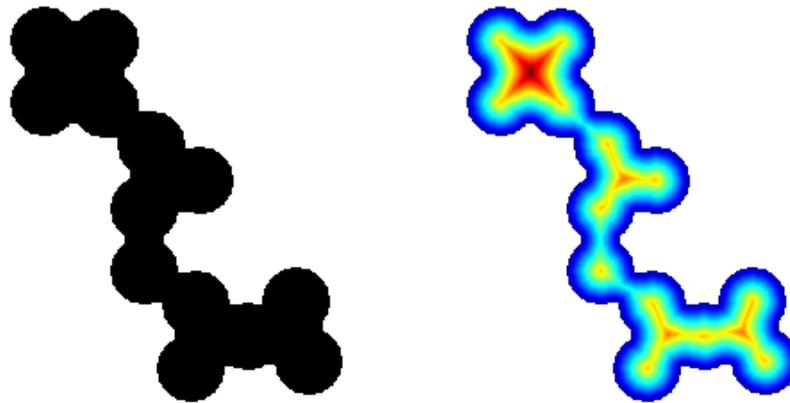


Figure 9.1. – Carte des distances d'une image binaire

La carte des distances a de nombreuses applications :

- calcul du squelette d'une particule
- calcul de l'épaisseur moyenne le long du squelette
- calcul des zones d'influences de petits objets
- séparation de particules coalescentes
- ...

9.2. Nettoyage des résultats de segmentation

De nombreux outils sont disponibles pour nettoyer les résultats obtenus après segmentation.

9.2.1. Suppression des petits objets

L'application d'un seuillage sur une image binaire produit souvent de nombreux faux positifs, correspondant à des variations d'intensité dans le fond de l'image qui sont détectés comme structure d'intérêt

Filtrage morphologique

Une manière simple de supprimer ces artefacts est d'appliquer une ouverture morphologique binaire à l'image seuillée. Les particules dont la taille est inférieure à celle de l'élément structurant seront supprimées, tandis que les particules correspondant aux structures d'intérêt seront globalement préservées.

Si la forme des particules est très altérée par l'ouverture (ce qui peut arriver avec un élément structurant carré de grande taille), une possibilité est de rajouter une reconstruction géodésique, en utilisant comme marqueur l'image après ouverture, et comme masque l'image seuillée avant ouverture.

Ouverture par attribut

Une alternative aux filtres morphologiques est d'utiliser une « ouverture par attribut ». Le principe est de choisir un paramètre descriptif, et une plage de valeurs correspondant aux particules que l'on veut garder. Un choix typique est l'aire, et une valeur minimum correspondant aux particules « suffisamment grandes » (Figure 9.2). On calcule le ou les paramètres choisis pour chacune des particules identifiées, et on ne garde que les particules pour lesquelles les paramètres sont dans la plage de validité.

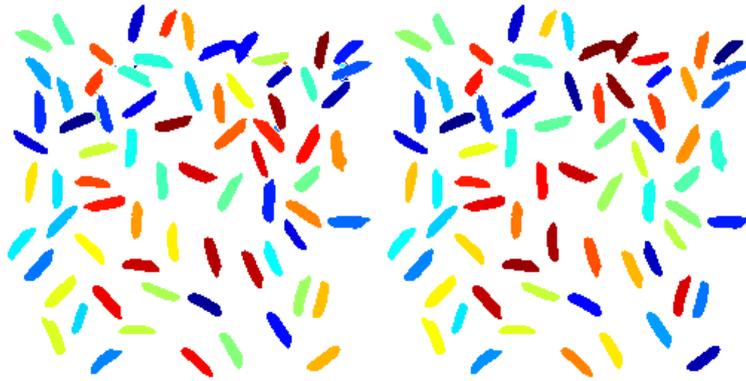


Figure 9.2. – Ouverture par attribut sur une image étiquetée. Seules les particules d'une taille supérieure à 15 pixels ont été gardées.

Cette méthode est implémentée dans ImageJ, dans les options de la commande « Analyze Particles... », mais uniquement pour les paramètres de taille et de circularité. Sous Matlab, on peut utiliser la fonction « `bwareaopen` », ou bien passer par la fonction « `regionprops` » pour un traitement plus précis.

9.2.2. Suppression des particules touchant les bords

En imagerie microscopique, il arrive couramment que les objets d'intérêt ne soient pas entièrement visibles dans l'image, et qu'une partie soit tronquée. Une pratique courante consiste donc à supprimer les objets qui touchent le bord de l'image (Figure 9.3).

