



**UNIVERSITÉ  
DE LORRAINE**

**BIBLIOTHÈQUES  
UNIVERSITAIRES**

## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : [ddoc-theses-contact@univ-lorraine.fr](mailto:ddoc-theses-contact@univ-lorraine.fr)  
*(Cette adresse ne permet pas de contacter les auteurs)*

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

# Éléments pour l'analyse et le traitement d'images : application à l'estimation de la qualité du bois

## THÈSE

présentée et soutenue publiquement le 5 juillet 2022

pour l'obtention du

**Doctorat de l'Université de Lorraine**

**(Mention Informatique)**

par

Rémi Decelle

### Composition du jury

*Présidents :*

Laure TOUGNE

Professeure, LIRIS, Université Lumière Lyon 2

*Rapporteur & Rapporteuse :*

Fabien FESCHET

Professeur, LIMOS, Université de Clermont Auvergne

Laure TOUGNE

Professeure, LIRIS, Université Lumière Lyon 2

*Examineurs :*

Philippe BORIANNE

Ingénieur de Recherche, Cirad

Erwan KERRIEN

Chargé de Recherche, INRIA, LORIA, Université de Lorraine

*Invités :*

Bertrand KERAUTRET

Professeur des universités, LIRIS, Université Lumière Lyon 2

Frédéric MOTHE

Chargé de Recherche, Silva, Université de Lorraine

Phuc NGO

Maître de Conférences, LORIA, Université de Lorraine

*Encadrantes :*

Isabelle DEBLED-RENNESON

Professeure, LORIA, Université de Lorraine

Fleur LONGUETAUD

Chargée de Recherche, Silva, Université de Lorraine

Mis en page avec la classe thesul.

## Remerciements

La thèse a pu être réalisée à l'aide du financement de l'Agence Nationale de la Recherche (ANR). Je remercie le laboratoire LORIA qui m'a accueilli durant plusieurs années afin de me former au mieux à travers les différents stages effectués. Bien entendu, parmi ces années, il y a ces quatre années de thèse qui m'ont permis de réaliser mes recherches. Je pense au laboratoire INRAE et à l'UMR Silva qui m'ont accueilli pendant cette thèse.

Au sein du LORIA, je remercie particulièrement Isabelle Debled-Rennesson pour son encadrement de la thèse, mais aussi de mon stage de Master. Elle m'a beaucoup aidé en me guidant dans mes recherches, en corrigeant de nombreuses fois les écrits que j'ai pu produire et en proposant des solutions face aux différentes difficultés que j'ai pu rencontrer. Je remercie également Phuc Ngo qui a participé à cet encadrement. Je remercie Bertrand Kerautret avec lequel j'ai travaillé durant mon stage de Master. C'est aussi en partie grâce à lui, et à son intérêt pour les nouvelles technologies développées par Apple, que l'iPhone X a pu être acheté (pour mon stage de Master) et pu servir pour l'acquisition des images utilisées dans cette thèse. Je pense aussi à Florian Delconte, doctorant au sein de l'équipe ADAGIo, avec lequel j'ai partagé le bureau et avec qui j'ai pu échanger sur des sujets techniques et divers.

Au sein de Silva, je tiens à remercier Fleur Longuetaud pour son co-encadrement. Elle m'a permis de découvrir la biologie du bois et d'agrandir mes connaissances sur le bois de manière générale. Je la remercie pour avoir corrigé les erreurs que j'ai pu commettre que ce soit sur le français que sur la qualité du bois (particulièrement sur le mot cerne que j'ai mis trop régulièrement au féminin et non au masculin). Compagnon chercheur de Fleur depuis des années, Frédéric Mothe fait partie de mes remerciements pour les mêmes raisons.

Que serait Silva sans les personnes du bâtiment B' ? Parmi ces personnes, je souhaite remercier Rodolphe Bauer et Antoine Billard qui ont été mes compagnons de thèses (qu'ils ont déjà soutenues) durant ces quatre années. Je pense aussi à Philippe Jacquin, ingénieur de recherche, et à ses problèmes de voitures et de police.

Évidemment, je souhaite remercier toutes les personnes avec qui j'ai pu échanger et discuter au cours de cette thèse. Je pense aux techniciens et techniciennes de l'ombre qui ont aidé de près ou de loin durant cette thèse. Je pense également aux nombreux et nombreuses doctorants et doctorantes de Silva et du LORIA. Je pense également à Nathalie Bissieux, référente des ressources humaines pour l'équipe Silva, qui m'a beaucoup aidé dans les difficultés administratives (notamment celles avec l'Université de Lorraine et les contrats de DCCE/vacataire). Je pense aussi à Sarah-Louise Filleux (qui était chargée de communication) avec qui j'ai et j'aime beaucoup discuter (sur le cinéma, l'art, simplement refaire le monde, ...).

Je remercie aussi ma famille pour m'avoir soutenu et je remercie Valérian Accary pour m'avoir aussi soutenu, non seulement durant ces quatre années de thèses, mais également aux années précédentes.



# Table des matières

<b>Table des figures</b>	<b>7</b>
<b>Liste des tableaux</b>	<b>13</b>
<b>Introduction</b>	<b>15</b>
<b>1 Techniques de segmentation</b>	<b>23</b>
1.1 Introduction . . . . .	24
1.1.1 Représentation de la couleur et de l'intensité . . . . .	24
1.2 Objectifs, propriétés et évaluation de la segmentation . . . . .	27
1.2.1 Les objectifs d'une segmentation . . . . .	29
1.2.2 Évaluation de la segmentation . . . . .	29
1.2.3 Matrice de confusion . . . . .	30
1.3 Seuillage et histogramme . . . . .	31
1.3.1 Seuillage et histogramme . . . . .	31
1.3.2 Méthode d'Otsu . . . . .	32
1.3.3 Seuillage par hystérésis . . . . .	33
1.4 Approche par gradient . . . . .	34
1.4.1 Gradient d'une image . . . . .	35
1.4.2 Convolution et dérivée . . . . .	35
1.4.3 Canny . . . . .	36
1.5 Accroissement de régions . . . . .	37
1.5.1 Décomposition en régions . . . . .	38
1.5.2 Fusion des régions . . . . .	39
1.6 Contours actifs (ou Snakes) . . . . .	40
1.6.1 Énergie potentielle . . . . .	41
1.6.2 Énergie interne . . . . .	41

1.6.3	Énergie externe . . . . .	42
1.7	Apprentissage automatique - Réseaux de neurones . . . . .	42
1.7.1	Les principes de base . . . . .	43
1.7.2	Fonctions d'activation . . . . .	44
1.7.3	Apprentissage d'un neurone . . . . .	46
1.7.4	Régularisation d'un réseau de neurones . . . . .	47
1.7.4.1	<i>Dropout</i> et <i>Shakeout</i> . . . . .	48
1.7.4.2	Couche de normalisation . . . . .	48
1.7.4.3	Augmentation de données . . . . .	49
1.7.5	Réseaux de neurones convolutifs . . . . .	49
1.7.5.1	Couche de convolution . . . . .	50
1.7.5.2	Couche de normalisation dans le cadre d'un réseau convolutif	50
1.7.5.3	Couche d'échantillonnage - <i>Pooling</i> . . . . .	51
1.7.5.4	Couche de suréchantillonnage - <i>Upsampling</i> . . . . .	52
1.7.5.5	Couche d'attention - <i>Attention</i> . . . . .	53
1.7.6	Exemple de réseaux de neurones . . . . .	55
1.8	Résumé . . . . .	60
<b>2</b>	<b>Segmentation : les applications aux bois</b>	<b>63</b>
2.1	Introduction . . . . .	64
2.2	L'anatomie du bois . . . . .	64
2.2.1	Les cernes . . . . .	64
2.2.2	Orientation du bois . . . . .	65
2.2.3	Fonctions du bois . . . . .	66
2.2.4	L'aubier et le duramen . . . . .	67
2.3	Qualité du bois . . . . .	68
2.4	Segmentation de la grume . . . . .	68
2.4.1	Schraml et Uhl (2014) [ <a href="#">Schraml and Uhl, 2014</a> ] . . . . .	69
2.4.2	Wimmer et al. (2021) [ <a href="#">Wimmer et al., 2021</a> ] . . . . .	72
2.5	Segmentation du duramen . . . . .	72
2.5.1	Raatevaara et al. (2020) [ <a href="#">Raatevaara et al., 2020</a> ] . . . . .	72
2.6	Détection des cernes annuels . . . . .	74
2.6.1	Conner et al. (1998) [ <a href="#">Conner et al., 1998</a> ] . . . . .	74
2.6.2	Soille et Misson (2001) [ <a href="#">Soille and Misson, 2001</a> ] . . . . .	75
2.6.3	Laggoune et al. (2005) [ <a href="#">Laggoune et al., 2005</a> ] . . . . .	76

---

2.6.4	Cerda et al. (2007) [ <a href="#">Cerda et al., 2007</a> ] . . . . .	78
2.6.5	Norell (2011) [ <a href="#">Norell, 2011</a> ] . . . . .	79
2.6.6	Zhou et al. (2012) [ <a href="#">Zhou et al., 2012</a> ] . . . . .	80
2.6.7	Sundari et al. (2014) [ <a href="#">Sundari and Kumar, 2014</a> ] . . . . .	81
2.6.8	Kennel et al. (2015) [ <a href="#">Kennel et al., 2015</a> ] . . . . .	83
2.6.9	Fabijańska et Danek (2018) [ <a href="#">Fabijańska and Danek, 2018</a> ] . . . . .	84
2.7	Conclusion . . . . .	86
<b>3</b>	<b>Nouvelles approches du traitement d'images appliquées au bois</b>	<b>89</b>
3.1	Introduction . . . . .	90
3.2	Jeux de données . . . . .	90
3.2.1	Capteurs utilisés . . . . .	91
3.2.2	Corcieux . . . . .	92
3.2.3	FVA . . . . .	93
3.2.4	Besle . . . . .	93
3.2.5	BBF . . . . .	93
3.2.6	Logyard . . . . .	94
3.2.7	Sawmill . . . . .	95
3.3	Segmentation de la grume . . . . .	96
3.3.1	K-means . . . . .	97
3.3.2	Contours actifs . . . . .	100
3.3.3	Réseaux de neurones convolutifs . . . . .	101
3.3.4	Résultats . . . . .	102
3.4	Segmentation de l'aubier et du duramen . . . . .	106
3.4.1	Réseaux de neurones convolutifs . . . . .	107
3.4.1.1	Couche d'attention morphologique . . . . .	107
3.4.1.2	Architecture proposée . . . . .	109
3.4.2	Résultats . . . . .	110
3.5	Détection des cernes . . . . .	111
3.5.1	Filtrage directionnel . . . . .	113
3.5.2	Filtrage directionnel : application au bois . . . . .	115
3.5.3	Seuillage . . . . .	117
3.5.4	Affinement du seuillage . . . . .	117
3.5.5	Calcul de la largeur moyenne . . . . .	122
3.5.5.1	Décomposition en quadtree . . . . .	122