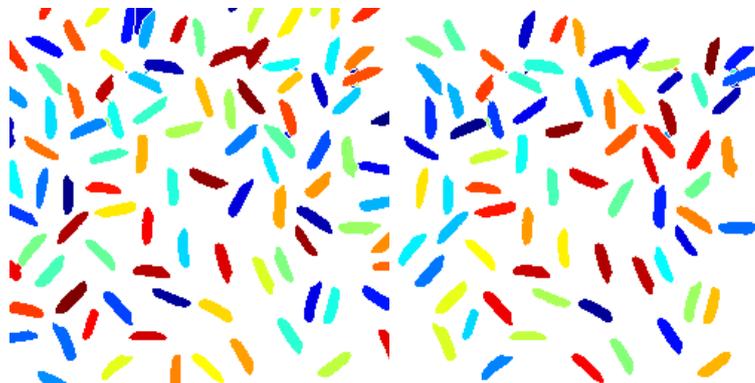


Figure 9.3. – Suppression des particules touchant les bords.

En pratique, la suppression s'opère à l'aide d'une reconstruction géodésique, avec comme marqueur l'intersection des objets avec le bord de l'image, et comme masque l'image binaire elle-même. On obtient comme résultat l'image des objets touchant le bord, qu'il suffit ensuite de soustraire de l'image d'origine. Cet outil est souvent implémenté comme une fonction à part entière dans les logiciels (fonction `imclearborder` sous Matlab, option de la fonction `Analyze Particles` sous ImageJ, outil « `borderKill` »...).

Il faut toutefois prendre garde qu'en opérant ainsi, on introduit un biais dans les mesures qui seront faites par la suite, car les paramètres seront mesurés uniquement sur les particules entièrement visibles. Ainsi, plus une particule est petite, plus sa probabilité d'être observée sera importante, et les mesures globales seront biaisées en faveur des petites particules. Une possibilité d'obtenir des mesures plus proches de la réalité est d'appliquer la correction de Miles-Lantuéjoul, qui pondère les mesures sur les particules en fonction de leur probabilité d'échantillonnage.

9.3. Codage des contours

On considère, pour chaque région ou particule de l'image binaire, la succession des pixels qui forment son contour. Au lieu de stocker l'ensemble des coordonnées de chaque pixel, on code la direction du pixel suivant, en utilisant une valeur entre 0 et 7. On obtient ainsi le **code de Freeman**.

9.4. Séparation de particules se recouvrant

La ligne de partage des eaux est souvent utilisée pour séparer des particules qui se touchent après un seuillage (9.4-a). On commence par calculer la fonction distance par

rapport au complémentaire des particules. Les maxima de l'image résultat sont situés vers le centre des particules. On inverse cette fonction distance, et on filtre un peu : les minimas sont alors les centres des particules (9.4-b).



Figure 9.4. – Application de la ligne de partage des eaux pour séparer des particules connectées. (a) Image d'origine (b) fonction distance inversée et ligne de partage des eaux sur la fonction distance (c) séparation des particules.

On applique alors l'algorithme de ligne de partage des eaux. Les lignes de crêtes détectées sont les séparations entre les particules (9.4-c). En combinant avec l'image binaire d'origine, on arrive à séparer la plupart des particules (9.4-d).

9.5. Squeletisation

La squeletisation consiste à supprimer certains pixels pour ne garder que ceux situés le long des axes principaux. L'objectif est de garder la forme générale de la particule, afin de pouvoir en extraire des informations de nature topologique : points terminaux (extrémités du squelette), les points multiples (croisements), points formant les branches... Quelques exemples de squelettes sont montrés sur la figure 9.5.

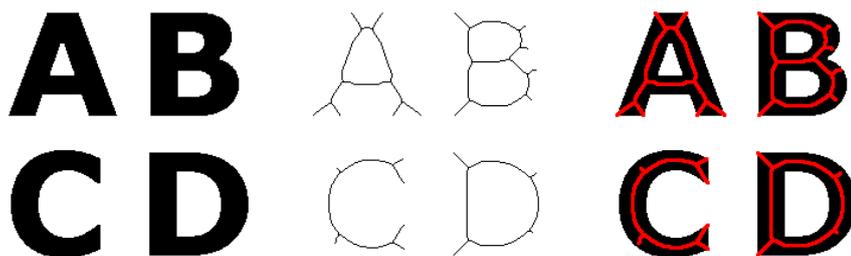


Figure 9.5. – Squeletisation d'une image binaire. Image d'origine, résultat de la squeletisation, et superposition du squelette sur l'image d'origine.

On définit le squelette dans un espace continu comme l'ensemble des points de la particule qui sont situés à égale distance d'au moins deux points de la frontière de l'ensemble

d'intérêt [2, p. 162]. Une définition plus formelle fait appel à l'ensemble des centres des boules de rayon maximal.

Les algorithmes de squeletisation utilisent la définition du squelette, adaptée à la nature discrète des images, pour calculer une approximation du squelette euclidien. On obtient une image binaire dont les structures ont une épaisseur maximale d'un pixel, et qui reprennent la forme générale des particules.

Un des problèmes souvent rencontrés est que la squeletisation est une opération très sensible au bruit : une petite variation de forme peut engendrer des variations importantes du squelette. Cela se traduit notamment par la présence de nombreuses « barbules », qui relient l'axe du squelette aux petites aspérités du contour (sur les parties courbes du B, par exemple). Des outils dits « d'ébarbulage » (en anglais : *pruning*) sont parfois disponibles dans les logiciels.

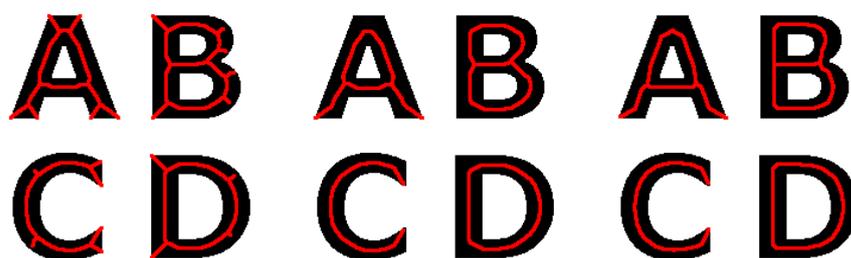


Figure 9.6. – Différents algorithmes de squeletisation : résultats obtenus sous Matlab (par squeletisation et amincissement homotopiques), et sous ImageJ.

Un autre problème est qu'il existe de nombreux algorithmes de squeletisation chacun donnant des résultats différents (voir la figure 9.6). Les algorithmes les plus courants se basent soit sur des cartes de distance, soit sur des amincissements homotopiques (amincissements successifs qui préservent la topologie des particules).

Quatrième partie

L'analyse et l'exploitation des
images

10. Analyse

L'analyse d'image a pour but d'extraire des paramètres quantitatifs des images. Ces paramètres peuvent ensuite être soumis à une analyse statistique plus poussée, et/ou être confrontés à d'autres paramètres issus de méthodes d'acquisition complémentaires : tests mécaniques, analyses sensorielles, analyses physico-chimiques...

On distingue différentes familles de méthodes en fonction du type d'image à analyser.

- L'analyse de la **morphologie de particules** est la démarche qui vient souvent en premier lieu à l'esprit. Elle consiste à déterminer, pour chaque particule identifiée dans l'image, des paramètres décrivant le plus souvent la taille, la forme, ou les statistiques des niveaux de gris de la particule
- Pour des **images 3D**, l'analyse des particules est un peu plus difficile, notamment du fait qu'il est plus compliqué d'observer entièrement un nombre significatif de particules sur une même image.
- Dans certains cas, la structure d'intérêt se prête mal à une description par un ensemble de particules : les solides alvéolaires comme la mie de pain, les structures ramifiées... Des **paramètres géométriques globaux** permettent néanmoins une description qui peut s'avérer pertinente.
- Un problème particulièrement ardu est la détermination de paramètres 3D quand on ne dispose que d'images de coupes planes 2D. On entre ici dans le champ de la **stéréologie**, une discipline qui mêle géométrie stochastique et statistiques.
- Enfin, de nombreuses applications fournissent des images dans lesquelles les structures d'intérêt ne peuvent être clairement identifiées. Dans ce cas, les méthodes d'**analyse de texture d'image** peuvent se révéler fructueuses.

Il peut être utile de rappeler que dans tous les cas, il est nécessaire de disposer de la résolution de l'image. Cette résolution peut être fournie par le dispositif d'acquisition (microscope...). Dans le cas inverse, l'utilisation de mires de calibration permet de connaître a posteriori la résolution des images.

10.1. Morphologie de particules

De nombreuses images présentent un nombre parfois important de particules facilement identifiables : cellules dans un tissu, différents échantillons pris en photographie, scans de feuilles de plantes... Les questions qui se posent sont de déterminer des paramètres morphologiques pertinents permettant par exemple de discriminer des populations d'individus [2]. Ces paramètres peuvent ensuite être utilisés pour des tests statistiques, ou pour classer les particules en différentes classes.

10.1.1. Aire et périmètre

Aire

L'**aire** permet de caractériser la taille d'une particule. C'est un des paramètres les plus simples à mesurer : pour une particule entièrement visible, il suffit de compter le nombre de pixels qui la composent, et de multiplier par l'aire d'un pixel.