3.5.5.2	Calcul de la qualité . . . . .	122
3.5.5.3	Largeur moyenne des cernes . . . . .	127
3.5.6	Résultats . . . . .	127
3.6	Synthèse . . . . .	128
<b>4</b>	<b>Détection de la moelle dans l'image</b>	<b>131</b>
4.1	État de l'art . . . . .	132
4.1.1	Norell et Borgfors (2008) [ <a href="#">Norell and Borgfors, 2008</a> ] . . . . .	132
4.1.1.1	Filtres en quadrature . . . . .	133
4.1.1.2	Pyramide Laplacienne . . . . .	135
4.1.1.3	Résultats . . . . .	136
4.1.2	Schraml et Uhl (2013) [ <a href="#">Schraml and Uhl, 2013</a> ] . . . . .	136
4.1.2.1	Calcul de l'orientation . . . . .	136
4.1.2.2	Création de l'espace d'accumulation . . . . .	137
4.1.2.3	Estimation de la position de la moelle . . . . .	138
4.1.2.4	Résultats . . . . .	138
4.1.3	Kurdthongmee et al. (2018) [ <a href="#">Kurdthongmee et al., 2018</a> ] . . . . .	138
4.1.3.1	Calcul de l'orientation . . . . .	139
4.1.3.2	Création de l'espace d'accumulation . . . . .	139
4.1.3.3	Estimation de la position de la moelle . . . . .	140
4.1.3.4	Résultats . . . . .	140
4.1.4	Kurdthongmee (2020) [ <a href="#">Kurdthongmee, 2020</a> ] . . . . .	141
4.2	Approche proposée . . . . .	142
4.2.1	Pré-traitement . . . . .	144
4.2.1.1	Conversion en niveaux de gris . . . . .	144
4.2.1.2	Suppression de coups de scies . . . . .	144
4.2.1.3	Calcul de l'orientation . . . . .	145
4.2.2	Application à la détection de la moelle . . . . .	147
4.2.2.1	Initialisation . . . . .	148
4.2.2.2	Construction de la matrice de transition probabiliste . . . . .	148
4.2.2.3	Mise à jour de la carte de phéromones . . . . .	149
4.2.3	Dépôt des phéromones . . . . .	149
4.2.4	Extraction de la position de la moelle . . . . .	150
4.3	Résultats . . . . .	151
4.4	Analyse de la méthode proposée . . . . .	153

---

4.4.1	Analyse de la complexité . . . . .	153
4.4.2	Effet des paramètres sur la convergence de l'algorithme . . . . .	155
4.5	Synthèse . . . . .	158
<b>5</b>	<b>Filtre directionnel discret</b>	<b>161</b>
5.1	Introduction . . . . .	162
5.2	Notions de base . . . . .	165
5.2.1	Droites et segments discrets . . . . .	165
5.2.2	Arbre de Stern-Brocot et suite de Farey [ <a href="#">G. Hardy, 1989</a> , <a href="#">Stern, 1858</a> ] . . . . .	166
5.3	Filtre directionnel : le cas binaire . . . . .	167
5.3.1	Calcul des segments naïfs symétriques . . . . .	167
5.3.2	Complexité en temps . . . . .	168
5.3.3	Application à l'estimation de caractéristiques . . . . .	169
5.4	Filtre directionnel : image en niveaux de gris . . . . .	172
5.4.1	Première approche . . . . .	173
5.4.2	Inclusion de l'épaisseur . . . . .	173
5.4.3	Simplification du calcul pour l'épaisseur . . . . .	176
5.4.4	Estimation de l'orientation . . . . .	178
5.4.4.1	Principe du calcul . . . . .	178
5.4.4.2	Résultats préliminaires . . . . .	179
5.4.5	Application pour le filtrage d'images . . . . .	180
5.5	Conclusion . . . . .	181
<b>6</b>	<b>Conclusion</b>	<b>183</b>
6.1	Contributions . . . . .	184
6.2	Perspectives . . . . .	186
	<b>Acronymes</b>	<b>191</b>
	<b>Glossaire</b>	<b>193</b>
	<b>Bibliographie</b>	<b>195</b>



# Table des figures

1	Exemple de données issues de différents type de capteurs, de gauche à droite : smartphone, scanner à rayons X et LIDAR après maillage (provenant de [Nguyen et al., 2018]). . . . .	18
2	Taux de boisement des départements (France hexagonale), d'après IGN (2020). . . . .	19
3	Localisation des scieries et production régionale de sciages en France hexagonale en 2018 [FCBA, 2020]. . . . .	20
1.1	Exemple de l'espace RVB. De gauche à droite : l'image d'origine en RVB, la composante rouge, la composante verte, la composante bleue. . . . .	25
1.2	Exemple de l'espace HSV. De gauche à droite : l'image d'origine en RVB, la composante H (dont l'histogramme a été égalisé afin de mieux visualiser cette composante), la composante S et la composante V. . . . .	26
1.3	Exemple de l'espace CieLAB. De gauche à droite : l'image d'origine en RVB, la composante $L^*$ , la composante $a^*$ , la composante $b^*$ . . . . .	27
1.4	Image d'une loutre à droite. Au centre, une segmentation possible. À droite une binarisation possible. . . . .	28
1.5	L'histogramme (à droite) de l'image en niveaux de gris (à gauche). . . . .	32
1.6	Résultat de la méthode d'Otsu pour segmenter une image. . . . .	33
1.7	Résultat de la méthode par hystérésis pour segmenter une image. . . . .	34
1.8	Image en niveaux de gris. Dérivées en X et Y à l'aide des masques de Sobel. . . . .	36
1.9	Résultat de la détection des contours par la méthode de Canny, $\sigma_{bas} = 150$ et $\sigma_{haut} = 220$ ont été choisis pour l'étape 5, le seuillage par hystérésis. . . . .	37
1.10	Résultat de la décomposition d'une image en superpixels à l'aide de l'algorithme SLIC. . . . .	39
1.11	À gauche, la région initialisée avec les superpixels du bord de l'image. À droite, la région finale obtenue par fusion avec pour vecteur descripteur la moyenne des composantes RVB et un seuil de 22. . . . .	40
1.12	À gauche, l'image en entrée (la taille a été divisée par 4 pour avoir une convergence plus rapide du contour). À droite, en bleu le contour initial (100 points) et en rouge le contour final obtenu avec $\alpha(s) = 0.03$ , $\beta(s) = 1.0$ , $\delta = 2$ et $\varepsilon = -4$ . . . . .	42
1.13	Schéma d'un neurone artificiel. . . . .	43
1.14	Exemple d'un réseau de neurones à propagation. . . . .	44

1.15	Exemples de couches convolutives. Les cases bleutées représentent les entrées, les cases blanches représentent les ajouts dus au <i>padding</i> , les cases violettes représentent les sorties après convolution. Celles de couleurs soulignent l'influence des paramètres. . . . .	51
1.16	Exemple du <i>max pooling</i> de taille $3 \times 3$ avec un pas de 2 et du <i>average pooling</i> avec les mêmes paramètres. . . . .	52
1.17	Interpolation bilinéaire d'un pixel $(i, j)$ . . . . .	53
1.18	Principe de la couche d'attention proposée par [Oktay et al., 2018]. . . . .	54
1.19	Architecture du réseau ResNet-50 d'après [He et al., 2016]. . . . .	56
1.20	Architecture du réseau SegNet d'après [Badrinarayanan et al., 2017]. . . . .	57
1.21	Architecture du réseau U-Net, d'après [Ronneberger et al., 2015] . . . . .	57
1.22	Architecture du réseau Mask-RCNN d'après [He et al., 2017]. . . . .	58
1.23	Architecture du réseau RefineNet d'après [Lin et al., 2017]. . . . .	59
1.24	Architecture du réseau YOLO V3. . . . .	60
2.1	Exemple de coupes transversales d'épicéa. La limite de cerne est visible à l'échelle microscope et macroscopique. . . . .	65
2.2	Les trois directions et plans possibles dans le bois, d'après [George et al., 2020]. . . . .	66
2.3	Exemple montrant la limite entre le duramen et l'aubier sur le Douglas, un bois à aubier différencié. Sur cette essence, le duramen prend une couleur rouge/rose. . . . .	67
2.4	Exemple des images utilisées dans l'article [Schraml and Uhl, 2014]. . . . .	69
2.5	Schéma du principe du calcul du motif binaire local (avec $Q = 8$ et $R = 1.0$ ) 70	
2.6	Exemple de la méthode sur deux blocs de taille $100 \times 100$ issus d'une image de coupe transversale. La distance entre ces deux blocs est de 1,352. . . . .	71
2.7	Post-traitement proposé par [Schraml and Uhl, 2014]. De gauche à droite : l'image avec les blocs appartenant à la région finale, les blocs au bord, ajustement du cercle (en rouge) et les blocs retirés (blocs rouges), ajustement de l'ellipse (en vert) et les blocs retirés (blocs verts) et approximation par une forme $\alpha$ (en bleu). . . . .	71
2.8	Exemple d'une image utilisée dans l'article [Raatevaara et al., 2020]. En (a) l'image d'origine, en (b) la segmentation manuelle de la grume, en (c) la vérité terrain (en blanc le duramen, en gris l'aubier). . . . .	73
2.9	Exemple d'une image utilisée dans l'article [Raatevaara et al., 2020]. En (a) la <i>trimap</i> obtenue à partir de la première segmentation, en (b) le résultat du <i>KNN-Matting</i> , en (c) la segmentation finale. . . . .	74
2.10	Image issue de l'article [Soille and Misson, 2001]. . . . .	75
2.11	Résumé de la méthode de [Soille and Misson, 2001] pour la détection des cernes. . . . .	77
2.12	Exemple d'image issue de l'article [Cerda et al., 2007]. . . . .	78
2.13	Exemple des cernes simulés d'après la méthode de [Cerda et al., 2007]. En (a) l'image en entrée, en (b) les cernes simulés d'après l'équation 2.2. . . . .	79
2.14	Illustration de la décomposition en sous-images proposée par [Norell, 2011]. 80	
2.15	Les images utilisées dans l'article [Zhou et al., 2012]. . . . .	81

---

2.16	Image extraite de [Kennel et al., 2015]. À gauche l'image en entrée, à droite la superposition avec les cernes détectés. . . . .	84
2.17	Image extraite de [Fabijańska and Danek, 2018]. Le trait indique la largeur de cerne. De gauche à droite, les essences sont sapin blanc européen, épicéa commun, chêne, frêne européen, hêtre européen et tilleul à petites feuilles orme. . . . .	85
2.18	Image extraite de [Fabijańska and Danek, 2018]. En a) la sortie du réseau de neurones, en b) le seuillage appliqué, en c) résultat après la squeletisation et en d) la superposition du résultat sur l'image d'entrée. . . . .	85
3.1	Exemples issus du jeu de données Corcieux avec le capteur Huawei PRA-LX1.	92
3.2	Exemples issus du jeu de données Corcieux avec le capteur Panasonic Lumix DMC-FZ45. . . . .	92
3.3	Exemples issus du jeu de données FVA. Il s'agit des mêmes grumes qu'aux figure 3.1 et figure 3.2. . . . .	93
3.4	Exemples issus du jeu de données Besle. La dernière image est celle de la souche. . . . .	94
3.5	Exemples issus du jeu de données BBF. Les mêmes grumes que celles montrées pour le jeu Besle (à l'exception de la dernière). . . . .	94
3.6	Schéma représentant les parties de la grume prises en photo aussi bien pour le jeu <b>Logyard</b> que pour le jeu <b>Sawmill</b> . . . . .	95
3.7	Exemples issus du jeu de données Logyard. Il s'agit des quatre section la même grume : D02a, D02b, D02c et D02d. . . . .	95
3.8	Schéma représentant l'installation du capteur en scierie pour le jeu de données de la scierie. . . . .	96
3.9	Exemples issus du jeu de données Sawmill. De gauche à droite : D02a, D02b, D02c et A01b. . . . .	96
3.10	Image d'exemple et les trois composantes de l'espace CieLAB dans l'ordre $L$ , $a^*$ et $b^*$ . . . . .	97
3.11	Les 6 clusters (image en noir et blanc) trouvés par la méthode du K-Means [Arthur and Vassilvitskii, 2006] et les composantes connexes associées (sous chaque clusters, une couleur par composante). . . . .	98
3.12	Exemple de l'algorithme proposée pour la segmentation de la grume en utilisant l'algorithme de KMeans [Arthur and Vassilvitskii, 2006] . . . . .	99
3.13	Contre-exemple montrant une limite du calcul de circularité. . . . .	100
3.14	Architecture de U-Net proposée pour la segmentation de la coupe transversale. La taille des convolutions est marquée au-dessus des flèches correspondantes. . . . .	102
3.15	Prédictions des modèles sur une image du jeu Corcieux. En vert, les pixels TP. En bleu, les pixels FN et en rouge les pixels FP. . . . .	103
3.16	Prédictions des modèles sur une image du jeu sbgTS3. En vert, les pixels TP. En bleu, les pixels FN et en rouge les pixels FP. . . . .	104
3.17	Image issue du jeu d'images <i>Sawmill</i> montrant les limites liées à l'image : présence d'une deuxième grume, en première ligne les composantes de l'espace CieLAB et en deuxième ligne les composantes de l'espace HSV. . . . .	105

3.18	Couche d'attention morphologique proposée. . . . .	109
3.19	Architecture du réseau LU-Net proposée. . . . .	109
3.20	Contour de la segmentation du duramen par les différents réseaux de neurones proposés. . . . .	112
3.21	Zoom d'images montrant les difficultés liées au type d'images acquises. . .	113
3.22	Illustration des masques pour un filtre dérivatif gaussien orientable. . . . .	114
3.23	Illustration de masques pour un filtre orientable de la dérivée seconde d'une gaussienne. . . . .	114
3.24	Deux exemples du filtrage bilatéral directionnel. À gauche, l'image en niveaux de gris. À droite, l'image filtrée. En haut à gauche de chaque image, un zoom du rectangle rouge. . . . .	116
3.25	Deux exemples du seuillage. À gauche, le seuillage adaptatif local avec la moyenne. À droite, le seuillage avec la suppression des petites composantes connexes. En haut à gauche de chaque image, un zoom du rectangle rouge. . . . .	118
3.26	Deux exemples de l'affinement du seuillage. À gauche, l'image seuillée. À droite, le seuillage affinée avec uniquement les transition noir/blanc le long d'un rayon partant de la moelle vers l'écorce. En haut à gauche de chaque image, un zoom du rectangle rouge. . . . .	119
3.27	Résultat final de la segmentation des limites de cernes. En haut à gauche de chaque image, un zoom du rectangle rouge. . . . .	120
3.28	Résultat final de la segmentation des limites de cernes. En haut à gauche de chaque image, un zoom du rectangle rouge. . . . .	121
3.29	Décomposition en quadtree avec $\epsilon = 0,99$ , $d_{min} = 4, d_{max} = 5$ . La couleur du nœud indique la qualité du nœud. L'orientation moyenne est représentée par le trait au centre d'un nœud. Les nœuds bleus sont ceux pour lesquels il n'y a pas eu de qualité estimée. . . . .	125
3.30	Décomposition en quadtree avec $\epsilon = 0,99$ , $d_{min} = 4, d_{max} = 5$ . La couleur du nœud indique la qualité du nœud. L'orientation moyenne est représentée par le trait au centre d'un nœud. Les nœuds bleus sont ceux pour lesquels il n'y a pas eu de qualité estimée. . . . .	126
3.31	Largeur moyenne estimée en fonction de la largeur moyenne de la vérité terrain. La droite représente $y = x$ . . . . .	127
4.1	Création d'un espace d'accumulation à l'aide des normales aux tangentes des cernes. . . . .	133
4.2	Images utilisées dans [Norell and Borgfors, 2008]. La première montre l'estimation de la première étape puis le résultat de la seconde étape. Dans les deux, il s'agit de la méthode utilisant la pyramide Laplacienne. . . . .	134
4.3	Résumé des deux méthodes proposées par [Norell, 2011] . . . . .	134
4.4	Deux exemples du type d'images traitées par [Schraml and Uhl, 2013]. . . . .	137
4.5	Exemples de couples image d'origine/FFT pour des blocs montrant des cernes de croissances. . . . .	137
4.6	Deux images issues du jeu d'images utilisé par [Kurdthongmee et al., 2018] et [Kurdthongmee and Suwannarat, 2019]. . . . .	139

---

4.7	Illustration des directions possibles en utilisant les HOGs avec un paramètre de 9 possibilités. À droite, en rouge les directions uniquement possibles selon la position dans l'image. . . . .	140
4.8	Craquelures présentes proches de la moelle. Ces craquelures apparaissent lors du séchage du bois. . . . .	142
4.9	Image illustrant la notion de champ d'orientation et de champs de vecteurs dans le cas de la détection de la moelle. L'étoile verte représente le point fixe recherché. . . . .	143
4.10	Les trois étapes de la méthode proposée. . . . .	144
4.11	Étapes du pré-traitement : (a) Niveaux de gris, (b) FFT, (c) FFT filtré, (d) FFT après seuillage, (e) droite détectée par l'ACP, (f) l'image de sortie.	146
4.12	Image d'entrée (à gauche) et le nuage de points correspondant lors du pré-traitement (au milieu) où les deux directions principales (flèches rouges) dans le domaine de Fourier sont non-orthogonales. L'ACP ne peut estimer ces deux directions. À droite, le résultat du calcul de l'orientation des cernes (segments rouges). Quelques orientations ne sont pas correctes, cela est dû à la suppression des coups de scies. . . . .	147
4.13	D'après [Decelle et al., 2021b], exemple du calcul de la matrice de transition probabiliste $\mathbf{p}$ . Le carré rouge représente la position actuelle de la fourmi. Après le calcul, la fourmi a une probabilité de 36,2% de se déplacer vers la position contenant le plus de phéromones (carrés verts). Le résultat est obtenu avec $\alpha = 2$ et $\beta = 1$ . . . . .	149
4.14	À gauche, image d'origine avec les fourmis en croix jaunes, l'estimation courante de la moelle en rouge et un ensemble de $n = 3$ blocs de taille $\omega = 8$ (grille rouge). Les normales servant d'accumulation sont les lignes vertes. En haut à droite, un zoom sur le rectangle rouge. À droite, la matrice de phéromones. Une position $(u, v)$ est égal à $m = 5$ pixels en coordonnées image. . . . .	150
4.15	Boîtes à moustaches représentant la distance entre l'estimation et la vérité terrain (en mm) pour les méthodes [Schraml and Uhl, 2013, Kurdthongmee et al., 2018, Kurdthongmee, 2020] et notre méthode. . . . .	152
4.16	Deux exemples où notre méthode n'arrive pas à moins de 5 mm de la vérité terrain. La vérité terrain est la croix blanche, la croix verte est le résultat du réseau YOLO v3 [Kurdthongmee, 2020], celle en bleue montre la moelle détectée par [Schraml and Uhl, 2013], et notre méthode est représentée par la croix rouge. . . . .	153
4.17	Temps de calcul pour une itération selon la taille (nombre de pixels) de l'image. . . . .	154
4.18	Temps de calcul pour une itération selon (a) le nombre de fourmis $K$ , (b) la taille de $\tau$ , (c) la taille $\omega$ d'un bloc et (d) le taille $n$ du nombre de blocs.	155
4.19	Variations selon le paramètre $K$ entre la position de la moelle à l'instant $t$ et celle à l'instant $t + 1$ . . . . .	156
4.20	Variations selon le paramètre $n$ entre la position de la moelle à l'instant $t$ et celle à l'instant $t + 1$ . . . . .	156