On préfère parfois d'autres paramètres, qui en sont souvent dérivés :

le **diamètre équivalent**, qui correspond au diamètre qu'aurait une particule sphérique ayant la même aire

le **logarithme de l'aire**, qui a tendance à avoir une distribution plus symétrique que pour l'aire (on a souvent plus de petites particules que de grosses)

Périmètre

Après l'aire, le **périmètre** est souvent le deuxième paramètre que l'on cherche à mesurer sur une particule. Malheureusement, il existe plusieurs méthodes pour mesurer le périmètre, et elles ne sont pas toujours implémentées de manière identique selon les logiciels.

Une méthode naïve pour mesurer le périmètre consiste à **compter le nombre de pixels** composant la frontière de la particule (voir la figure 10.1).

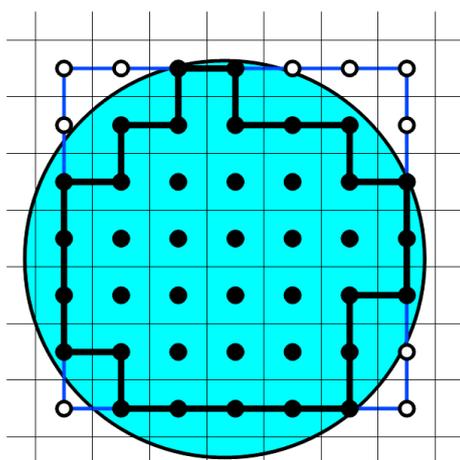


Figure 10.1. – Méthode naïve pour déterminer le périmètre. On obtient le même périmètre pour le cercle et pour le carré contenant le cercle.

Cette méthode n'est cependant pas très précise. Si on considère par exemple l'image d'un disque, on obtiendra la même mesure de périmètre pour le disque que pour le carré contenant le disque, et ceci quelle que soit la résolution de l'image. On peut imaginer des pondérations pour les pixels diagonaux, mais cela ne résout pas le problème.

La **méthode de Crofton** consiste à compter le nombre d'intersections entre la structure et un ensemble de droites d'orientations variées. On peut en effet démontrer que

le nombre d'intersections est proportionnel au nombre de droites et au périmètre de la structure d'intérêt. Pour des images numériques, on utilise des droites horizontales, verticales, et éventuellement diagonales.

Circularité

Le périmètre et l'aire sont deux paramètres mesurant la taille des particules. Afin de s'affranchir des l'effet de la taille, on utilise parfois le rapport de ces deux paramètres, afin de définir un **facteur de circularité** (appelé aussi plus simplement mais de manière plus ambiguë « facteur de forme », ou encore « déficit isopérimétrique ») :

$$f = 4\pi \frac{A}{p^2}$$

La constante 4π permet de normaliser le paramètre entre 0 (particule très allongée ou très rugueuse) et 1 (particule circulaire et donc très compacte). Du fait des erreurs qui peuvent être faites sur les mesures (notamment de périmètre), il peut arriver que la valeur calculée pour la circularité dépasse 1, il faut donc dans ce cas tronquer le résultat. On considère parfois l'inverse de la circularité, ce qui permet d'avoir un paramètre plus facilement interprétable. Un autre inconvénient de ce paramètre est qu'il combine deux informations : la rugosité de la frontière, et l'allongement de la particule. L'interprétation des résultats peut donc être difficile.

10.1.2. Diamètre de Féret

Un paramètre souvent utilisé est la mesure du **diamètre de Féret**. Elle consiste à mesurer l'étendue spatiale d'une particule dans une direction donnée (Figure 10.2). On la trouve parfois orthographiée « diamètre de Feret », ou « diamètre de Ferret », mais la version avec accent semble la plus ancienne, voir par exemple [2].

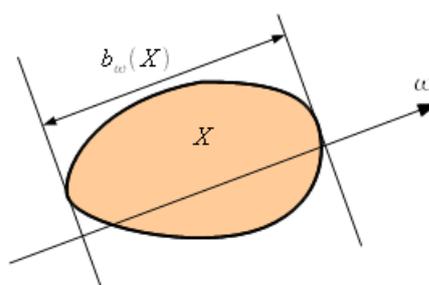


Figure 10.2. – Diamètre de Féret d'une particule mesuré dans la direction ω

En répétant la mesure pour différentes directions, on obtient une fonction qui associe un diamètre à une direction entre 0 et 180°. On peut résumer cette fonction en différents paramètres :

- diamètre de F eret maximum
- diamètre de F eret minimum
- diamètre de F eret dans la direction orthogonale au maximum
- diamètre de F eret moyenn e sur l'ensemble des directions (si la particule est convexe, on obtient une valeur proportionnelle au p erim etre)

Cette mesure peut facilement s' etendre aux particules observ ees dans des images 3D.