4.21	Variations selon le paramètre $\omega$ entre la position de la moelle à l'instant $t$ et celle à l'instant $t + 1$ . . . . .	157
4.22	Variations selon le paramètre $m$ entre la position de la moelle à l'instant $t$ et celle à l'instant $t + 1$ . . . . .	157
5.1	Processus d'estimation des caractéristiques d'un cerne de croissance : détection des limites de cernes, labellisation, analyse d'un cerne en tant qu'objet discret, estimation de la longueur, de l'épaisseur et de l'orientation $\theta$ . . . . .	162
5.2	Images issues de l'article [Soille et al., 1996]. Ouverture morphologique par des segments de droite discrète de 15 pixels. (a) Image d'origine (de taille $66 \times 56$ pixels). Résultat de l'ouverture par l'élément structurant : (b) $DSS(5, -1, 0)$ , (c) $DSS(5, -1, -1)$ , (d) $DSS(5, -1, -2)$ , (e) $DSS(5, -1, -3)$ et (f) $DSS(5, -1, -4)$ . . . . .	164
5.3	Segment naïf de taille 15 (pixels) pour $x \in [0, 14]$ et les segments de droites $DSS(4, 11, 0)$ (blanc), $DSS(4, 11, 11)$ (gris clair), $DSS(4, 11, -11)$ (gris foncé). Le segment épais $TDSS_{1,1}(4, 11)$ contient tous les pixels pour $x \in [0, 14]$ . La valeur $4x - 11y$ est indiquée pour chaque pixel $(x, y)$ . . . . .	165
5.4	Répartition des directions obtenues par les suite de Farey ( $\arctan \frac{a}{b}$ ). . . . .	167
5.5	Exemple des segments naïfs symétriques construits pendant l'algorithme. . . . .	168
5.6	Exemple, pas à pas, du cas particulier de segment naïf symétrique. En vert, les fractions $(a, b)$ des segments pouvant être étendus. En rouge, celles ne pouvant plus être étendus. . . . .	170
5.7	En vert clair le segment symétrique $DSS(5, 4, -4)$ le plus long passant par le pixel en bleu. Le segment en vert foncé est le segment $TDSS_{5,4}(1, 0)$ . . . . .	171
5.8	Éléments structurants pour obtenir la transformée $\mathcal{E}$ selon les coordonnées $(i, j)$ (de la transformée). . . . .	174
5.9	Exemples des transformées en niveaux de gris. Un point rouge sur les images indique où la transformée a été appliquée. . . . .	175
5.10	Illustration de l'approche proposée avec des variations de longueurs. . . . .	177
5.11	Relation en dilatation morphologique et segment épais. . . . .	178
5.12	Résultats de notre méthode et celle de [Soille and Talbot, 2001]. (a) Image de cercles concentriques (b) Vérité terrain. (c) [Soille and Talbot, 2001]. (d) Variations d'épaisseur. (e) Variations de longueur. (f) Les deux variations. . . . .	179

# Liste des tableaux

1.1	Exemples de fonctions d'activation. . . . .	45
1.2	Résumé des méthodes vues au cours de ce chapitre avec leurs principaux avantages et inconvénients. . . . .	62
2.1	Résumé des méthodes vues au cours de ce chapitre avec des avantages et limites pour notre cas d'études. . . . .	87
3.1	Spécifications des capteurs utilisés pour les jeux de données . . . . .	91
3.2	Jeux de données, capteurs correspondant et nombre d'images. . . . .	91
3.3	MCC pour les différents modèles sur les jeux de données utilisés. En caractères gras, le MCC le plus élevé. . . . .	103
3.4	Temps de calcul moyen (en ms) pour une image pour les différents modèles. . . . .	105
3.5	Valeur du MCC sur une validation croisée avec 8 groupes pour chaque modèle et les deux jeux d'images utilisés. La deuxième colonne (#) est le nombre de paramètres du réseau. . . . .	111
3.6	Temps de calcul moyen pour une image pour les différents modèles en ms. . . . .	111
4.1	Moyenne, écart-type, minimum et maximum entre les vérités de terrain et l'estimation de la position de la moelle en pixels d'après [ <a href="#">Kurdthongmee, 2020</a> ]. . . . .	142
4.2	Moyenne, écart-type, minimum et maximum entre les vérités de terrain et l'estimation de la position de la moelle pour notre méthode et les méthodes de [ <a href="#">Schraml and Uhl, 2013</a> , <a href="#">Kurdthongmee et al., 2018</a> , <a href="#">Kurdthongmee, 2020</a> ] (en mm) et temps moyen pour traiter une image (en ms). . . . .	151
4.3	Résumé des différentes méthodes pour estimer la moelle . . . . .	159
5.1	Extraction des caractéristiques géométriques à partir des segments symétriques . . . . .	172
5.2	RMSD calculé pour les différentes méthodes. . . . .	180
5.3	Image d'entrée (les images de vaisseaux de fond d'œil proviennent de [ <a href="#">Hoover et al., 2000</a> ]), le filtrage selon [ <a href="#">Soille and Talbot, 2001</a> ] et le filtrage avec notre méthode uniquement des variations de longueurs (épaisseur fixée à 1). . . . .	181



# Introduction

Cette thèse s’inscrit dans un cadre applicatif de traitement d’images pour l’estimation de la qualité du bois. Il est nécessaire d’introduire à la fois le contexte informatique et biologique en même temps. C’est ainsi que nous allons exposer dans l’immédiat l’objectif de cette thèse :

**Extraire des informations sur la qualité du bois à partir de l’analyse d’images (Rouge Vert Bleu (RVB)), issues de capteurs low-cost (par exemple smartphones), de sections transversales de grumes ou billons de bois acquises en forêt ou en scierie.**

Dans la suite de cette introduction, nous allons décrire les motivations quant aux choix : du traitement d’images, de la qualité du bois, de l’image de la coupe transversale et des capteurs low-cost utilisés pour photographier ces coupes. Mais avant d’aborder toutes ces motivations, la naissance du projet doit faire l’objet d’une présentation.

## Collaboration ADAGIo et Silva

Les prémices de ces quatre années de thèse ont eu lieu en 2018, dans le cadre de mon stage de deuxième année du Master « Interaction, Perception, Apprentissage, Connaissance » (devenu le Master « Apprentissage, Vision, Robotique »). J’ai intégré l’équipe Algorithmique Discrète et ses Applications à la Géométrie et à l’Imagerie (ADAGIo) du **Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)**.

Le domaine privilégié de l’équipe ADAGIo, la géométrie discrète, s’attache à l’étude des structures mathématiques discrètes. C’est un outil pertinent dans l’estimation précise de caractéristiques géométriques (longueurs, tangentes, courbure, ...) des objets discrets par exemple les objets présents dans les images numériques.

L’équipe ADAGIo venait d’être sollicitée, par une chercheuse, Fleur Longuetaud, et un chercheur, Frédéric Mothe, de l’UMR Silva (équipe Woodstock) de l’**Institut national de recherche pour l’agriculture, l’alimentation et l’environnement (INRAE)**, pour un projet commun : TreeTrace. Fleur deviendra ensuite co-directrice de la thèse et Frédéric deviendra interlocuteur et collaborateur privilégié et impliqué pendant toute la durée de la thèse.

Le projet TreeTrace est un projet binational franco-autrichien, financé par l’**Agence Nationale de la Recherche (ANR)** pour la partie française et le **Fonds zur Förderung der wissenschaftlichen Forschung (FWF)** (l’équivalent autrichien de l’ANR) en Autriche. Cela nous amène à collaborer avec le département informatique de l’Université de Salzbourg. Ce projet a pour but deux applications possibles des technologies d’imagerie, notamment des

données issues de capteurs **RVB**. La première application est la question de la traçabilité des billons d'arbres depuis le site d'exploitation forestière jusqu'à la scierie, voire jusqu'au client final, en utilisant des techniques de reconnaissance liées à la biométrie et basées uniquement sur le traitement d'image de coupes transversales. Cette approche suppose bien sûr la disponibilité supplémentaire de capteurs d'imagerie dans la forêt. La partie autrichienne de ce projet s'occupe de cette application. La deuxième est l'évaluation de la qualité du bois à partir d'images **RVB** de coupes transversales, applicable en forêt et/ou en scierie. Cette thèse s'inscrit dans ce projet commun et notamment sur ce deuxième objectif.

Les collaborations entre ADAGIo et Silva ne datent pas de ce projet. En 2014, Adrien Krähenbühl a effectué sa thèse sur la détection de nœuds dans des images tomodesitométriques de **grumes** de bois [Krähenbühl et al., 2014]. En 2018, l'équipe ADAGIo intervient comme interlocuteur de la thèse de Van Tho Nguyen [Nguyen et al., 2018, Nguyen et al., 2020] dont l'objectif est de détecter les défauts en surface à partir de données **LIDAR** (voir la **figure 1**).

Du côté de l'équipe Woodstock, le projet est un cadre applicatif idéal pour savoir si la qualité du bois, déjà très étudiée sur des images tomodesitométrique ou **LIDAR**, est possible avec des capteurs low-cost comme ceux d'**appareils photographiques numériques (APN)**, provenant principalement de smartphones ou de petites caméras industrielles et à partir de photos de sections à l'extrémité uniquement. Du côté d'ADAGIo, le projet est un cadre applicatif idéal pour expérimenter les outils géométriques, de l'équipe et du domaine, avec la perspective d'en développer de nouveaux.

En particulier, les objets à étudier dans une image de coupe transversale de **grume** générée par un **APN** s'apparentent à des objets discrets. Nous le verrons par la suite, mais l'étude de caractéristiques géométriques de ces objets (notamment les **cernes**) permettrait d'obtenir des informations sur la qualité de la **grume**. Ces images, en plus des outils de traitement d'images, se prêtent tout à fait à l'expérimentation et à la prospection de nouveaux outils et de nouvelles approches discrètes.

## Le traitement d'images

Le besoin de traiter les images est très vite devenu nécessaire, dès les années cinquante en physique des particules (où les scientifiques observent ces particules élémentaires à l'aide d'accélérateur de particules) notamment pour détecter des trajectoires notamment complexes.

Depuis, **appareils photographiques numériques (APN)**, scanners ou encore images satellites météo se sont développés et les images ont envahi notre vie quotidienne. Leur traitement est désormais devenu usuel dans beaucoup de domaines. Il touche aussi bien la vie courante à travers la reconnaissance faciale ou celle de l'empreinte digitale pour déverrouiller son smartphone [Chen et al., 2014], à travers une recherche à l'aide d'une image (et non d'un texte) avec des outils comme *Tineye*<sup>1</sup> ou Google Image) [Mamrosh and Moore, 2015], ou encore à travers la détection de modification d'image à l'aide d'ou-

---

1. <https://tineye.com/>

---

tils comme Foto Forensics ou Sherloq<sup>2</sup>. Le traitement d'images est devenu essentiel dans beaucoup de domaines de recherches comme la recherche médicale (analyse de radiographie, d'échographie, ...) avec des techniques de restauration d'images, de segmentation. Nous pouvons aussi citer la recherche archéologique [Even and Ngo, 2020], pour laquelle un objectif du traitement d'images obtenues par **Laser Imaging Detection And Ranging (LIDAR)** pourrait être de détecter si un site est propice ou non à une fouille archéologique. Le traitement d'image est également utilisé dans le monde industriel avec comme exemple le développement de la voiture autonome qui nécessite de reconnaître le paysage pour déterminer les actions à faire, ou bien encore en architecture pour la reconstruction 3D de bâtiments [Dimitrov and Golparvar-Fard, 2015]. Le contrôle de la qualité est aussi un enjeu important dans le monde industriel. Sur ce point, le traitement d'images a des solutions à apporter, par exemple dans l'estimation de la qualité de la nourriture [Basantia et al., 2018, Zhu et al., 2021b].