10.1.3. Moments g eom etriques et ellipse d'inertie

Les moments de l'image correspondent  a une int egrale sur les valeurs des pixels, pond er ee par leur position. On d efinit le **moment g eom etrique** de degr e (p, q) par

$$m_{p,q} = \sum_x \sum_y x^p y^q I(x, y) \quad (10.1)$$

Pour une image binaire, la valeur $I(x, y)$ vaut 1 si le point (x, y) appartient  a la particule, et 0 sinon. Le moment de degr e $(0, 0)$ correspond simplement  a l'aire de la particule. Les moments m_{10} et m_{01} , apr es normalisation par le moment m_{00} , correspondent aux coordonn ees du **centro ide**, ou centre de gravit e de la particule :

$$\begin{aligned} x_c &= \frac{m_{10}}{m_{00}} \\ y_c &= \frac{m_{01}}{m_{00}} \end{aligned}$$

Ellipse d'inertie  equivalente

En pratique, on utilise rarement les moments, mais plut ot des valeurs d eriv ees qui permettent de calculer l'inertie de la particule. On utilise en g en eral les moments d'ordre 2, qui permettent de d efinir une ellipse  equivalente ayant les m emes moments que la particule. La longueur du grand axe et du petit axe, ainsi que l'orientation de l'ellipse, permettent ainsi de synth etiser la forme globale de la particule (Figure 10.3).

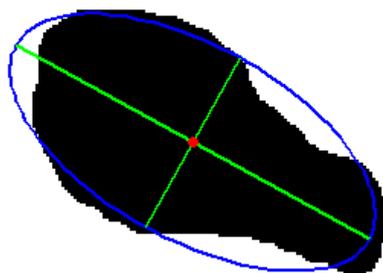


Figure 10.3. – Ellipse d'inertie d'une particule, ainsi que le centro ide et les axes principaux

Les paramètres synthétiques obtenus sont :

- la position du centre d'inertie
- la longueur du grand axe
- la longueur du petit axe
- l'orientation de l'ellipse
- facteur d'allongement (rapport des longueurs des axes), qui permet de décrire la forme générale de la particule (1 : forme ronde, valeur élevée : forme allongée)

Moments normalisés et Invariants

Il est souvent plus pratique de manipuler les moments centrés :

$$\mu_{p,q} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q I(x, y) \quad (10.2)$$

Enfin, pour s'affranchir de l'effet de la taille de la particule on considère en général les moments normalisés, en divisant par le moment μ_{00} :

$$\bar{\mu}_{p,q} = \frac{\mu_{p,q}}{\mu_{00}^{(p+q+2)/2}}$$

Il est ensuite possible de définir des paramètres normalisés, appelés invariants, qui décrivent la forme de la particule. Les invariants couramment utilisés sont les invariants de Hu. Les invariants de Hu couramment utilisés sont au nombre de 7, et restent identiques (aux erreurs de discrétisation près) après translation, changement de taille ou rotation d'une particule donnée.

$$\begin{aligned} H_1 &= \bar{\mu}_{20} + \bar{\mu}_{02} \\ H_2 &= (\bar{\mu}_{20} - \bar{\mu}_{02})^2 + 4\bar{\mu}_{11}^2 \\ H_3 &= (\bar{\mu}_{30} - 3\bar{\mu}_{12})^2 + (\bar{\mu}_{21} - \bar{\mu}_{03})^2 \\ H_4 &= (\bar{\mu}_{30} + \bar{\mu}_{12})^2 + (\bar{\mu}_{21} + \bar{\mu}_{03})^2 \\ H_5 &= (\bar{\mu}_{30} - 3\bar{\mu}_{12}) \cdot (\bar{\mu}_{30} + \bar{\mu}_{12}) \cdot \left[(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - 3(\bar{\mu}_{21} + \bar{\mu}_{03})^2 \right] \\ &\quad + (3\bar{\mu}_{21} - \bar{\mu}_{03}) \cdot (\bar{\mu}_{21} + \bar{\mu}_{03}) \cdot \left[3(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - (\bar{\mu}_{21} + \bar{\mu}_{03})^2 \right] \\ H_6 &= (\bar{\mu}_{20} + \bar{\mu}_{02}) \cdot \left[(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - (\bar{\mu}_{21} + \bar{\mu}_{03})^2 \right] \\ &\quad + 4\bar{\mu}_{11} \cdot (\bar{\mu}_{30} + \bar{\mu}_{12})^2 \cdot (\bar{\mu}_{21} + \bar{\mu}_{03})^2 \\ H_7 &= (3\bar{\mu}_{21} - \bar{\mu}_{03}) \cdot (\bar{\mu}_{30} + \bar{\mu}_{12}) \cdot \left[(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - 3(\bar{\mu}_{21} + \bar{\mu}_{03})^2 \right] \\ &\quad + (3\bar{\mu}_{12} - \bar{\mu}_{30}) \cdot (\bar{\mu}_{21} + \bar{\mu}_{03}) \cdot \left[3(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - (\bar{\mu}_{21} + \bar{\mu}_{03})^2 \right] \end{aligned}$$

Les notations ci-dessus proviennent de [1]. Voir aussi [6]. Comme la plage de variation de ces paramètres est souvent différente, on utilise couramment leur logarithme.