Il semble pertinent de se poser la question pour la qualité du bois. Le traitement d'images est vraisemblablement un outil intéressant, à la vue de ses multiples applications, pour l'estimation de la qualité du bois.

## Système d'acquisition d'images

### Scanner à rayons X

Avant de pouvoir traiter des images, il faut d'abord les acquérir. Une technique d'imagerie très utilisée est la tomographie par absorption de rayons X. Elle est non destructive et permet la reconstruction d'images « en coupe » d'un objet à trois dimensions.

Comprendre l'organisation interne d'un arbre et les principes qui régissent sa croissance nécessite d'obtenir une vision globale et précise du tronc. C'est pourquoi ce système d'acquisition d'images a été adopté très tôt par la recherche sur le bois. En effet, les scanners à rayons X sont tout à fait adaptés à l'étude de grumes : connaître les variations de densité est une information suffisamment précise pour décrire l'organisation interne de la grume, mais également pour identifier des éléments clés pour l'estimation de la qualité du bois [Taylor et al., 1984, Funt and Bryant, 1987, Som et al., 1993, Andreu and Rinnhofer, 2003, Longuetaud et al., 2004, Entacher et al., 2007, Freyburger et al., 2009, Wei et al., 2011, Krähenbühl et al., 2012, Berglund et al., 2013, Johansson et al., 2013, Gazo et al., 2020] (voir la [figure 1](#)).

D'autre part, l'étude non destructive du bois est également un potentiel intéressant pour les scieries [Hodges et al., 1990, Rais et al., 2017]. Les scanners à rayons X leur donnent la possibilité d'analyser chaque tronc avant de le découper en planches. Cette analyse doit leur permettre de définir l'orientation optimale du sciage, celle qui maximise la qualité finale des planches. Stängle et al. [Stängle et al., 2015] ont montré que si la rotation optimale était appliquée à chaque tronc, le rendement total pouvait être amélioré de 24% par rapport au rendement moyen d'un ensemble de douze orientations.

Cependant, cette technologie s'avère très coûteuse d'un point de vue financier. Le prix varie fortement selon la taille du scanner, pouvant aller de quelques dizaines de milliers

---

2. <https://inteltechniques.com/blog/2021/08/14/sherloq-an-open-source-image-forensic-toolset/>

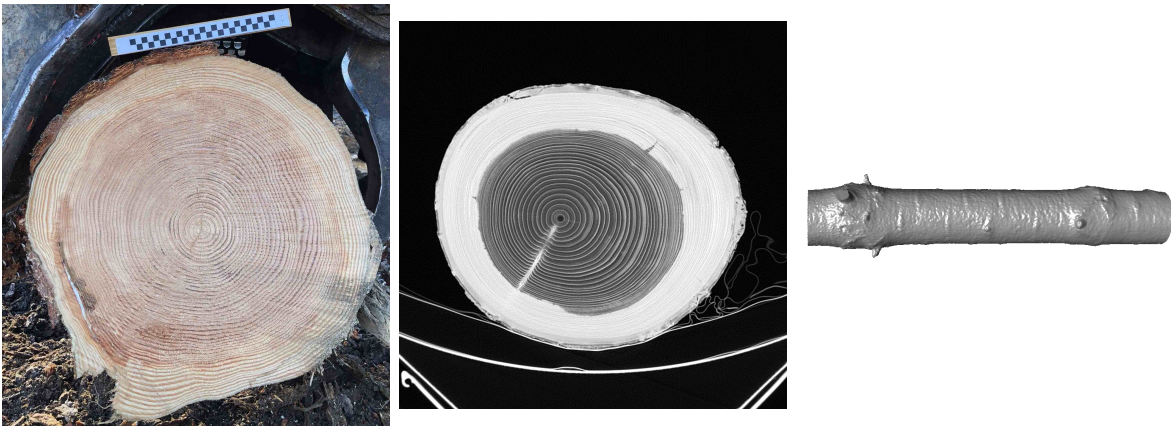


FIGURE 1 – Exemple de données issues de différents type de capteurs, de gauche à droite : smartphone, scanner à rayons X et LIDAR après maillage (provenant de [Nguyen et al., 2018]).

euros à presque un million euros. Elle est également gourmande en terme de ressources. De fait, cet appareil nécessite un local pour l’entreposer, de personnes qualifiées pour l’utiliser, etc. Enfin, le volume de données généré nécessite du temps pour les analyser et ce, malgré les avancées en analyse d’images. Il existe en France deux scieries disposant de scanner à rayons X industriels. Ils sont conçus pour traiter les données en temps réel et ne stockent pas les données, ce qui enlève la question de leur stockage.

## Capteurs low-cost

Il est possible d’acquérir d’autres types d’images de façon moins onéreuse et moins contraignante. Il s’agit d’images Rouge Vert Bleu (RVB) en utilisant des capteurs issus d’appareil photographique numérique (APN) (voir la figure 1).

Bien entendu, les images photographiées à partir de ce type de capteur ne sont pas de même nature que celles de scanner à rayons X. Ces images ne peuvent pas montrer l’organisation interne d’un billon. Cependant, il existe des caractéristiques externes qui permettent d’identifier la qualité du bois (voir le chapitre 2). Ces caractéristiques sont très visibles sur une coupe transversale d’une grume de bois. C’est la raison pour laquelle les images qui sont analysées dans ce manuscrit pour estimer la qualité du bois sont celle d’images de coupes transversales de grume de bois.

En 2020, le nombre d’APN vendus est un peu moins de 9 millions [CIPA, 2020]. Le nombre de smartphones vendus en 2020 a atteint les 1,29 milliards [Mongardini and Radzikowski, 2020]. Cet quantité d’appareils produits a eu un effet considérable sur leur prix d’achat. Les prix des smartphones, en France, vont de 50 euros pour un smartphone d’entrée de gamme à 2000 euros pour un smartphone haut de gamme. Autrement dit, l’achat de ce type d’appareil est 100 fois moindre que l’achat d’un scanner à rayons X.

Ajoutons également qu’il n’y a pas la nécessité d’avoir un local pour entreposer un smartphone. Cela permet d’être utilisable plus facilement. D’autant plus que les smartphones sont des appareils embarqués, il est possible de photographier directement en forêt le billon. Par ailleurs, cela réduit la logistique de déplacement des billons.

---

Soulignons aussi qu'il n'y a pas le besoin de personne qualifiée pour l'utiliser. N'importe quelle personne sachant se servir d'un APN est capable de photographier l'extrémité d'un billon de bois. Ce point peut servir à acquérir davantage des données, ce qui est, pour l'analyse de grandes quantités de données à travers par exemple l'apprentissage supervisé, devient de plus en plus nécessaires [Jain and Chandrasekaran, 1982].

En somme, le temps d'acquisition d'une image par ce type de capteur est instantanée. Cet instantanéité facilite l'acquisition d'une nouvelle image s'il s'avère que l'acquisition précédente est jugée de mauvaise qualité. D'ailleurs, ce temps d'acquisition simplifie l'installation d'APN dans la chaîne de montage de la scierie.

Pour finir, le volume d'une image issu d'un APN est moindre que le volume de données générées par un scanner à rayons X. Cela diminue potentiellement le temps de traitement.

Tous les avantages présentés font de l'APN un objet d'attention pour le contrôle de la qualité du bois, aussi bien pour le monde de la recherche que celui de l'industrie.

## Les enjeux pour la filière forêt-bois

Des enjeux écologiques, industriels, mais aussi individuels, par exemple à travers l'esthétique et la traçabilité, doivent être pris en compte dans l'intérêt, toujours plus croissant, pour la qualité du bois et de son estimation.

En France hexagonale, la forêt recouvre environ 16,9 millions d'hectares sur les 54,9 millions d'hectares du territoire [IGN, 2020]. Ainsi, la forêt occupe près de 31% du territoire européen de la France. En terme de surface, la forêt française est en deuxième position

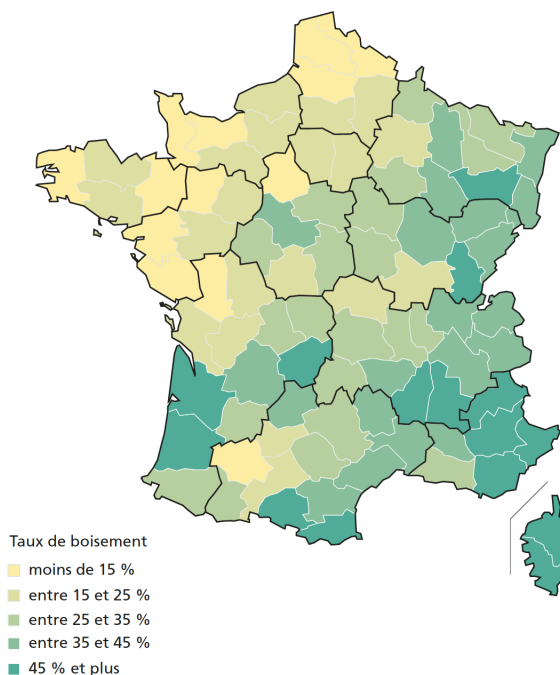


FIGURE 2 – Taux de boisement des départements (France hexagonale), d'après IGN (2020).



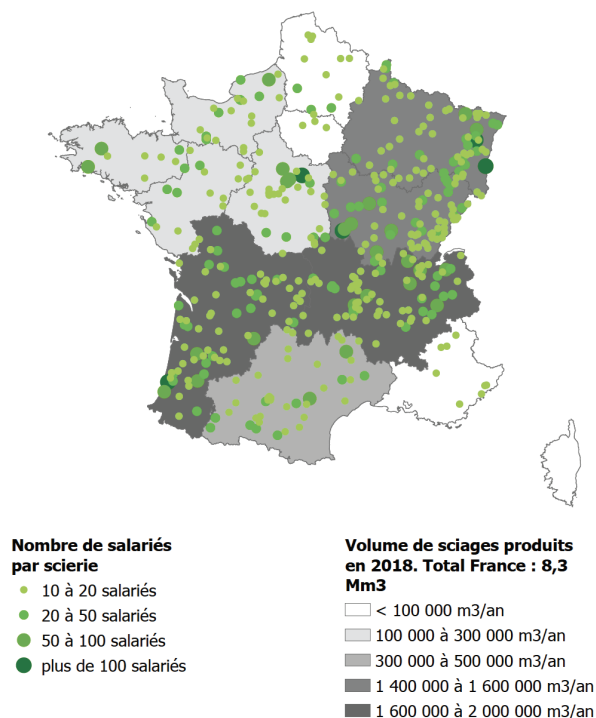


FIGURE 3 – Localisation des scieries et production régionale de sciages en France hexagonale en 2018 [FCBA, 2020].

juste après les terres agricoles qui occupent un peu plus de la moitié du territoire.