10.1.4. Paramètres géodésiques

Diamètre géodésique

Pour des particules de forme complexe, le diamètre ne reflète pas vraiment la taille réelle de la particule. On peut chercher à caractériser la plus grande distance géodésique (calculée en restant à l'intérieur de la particule) entre deux points appartenant à la particule.

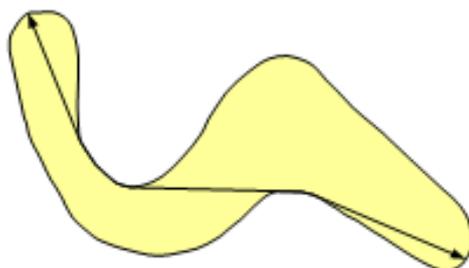


Figure 10.4. – Diamètre géodésique d'une particule

Ce paramètre est particulièrement bien adapté pour les particules non convexes, de forme allongées, ou très ramifiées. Dans ce dernier cas, le diamètre géodésique correspond à la longueur de la plus grande ramification.

Tortuosité

Un paramètre morphologie intéressant est la tortuosité, que l'on peut définir comme le rapport de la longueur géodésique et du diamètre de Feret de la particule. Ce paramètre vaut environ 1 pour des formes convexes, et augmente pour des formes allongées.

Allongement géodésique

Il est possible de calculer un **allongement géodésique** de la particule. On peut utiliser plusieurs formules :

- le rapport du carré de la longueur géodésique et de l'aire de la particule
- le rapport de la longueur géodésique et de l'épaisseur de la particule

10.1.5. Compacité

La **compacité** (appelée aussi **convexité**, ou encore **solidité**) permet de discriminer les particules globalement convexes de celles ayant une forme plus complexe. On commence par calculer l'enveloppe convexe de la particule, qui correspond au plus petit domaine convexe contenant entièrement la particule d'origine. La compacité est alors définie comme le rapport de l'aire de la particule sur l'aire de son enveloppe convexe.

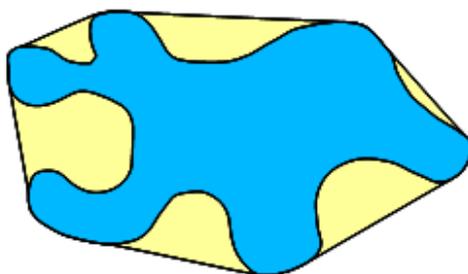


Figure 10.5. – Compacité d'une particule

Les particules convexes ou quasi-convexes auront une compacité proche de 1, tandis que des particules aux formes complexes verront leur compacité décroître.

10.1.6. Comparaison des facteurs de forme

Le tableau 10.1 présente les valeurs de quelques facteurs de forme pour des particules ayant des formes contrastées : un œuf, une ellipse, une forme étoilée, une fibre courbe.

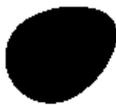
				
	œuf	ellipse	étoile	fibre
Circularité	0.966	0.834	0.395	0.280
Compacité	0.973	0.961	0.685	0.556
Allongement	1.29	1.99	1.00	2.63
Tortuosité	1.06	1.06	1.04	1.26

TABLE 10.1. – Facteurs de forme de quelques particules contrastées. Tous les paramètres présentés valent 1 dans le cas d'un disque.

On constate qu'en fonction de la nature des particules étudiées, certains paramètres sont plus pertinents que d'autres. Ainsi, la compacité de l'œuf et de l'ellipse ne permet pas de les différencier d'un disque, au contraire de l'allongement. Par contre, pour une forme étoilée, l'allongement est quasiment identique à celui d'un disque. Dans ce cas, la compacité ou la circularité sont des paramètres plus utiles. Enfin, pour des particules en forme de fibres, la tortuosité est un bon indicateur de l'allongement.

10.2. Analyse de particules en 3D

On s'intéresse ici à un ensemble de particules observées en 3D. On dispose là aussi de plusieurs paramètres descriptifs.

Notons que dans de nombreux cas on observe de nombreuses particules qui touchent les bords. C'est le cas par exemple dans des images de tissus végétaux en microscopie confocale. Si on les supprime de l'analyse, on risque de perdre de l'information. Si on les garde, on ne mesure pas les paramètres sur la totalité des particules.

10.2.1. Paramètres géométriques

Il existe quatre paramètres fondamentaux pour caractériser des particules ou des ensembles en 3D : le volume, la surface, l'épaisseur moyenne et la caractéristique d'Euler-Poincaré.

Volume

Le **volume** joue le même rôle que l'aire en deux dimension. On le mesure en comptant le nombre de pixels (ou voxels) appartenant à la particule, et en multipliant par le volume élémentaire d'un voxel.

Aire de surface

La mesure de **l'aire de surface** nécessite de prendre les mêmes précautions que pour le périmètre en 2D. Une première méthode consiste à calculer une représentation surfacique de la particule sous la forme d'un maillage triangulaire, puis à calculer la somme des aires de chaque triangle.

Comme pour le périmètre, il est aussi possible d'estimer la surface en utilisant la **formule de Crofton**. Le principe consiste à calculer le nombre d'intersections avec des droites discrètes dans l'image. On peut facilement calculer les intersections pour des ensembles de 3, 9 ou 13 directions dans l'image.

Épaisseur moyenne

L'épaisseur moyenne est définie pour des particules lisses comme une intégrale de courbure moyenne. En pratique, pour une particule convexe, elle est proportionnelle à la largeur moyenne en fonction des directions, et pour une fibre allongée, elle est proportionnelle à la longueur. C'est donc un paramètre morphologique complémentaire du volume et de la surface.

Il se mesure lui aussi en faisant appel à la formule de Crofton, mais cette fois en considérant des plans discrets dans l'image d'origine, et en mesurant le nombre de composantes connexes dans les plans.

Caractéristique d'Euler-Poincaré

La caractéristique d'Euler-Poincaré (appelée aussi nombre d'Euler, ou connexité) permet de caractériser la topologie d'une structure. C'est une valeur entière, sans dimension (c'est à dire sans unité).

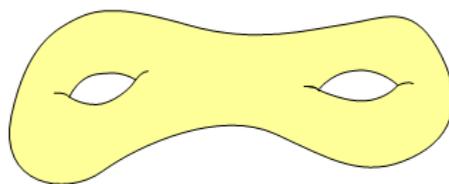


Figure 10.6. – La caractéristique d’Euler-Poincaré d’une structure avec deux anses est égale à -1 (+1 pour le nombre de composantes connexes, -2 pour le nombre d’anses)

Pour une particule connexe, la caractéristique d’Euler-Poincaré vaut 1. Pour une particule avec des trous, ou des anses (comme pour une tasse à café), la caractéristique d’Euler-Poincaré est égale à 1 moins le nombre de tunnels. Si on a des « bulles » (espaces vides complètement à l’intérieur de la particule), on ajoute 1 par bulle. Enfin, on peut aussi considérer cette caractéristique pour un ensemble de particules, dans ce cas elle est égale à la somme des caractéristiques de chaque particule individuelle.

10.2.2. Paramètres de forme

Comme en 2D, il est possible de définir des paramètres de forme pour des particules 3D, en se basant sur des rapports de mesures de taille. Une première famille de paramètres de forme se base sur les rapports normalisés du volume V , de la surface S , et de l’épaisseur moyenne B . En exprimant les paramètres entre 0 et 1, la valeur 1 correspondant à une particule complètement sphérique, on obtient les paramètres suivants :

$$\begin{aligned} f_1 &= 36\pi \frac{V^2}{S^3} \\ f_2 &= \frac{6}{\pi} \frac{V^2}{S^3} \\ f_3 &= \frac{1}{\pi} \frac{S}{B^2} \end{aligned}$$

Ces paramètres ont les mêmes défauts que le déficit isopérimétrique : ils sont très sensibles aux erreurs de mesures faites sur la surface ou l’épaisseur moyenne, et leur interprétation est difficile. Une autre famille de paramètres se base sur les rapports des longueurs des axes de l’ellipsoïde d’inertie.

$$\begin{aligned} r_{12} &= r_1/r_2 \\ r_{13} &= r_1/r_3 \\ r_{23} &= r_2/r_3 \end{aligned}$$

L’avantage des paramètres de forme basés sur l’ellipsoïde d’inertie est qu’ils sont assez peu sensibles aux erreurs de segmentation sur la frontière pour décrire la forme générale de particules.

10.3. Microstructure

Dans certains types d'images on observe une structure plus complexe qu'il est difficile d'assimiler à un ensemble de particules individuelles : mie de pain, mousses solides, agrégats...