La surface forestière en France est en augmentation depuis plus un siècle. Depuis les années 1985, l'accroissement est toujours d'actualité. En l'espace de 30 ans, le stock de bois sur pied a connu une augmentation d'environ 50%, atteignant les 2,9 milliards de mètres cube<sup>3</sup> [IGN, 2020, FCBA, 2020].

Pourtant, les 31% de surface occupés par la forêt ne suffisent pas à décrire la réalité géographique. En effet, il existe de fortes disparités entre les départements. Cette répartition est montrée sur la figure 2. Sept départements, tous situés dans le nord et l'ouest, ont un taux de boisement inférieur à 10%. Et cinq départements ont un taux supérieur à 60% [IGN, 2020].

L'importante surface forestière fait de la filière forêt-bois française un acteur majeur du territoire. En 2017, plus de 300 000 emplois étaient directement liés à la production, la transformation et la mise en œuvre des produits bois. La valeur ajoutée par ce secteur est estimée à plus de 15 milliards d'euros en 2017 [FCBA, 2020].

D'un autre côté, le changement climatique nous incite à modifier notre façon de vivre afin de limiter nos répercussions sur l'environnement. Le sixième rapport du groupe de

3. L'estimation du volume des arbres est faite à partir des mesures réalisées sur le terrain par l'Institut national de l'information géographique et forestière (IGN). Seuls les arbres dits « recensables » sont pris en compte dans l'estimation. Un arbre est recensable si sa circonférence à 1,30 m de hauteur est supérieure ou égale à 23,5 cm (soit 7,5 cm de diamètre).

---

travail n°1 du GIEC affirme une nouvelle fois la relation très étroite entre les émissions anthropiques de gaz à effet de serre (GES) et le réchauffement planétaire qu'elles engendrent [Masson-Delmotte et al., 2021]. La France s'est engagée, en signant l'Accord de Paris en 2015, à limiter ses émissions de six gaz à effet de serre (GES) d'origine anthropique dont le principal est le dioxyde de carbone (CO<sub>2</sub>). De plus, conformément aux lignes directrices du GIEC, les stocks de bois dans la construction peuvent être comptabilisés comme puits de carbone. Ainsi, cette filière a un rôle important à jouer face à ce problème.

La filière bois se sépare en trois grandes catégories : le bois d'œuvre (construction et ameublement), le bois d'industrie (palette, emballage et bois de trituration) et le bois énergie (chauffage).

Sur les 84,6 millions de mètres cubes de production biologique annuelle française, 38,9 millions de mètres cubes étaient prélevés en 2018, soit 46% de la production. Parmi ces 38,9 millions : environ 51% sont du bois d'œuvre, 26% du bois d'industries et 22% du bois énergie. La répartition des scieries en France est également inégale [FCBA, 2020] (voir figure 3).

À travers les chiffres, nous pouvons voir l'importance de la filière-bois, notamment dans la construction. Les potentielles solutions afin d'estimer la qualité du bois comportent un caractère financier important pour l'industrie du bois, notamment son automatisation et sa rapidité d'exécution. Cette estimation est aussi importante pour les consommateurs qui demandent de plus en plus de contrôle de la qualité, ainsi que de la traçabilité (pour rappel il s'agit du deuxième objectif du projet TreeTrace).

## Objectifs de la thèse

Après avoir expliqué pourquoi nous nous intéressons à des images, pourquoi nous nous focalisons sur la coupe transversale d'une grume de bois, pourquoi nous avons utilisé des capteurs low-cost pour prendre ces images et pourquoi nous nous concentrons sur la qualité du bois, nous indiquons maintenant les objectifs de la thèse.

L'analyse d'images RVB de grumes va permettre d'améliorer la valorisation de la ressource bois. De fait, les scieries ne peuvent pas forcément se permettre d'inclure un scanner CT dans leur chaîne de montage, pour diverses raisons, notamment le prix. Elles pourraient plus aisément investir dans un capteur RVB pour estimer la qualité de la grume et in fine prendre en compte cette information dans les plans de découpe.

Cela étant dit, un capteur RVB placé dans une chaîne de découpe des planches doit être en mesure de photographier la grume et d'analyser à une vitesse inférieure au temps de découpe d'une grume (de l'ordre d'une minute). C'est un enjeu important et les solutions proposées seront également analysées et évaluées sous l'angle de cette contrainte de temps.

Par ailleurs, l'analyse pourrait également servir d'autres domaines de la recherche. Ces derniers n'ont pas eux aussi forcément les moyens d'obtenir des images CT, malgré qu'elles aient été largement étudiées et continue de l'être dans ce cadre [Andreu and Rinnhofner, 2003, Longuetaud et al., 2004, Borianne et al., 2011, Wei et al., 2011, Beaulieu and Dutilleul, 2019]. Par exemple, l'estimation des cernes peut-être utilisée dans le domaine de la dendrochronologie qui est la science traitant de l'étude et de la datation des cernes de

croissance des arbres. Cette dernière permet par exemple d'estimer des fluctuations dans le climat passé. Une autre application est l'estimation de la largeur d'**aubier**. En effet, la largeur d'**aubier** est très corrélée à la santé de l'arbre [Becker et al., 1987].

L'objectif de cette thèse est de répondre aux questions suivantes :

- quels sont les caractéristiques ou propriétés du bois que nous pouvons analyser sur ce type d'images ?
- comment développer des algorithmes robustes et rapides analysant des images issues de capteurs **RVB** afin d'estimer ces caractéristiques ?
- pouvons-nous déduire une information sur la qualité de la grume à partir des informations analysées ?

## Organisation du manuscrit

Dans un premier temps nous présentons dans le **chapitre 1** des outils issus du traitement d'images. Une brève présentation de ce qu'est une image **RVB** et de ses représentations dans des espaces de couleurs est introduite. Les outils présentés abordent principalement la segmentation d'une image.

Nous abordons dans le **chapitre 2** leur application dans l'estimation de caractéristiques biologiques de grumes de bois sur des images **RVB**. Cet état de l'art permet de mettre en avant le peu de méthodes qui ont été faites sur le même type d'images pour estimer les caractéristiques. Nous constatons que bien souvent les méthodes proposées sont ad hoc. Ce chapitre est l'occasion d'introduire ces dites caractéristiques qui se fait à travers une courte synthèse de l'anatomie du bois.

Le **chapitre 3** présente les apports qui ont été faits pour estimer les caractéristiques. Nous détaillons les images acquises et utilisées durant le projet Treetrace. Nous abordons les solutions proposées pour la segmentation de la grume, la détection de la zone de duramen. Celle-ci permet d'en déduire la zone d'**aubier**. Enfin, nous traitons la question de la détection des cernes de croissance.

Par la suite, le **chapitre 4** s'intéresse à une caractéristique particulière : l'estimation de la position de la moelle. Nous y synthétisons l'état de l'art sur le sujet (traitant le même type d'images que celles de la thèse). Ce chapitre présente également notre apport concernant cette caractéristique.

Puis, le **chapitre 5** propose des nouveaux outils de géométrie discrète : un filtrage directionnel. Cet outil fait appel à des notions de morphologie mathématique. Il permet d'estimer des caractéristiques géométriques locales en chaque point d'image comme la largeur, l'orientation, ...

Enfin le **chapitre 6** dressera le bilan de cette thèse en résumant l'ensemble des travaux effectués.

# Techniques de segmentation

## Sommaire

---

<b>1.1</b>	<b>Introduction</b>	<b>24</b>
1.1.1	Représentation de la couleur et de l'intensité	24
<b>1.2</b>	<b>Objectifs, propriétés et évaluation de la segmentation</b>	<b>27</b>
1.2.1	Les objectifs d'une segmentation	29
1.2.2	Évaluation de la segmentation	29
1.2.3	Matrice de confusion	30
<b>1.3</b>	<b>Seuillage et histogramme</b>	<b>31</b>
1.3.1	Seuillage et histogramme	31
1.3.2	Méthode d'Otsu	32
1.3.3	Seuillage par hystérésis	33
<b>1.4</b>	<b>Approche par gradient</b>	<b>34</b>
1.4.1	Gradient d'une image	35
1.4.2	Convolution et dérivée	35
1.4.3	Canny	36
<b>1.5</b>	<b>Accroissement de régions</b>	<b>37</b>
1.5.1	Décomposition en régions	38
1.5.2	Fusion des régions	39
<b>1.6</b>	<b>Contours actifs (ou Snakes)</b>	<b>40</b>
1.6.1	Énergie potentielle	41
1.6.2	Énergie interne	41
1.6.3	Énergie externe	42
<b>1.7</b>	<b>Apprentissage automatique - Réseaux de neurones</b>	<b>42</b>
1.7.1	Les principes de base	43
1.7.2	Fonctions d'activation	44
1.7.3	Apprentissage d'un neurone	46
1.7.4	Régularisation d'un réseau de neurones	47
1.7.5	Réseaux de neurones convolutifs	49
1.7.6	Exemple de réseaux de neurones	55
<b>1.8</b>	<b>Résumé</b>	<b>60</b>

---

## 1.1 Introduction

La segmentation est un procédé qui vise à décomposer les pixels d'une image en un ensemble de **classes** qui satisfont certains critères. Les méthodes de segmentation peuvent se classer en deux grandes approches.

La première est l'approche par classes, où l'idée est d'identifier l'ensemble des pixels qui forme chacune des classes souhaitées. La deuxième approche est celle par contours (ou par frontières) où l'objectif n'est pas d'identifier tous les pixels appartenant à une classe mais de délimiter chaque classe, c'est-à-dire d'identifier les pixels aux frontières des classes.

Nous ne pouvons pas faire un état de l'art exhaustif des recherches dans le domaine de la segmentation tant celui-ci est vaste. Nous nous limiterons à celles considérant une image comme un ensemble de pixels. Par ailleurs, nous centrons notre intérêt uniquement sur les méthodes de segmentation d'images 2D en niveaux de gris et/ou en couleur RGB qui constituent le support principal du reste du manuscrit. Enfin, nous présenterons uniquement les méthodes de segmentation qui ont été utilisées dans une application au bois et qui s'inscrivent dans le cadre de la thèse.

Dans cette partie d'état de l'art, avant d'aborder les techniques de segmentation, nous allons rappeler les espaces de couleurs. nous allons dans un premier temps rappeler les objectifs de la segmentation et ses propriétés (section 1.2). Ensuite, nous aborderons les approches classiques de segmentation par seuillage et histogramme (section 1.3). Nous verrons la notion de dérivée d'une image (section 1.4) et quelques techniques de segmentation autour de cette notion. Enfin nous détaillerons quelques méthodes plus avancées pour la segmentation tel que l'accroissement de régions (section 1.5), les contours actifs (section 1.6) ainsi que les réseaux de neurones convolutifs (section 1.7). Toutes les méthodes présentées dans ce chapitre seront utilisées dans les chapitres suivants.

### 1.1.1 Représentation de la couleur et de l'intensité

Nous présentons quelques espaces couleurs usuels pour représenter une image. Dans une image, chaque pixel est associé à une intensité, ou niveau de gris, qui est proportionnelle à la quantité de lumière capturée par le capteur associé au pixel. Dans le cas d'une représentation sur 8 bits, les niveaux de gris vont d'une valeur de 0, correspondant au noir, jusqu'à 255, correspondant au blanc.

Les capteurs utilisés pour acquérir une image **RVB** prennent uniquement des longueurs d'ondes spécifiques. Cette technique, appelée filtres de Bayer [Bayer, 1976], est très utilisée. Les longueurs d'ondes de couleur verte, rouge et bleue sont acquises par des composants du capteur et sont combinées pour former une image en couleur.

Chaque couleur est échantillonnée à une fréquence spatiale basse, impliquant une résolution basse. Les informations potentiellement manquantes sont interpolées à l'aide des informations voisines, ce qui lisse chacune des composantes de couleur. Parfois, l'interpolation peut créer des artefacts, notamment lorsque qu'il y a un fort changement de couleur. Une fois l'acquisition faite, chaque pixel a une intensité pour les trois couleurs :

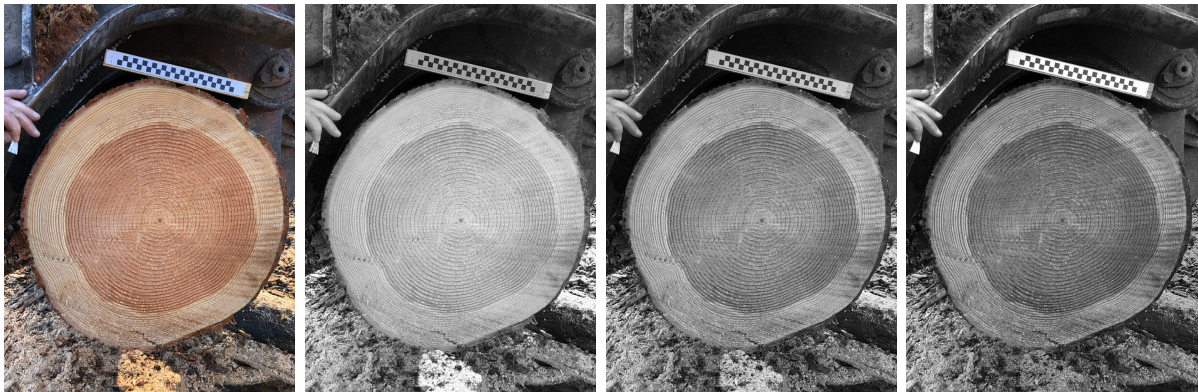


FIGURE 1.1 – Exemple de l’espace RVB. De gauche à droite : l’image d’origine en RVB, la composante rouge, la composante verte, la composante bleue.

rouge, verte et bleue. La majorité des APN utilisent ce principe ou des variantes<sup>4 5</sup>. Cette méthode d’acquisition fait que les images sont représentés dans l’espace des couleurs RVB.

La figure 1.1 montre la décomposition des trois composantes couleurs (rouge, verte et bleue) pour une image capturée avec les filtres de Bayer.

Il existe d’autres espaces de représentation des couleurs. L’espace Hue Saturate Value (HSV) (ou Teinte Saturation Brillance en français) propose de décomposer les couleurs en trois composantes. Ce système est le mieux adapté à la perception humaine des couleurs. La teinte correspond au nom des couleurs : rouge, rose, bleu, jaune, etc. La saturation est l’intensité de la couleur : plus la saturation est faible plus la couleur sera vue comme grise. Enfin plus la brillance est élevée plus la couleur sera claire, et inversement, une valeur faible de brillance donnera une couleur plus sombre.

La conversion RVB vers HSV est faite de la manière suivante. Soit  $R$ ,  $G$  et  $B$  les niveaux de gris, normalisés entre 0 et 1, associés aux composantes rouge, verte et bleue d’un pixel. Soient  $C_{max} = \max(R, G, B)$ ,  $C_{min} = \min(R, G, B)$  et  $\Delta = C_{max} - C_{min}$ , les valeurs des composantes  $H$  (valeur comprise entre 0 et 360),  $S$  (valeur comprise entre 0 et 1) et  $V$  (valeur comprise 0 et 255) et sont définies par :

$$H = \begin{cases} 0^\circ, & \text{si } C_{max} = C_{min} \\ (60^\circ \times \frac{G-B}{\Delta} + 360^\circ) \bmod 360^\circ, & \text{si } C_{max} = R \\ 60^\circ \times \frac{B-R}{\Delta} + 120^\circ, & \text{si } C_{max} = G \\ 60^\circ \times \frac{R-G}{\Delta} + 240^\circ, & \text{si } C_{max} = B \end{cases}$$

$$S = \begin{cases} 0, & \text{si } C_{max} = 0 \\ 1 - \frac{C_{min}}{C_{max}}, & \text{sinon} \end{cases}$$

$$V = C_{max}$$

4. <https://www.sony.com/en/SonyInfo/News/Press/201807/18-060E/>

5. <https://news.samsung.com/global/samsungs-108mp-isocell-bright-hm1-delivers-brighter-ultra-high-res-images-with-industry-first-nonacell-technology>

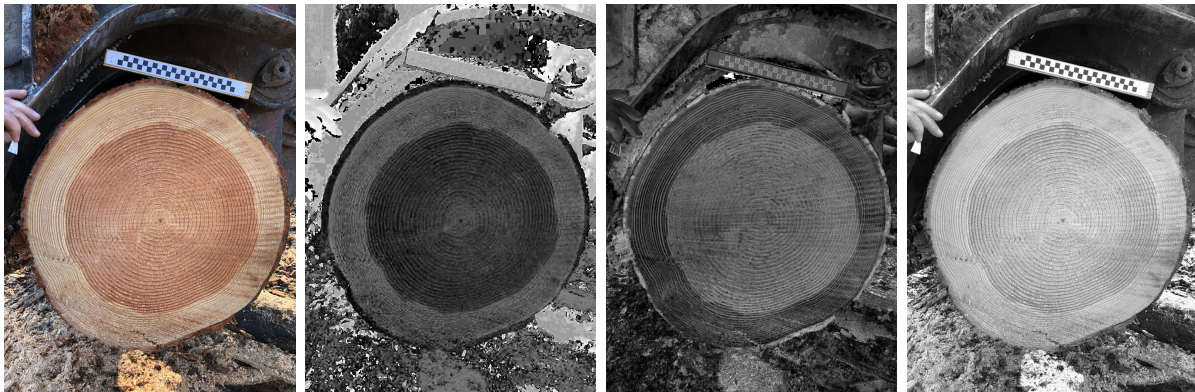


FIGURE 1.2 – Exemple de l'espace HSV. De gauche à droite : l'image d'origine en RVB, la composante H (dont l'histogramme a été égalisé afin de mieux visualiser cette composante), la composante S et la composante V.

La [figure 1.2](#) montre les trois composantes dans l'espace HSV. L'image est la même que celle présentée pour les composantes RVB.

Une autre représentation de couleur est l'espace [CIE LAB color space \(CieLAB\)](#). Cet espace est dérivé de l'espace XYZ proposé en 1931<sup>6</sup> [[Jones, 1943](#)]. La composante  $L$ , ou  $L^*$ , est très proche de la luminosité que perçoit l'œil humain. Lorsque  $L = 0$  la couleur perçue est le noir et lorsque que  $L = 100$  la couleur perçue est le blanc. La composante  $a^*$  représente la position entre le rouge et le vert (une valeur négative sera une couleur verte, et une valeur positive sera une couleur rouge). Enfin, la composante  $b^*$  représente la position entre le jaune et le bleu (une valeur négative sera une couleur bleue, et une valeur positive sera une couleur jaune).

La première étape de conversion de l'espace RVB à l'espace [CieLAB](#) est de passer de la représentation RVB à la représentation XYZ. Pour cela, il suffit d'effectuer la transformation linéaire suivante :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0,412453 & 0,212671 & 0,019334 \\ 0,357580 & 0,715160 & 0,119193 \\ 0,180423 & 0,072169 & 0,950227 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

La matrice de changement n'est pas forcément toujours la même, car elle dépend de l'emplacement de la couleur blanche dans l'espace XYZ. La matrice ci-dessus est celle actuellement utilisée dans la plupart des bibliothèques des traitements d'images.

À partir des coordonnées XYZ, il est possible de définir les composantes  $L$ ,  $a^*$  et  $b^*$  de l'espace [CieLAB](#). Cela se fait par le calcul suivant :

$$\begin{pmatrix} L^* \\ a^* \\ b^* \end{pmatrix} = \begin{pmatrix} 116f\left(\frac{Y}{Y_n}\right) - 16 \\ 500 \left[ f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \\ 200 \left[ f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \end{pmatrix}, \quad \text{où } f(t) = \begin{cases} t^{1/3} & \text{si } t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3} \left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{sinon} \end{cases}$$

6. La Commission Internationale de l'Illumination (CIE) propose cette représentation

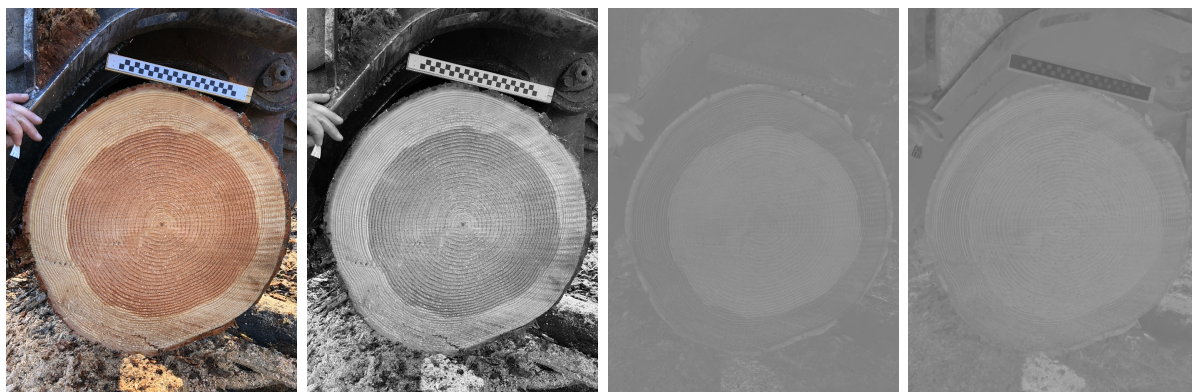


FIGURE 1.3 – Exemple de l’espace **CieLAB**. De gauche à droite : l’image d’origine en RVB, la composante  $L^*$ , la composante  $a^*$ , la composante  $b^*$

avec  $\begin{pmatrix} X_n \\ Y_n \\ Z_n \end{pmatrix} = \begin{pmatrix} 0,950456 \\ 1,0000 \\ 1,088754 \end{pmatrix}$  correspondant aux valeurs du blanc de référence également utilisé pour la conversion dans l’espace XYZ. La **figure 1.3** montre les trois composantes dans l’espace CieLAB.