Il est possible de quantifier des images de microstructure en adaptant certains paramètres morphométriques classiques. En 2D, on utilise par exemple le périmètre et l'aire, mais on ne mesure que la fraction de structure ou de frontière visible dans les images, que l'on ramène à l'aire de la zone observée. Il est ainsi possible de mesurer une **densité de périmètre** ou une **densité d'aire**.

La même stratégie est envisageable en 3D. On mesure cette fois la densité volumique, la densité de surface, ou encore la densité d'épaisseur moyenne.

10.4. Analyse de texture

10.4.1. Matrices de cooccurrence

Démarche :

1. Choix d'un vecteur (ex : +2;-1)
2. Construction des matrices
3. Paramètres synthétiques

Possibilité d'étudier l'évolution de certains paramètres quand on fait varier la longueur du vecteur d'étude.

10.4.2. Granulométrie

Idée : appliquer des filtres définis par un élément structurant de forme donnée, et faire varier la taille du filtre. On évalue ensuite les variations de volume de niveaux de gris pour construire une courbe granulométrique, qui décrit la texture visuelle de l'image.

10.5. Graphe d'adjacence des régions

Lorsque l'image représente un ensemble de régions disjointes mais adjacentes, on peut vouloir décrire les relations de voisinage entre les régions. Une première classe d'applications concerne les algorithmes de division-fusion (*split-and-merge*) : on procède d'abord à une division très fines de l'image (*quad-tree*, ligne de partage des eaux...), puis on cherche à fusionner les régions adjacentes.

Pour des applications d'analyse, on peut vouloir utiliser la topologie du graphe (nombre moyen de voisins...) pour décrire l'organisation spatiale des objets dans l'image (voir la figure suivante pour un exemple de description de cellules végétales).

L'algorithme le plus simple pour obtenir le graphe d'adjacence consiste à considérer chaque région tour à tour, à calculer la dilatation morphologique de cette région avec un élément structurant arbitraire, et à calculer l'intersection du dilaté avec les régions

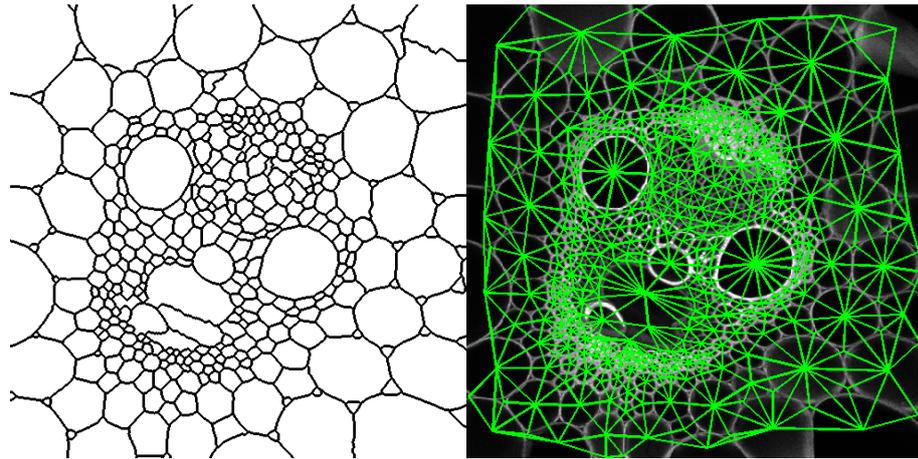


Figure 10.7. – Segmentation d’une image d’entrenoed de maïs, et graphe d’adjacence des régions montrant les relations de voisinages entre les cellules.

voisines. En répétant l’opération pour chaque région, on arrive à reconstruire le graphe d’adjacence des régions.

Une autre solution consiste à parcourir les lignes horizontales et verticales de l’image, et à détecter les transitions entre deux régions différentes. Dans le cas d’une segmentation produisant une frontière entre les régions, il faut évidemment tenir compte de l’épaisseur de cette frontière.

Bibliographie

- [1] Wilhem Burger and Mark J. Burge. *Digital Image Processing, An algorithmic introduction using Java*. Springer, 2008.
- [2] Michel Coster and Jean-Louis Chermant. *Précis d'analyse d'images*. CNRS PLUS. Presses du CNRS, 1989.
- [3] W. E. Lorensen and H. E. Cline. Marching cubes : a high-resolution 3D surface construction algorithm. In *Proceedings of the 14th ACM SIGGRAPH on Computer Graphics*, volume 21, pages 163–169, 1987. la référence du Marching cube.
- [4] Jean Serra. *Image Analysis and Mathematical Morphology. Volume 1*. Academic Press, London, 1982.
- [5] Pierre Soille. *Morphological Image Analysis*. Springer, 2nd edition, 2003.
- [6] Jeffrey Wood. Invariant pattern recognition : A review. *Pattern Recognition Lett.*, 29(1) :1–17, 1996.