Nous pouvons voir l’influence des espaces sur les éléments de la grume. Par exemple, la composante H de l’espace HSV met en avant le **duramen** qui a, dans le cas du douglas, une teinte rouge/rose. C’est aussi le cas pour la composante S de ce même espace. Soulignons que l’écorce, qui délimite la grume, est aussi mis en évidence par cette représentation des couleurs. Concernant l’espace CieLAB, nous observons aussi une différence entre l’aubier et le duramen. Tout comme la composante H, la composante  $a^*$  permet de mieux distinguer l’écorce, et donc de définir le contour de la grume.

Autrement dit, le changement d’espace de représentation pourrait être envisagé dans le cadre de la segmentation de la grume ou du duramen, qui sont deux éléments qui nous permettrons d’estimer la qualité de la grume (voir **chapitre 2**).

## 1.2 Objectifs, propriétés et évaluation de la segmentation

Nous venons de le voir, la segmentation consiste à regrouper un ensemble de pixels dans une même classe. Il convient d’abord de définir formellement une image, jusqu’à présent nous avons simplement défini l’image comme un ensemble de pixels. Une définition [Easton Jr, 2010] possible d’une image est la suivante :

**Définition 1.2.0.1 (Image 2D)** Une *image 2D*, notée  $I$ , est une fonction de deux variables spatiales  $(i, j)$ . C’est ainsi une application d’un ensemble  $E \subset \mathbb{Z}^2$  dans un ensemble  $F \subset \mathbb{R}^d$  :

$$I : \begin{array}{ccc} E & \rightarrow & F \\ (i, j) & \mapsto & I(i, j) \end{array}$$



Par exemple, une image 2D de dimension  $512 \times 512$  en niveaux de gris codés sur 256 valeurs dont l'origine est le pixel en bas à gauche aura comme ensemble  $E = \{0, \dots, 511\} \times \{0, \dots, 511\}$  et comme ensemble  $F = \{0, \dots, 255\}$  (i.e.  $d = 1$ ). La même image mais en RGB et avec des valeurs normalisées entre 0 et 1 aura comme ensemble  $F = [0, 1]^3$  (i.e.  $d = 3$ ).

À partir de cette définition, nous pouvons, de manière formelle, définir la segmentation comme suit :

**Définition 1.2.0.2 (Segmentation)** La *segmentation* d'une image  $I : E \rightarrow F$  est un partitionnement du domaine de définition  $E$  en un ensemble de sous-domaines  $(C_k)_{k \in \mathbb{N}}$ , appelés **classes**, tel que :

$$E = \bigcup_k C_k \quad \text{et} \quad \forall p, q, \quad p \neq q \implies C_p \cap C_q = \emptyset$$

Lorsqu'il n'y a que deux classes, la segmentation est appelée **binarisation**. La définition ci-dessus laisse sous-entendre que tout partitionnement d'une image est une segmentation. Bien que formellement cela soit le cas et que chaque méthode propose une segmentation différente, il est évident que partitionner de manière aléatoire une image ne constitue par une segmentation. En effet, le but de la segmentation est d'extraire un ensemble de classes d'une image afin de mieux les étudier individuellement, de leur donner un sens (une interprétation) ou pour simplement effectuer un traitement spécifique à chaque classe.

Par exemple dans la [figure 1.4](#), la deuxième segmentation (au centre) ne présente, a priori, aucune interprétation possible pour les classes définies, alors que la dernière (à droite) nous permet (du moins à notre cerveau) de donner un sens aux classes obtenues : l'arrière-plan (région en jaune) et la loutre (région bleutée).

Une fois la segmentation effectuée, il est possible de définir l'image segmentée obtenue  $\mathcal{L} : E \rightarrow \mathbb{N}$ , appelée parfois image labellisée, qui va associer à chaque élément de  $E$  un numéro  $i$  correspondant à la région à laquelle il appartient, c'est-à-dire :

$$\forall (i, j) \in E, \quad \mathcal{L}(i, j) = k \iff (i, j) \in C_k$$



FIGURE 1.4 – Image d'une loutre à droite. Au centre, une segmentation possible. À droite une binarisation possible.

### 1.2.1 Les objectifs d'une segmentation

La segmentation a plusieurs objectifs, notamment :

1. l'identification d'un ou plusieurs objets afin d'en faire une analyse spécifique ;
2. l'explication de l'intégralité de l'image.

Dans le premier cas, une analyse spécifique consiste à séparer l'objet du reste de l'image. L'objectif pouvant ensuite effectuer d'autres traitements pour, par exemple, calculer les dimensions de l'objet, etc. Cela peut aussi être l'attribution d'une **sémantique** (un sens) à l'objet (herbe et la loutre dans la [figure 1.4](#)).

Dans un cadre général, l'objet n'est pas simplement connexe (l'objet est d'un seul tenant et sans trous), ni unique, il peut par exemple s'agir de segmenter toutes les personnes. Quelque soit le cas, le reste de l'image qui ne correspond pas à la recherche est considéré comme peu utile et n'est pas à traiter.

C'est dans ce premier cas que la thèse s'inscrit. L'objectif principal visé est d'identifier plusieurs objets connus à l'avance, que ce soit la détection de la grume, du **duramen**, de la zone de bois juvénile (15 premiers **cernes**) ou bien dans l'analyse des **cernes**. L'objectif suivant est d'en faire une analyse plus spécifique pour récupérer des mesures pertinentes pour l'estimation de la qualité du bois.

### 1.2.2 Évaluation de la segmentation

Pour évaluer une segmentation, on peut utiliser des indicateurs tels que : la robustesse, la précision, le temps de calcul, etc.

La précision s'évalue par des mesures sur les classes, comme la surface, le périmètre, le centre de gravité, des distances entre classes, etc. L'objectif devient alors d'identifier des classes les plus proches possibles de la réalité physique des objets segmentés afin que les mesures soient le plus exactes possibles.

La robustesse se définit comme la capacité d'un algorithme à *résister* au bruit [[Vacavant, 2017](#)]. Le bruit est l'ensemble des éléments non contrôlables qui viennent perturber la qualité de l'image (l'ensoleillement, les nuages, mouvement de l'appareil de capture, le processus d'acquisition, etc.). Quant à la *résistance* au bruit, cela veut dire que les sorties de l'algorithme sont, théoriquement ou empiriquement, indépendantes par rapport au bruit. Cette notion de robustesse est parfois mélangée avec les termes d'efficacité, de persistance ou simplement de qualité (de la méthode). La robustesse est un objectif lorsque que le cadre applicatif ne peut pas garantir une acquisition similaire pour chaque image. Elle est aussi un objectif lorsque le cadre applicatif va différer par rapport à celui d'origine. L'idéal est d'avoir la robustesse et la précision, mais souvent l'une se fait au détriment de l'autre. Dans notre cas, le cadre d'applicatif industriel ne peut garantir suffisamment des conditions similaires. Bien que le capteur peut être fixé en scierie, celui ne l'est pas pour des photos en forêt ou sur le parc à grumes. Même en scierie, les conditions météorologiques, jouant sur la luminosité, peuvent parfois modifier les conditions. Nous chercherons donc à avoir des algorithmes robustes.

Un aspect important dans la segmentation est le temps de calcul, notamment dans un cadre d'applications industrielles, voire embarquées (traitement en direct après la prise de

la photo sur smartphone). C'est le cas de l'analyse des images de **grumes** traités dans la thèse. Si l'objectif de temps est très important, il peut justifier une plus grande indulgence sur les objectifs de précision et de robustesse. Bien entendu, s'il n'y a aucune contrainte de temps, il est possible de proposer des méthodes avec meilleur rapport précision/robustesse.

Il semble alors nécessaire de devoir évaluer chacun de ces aspects. En fonction des objectifs visés et de l'évaluation des critères, nous pourrions quantifier la qualité de la segmentation.

Pour évaluer une segmentation, la méthode consiste à avoir une base d'images pour lesquelles nous disposons de leur segmentation dite **vérité terrain**. Bien souvent, la segmentation de référence d'une image doit être faite par un être humain, ce qui induit une part de subjectivité à prendre en considération lors de l'évaluation. En effet, il est peu probable que pour une même image deux personnes segmentent (manuellement), au pixel près, de la même façon. Cela soulève la question de savoir quelle est la bonne segmentation ? Une segmentation qui serait différente des deux premières obtenues ne pourrait-elle pas être meilleure que celles-ci ?

### 1.2.3 Matrice de confusion

Un des indicateurs usuels pour estimer la qualité d'une segmentation est la matrice de confusion. Nous allons, ici, présenter cette matrice dans le cas d'une binarisation (c'est-à-dire 2 classes).

Puisqu'il s'agit d'un cas binaire, nous utiliserons les termes de positif et négatif pour désigner ces deux classes. Les vrais et faux positifs (TP/FP) font référence au nombre de positifs prédits qui sont corrects/incorrects. De même, les vrais et faux négatifs (TN/FN) font référence au nombre de négatifs prédits qui sont corrects/incorrects. La matrice de confusion est alors la synthèse de ces quatre valeurs :

		Valeur réelle	
		Positive	Négative
Valeur prédite	Positive	TP	FP
	Négative	FN	TN

Une fois la matrice définie, il est possible d'introduire des mesures de la qualité en utilisant les éléments de la matrice. Bien qu'il n'existe pas de moyen parfait de synthétiser la matrice de confusion en une seule valeur, le coefficient de corrélation de Matthews (MCC) [Matthews, 1975] est un bon indicateur. Ce coefficient est considéré comme l'une des meilleures mesures [Powers, 2020]. En effet, cet indicateur prend en compte l'ensemble des valeurs possibles et ce indépendamment de la répartition des classes (plus de vrai positifs que de vrai négatifs par exemple). Cela limite les biais de sur-représentation d'une classe sur l'autre. Le MCC s'exprime de la façon suivante :

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (1.1)$$

Ce coefficient est similaire au coefficient de corrélation de Pearson en statistiques. Son interprétation est donc le même. Autrement dit, si la valeur est à 1 alors il y a corrélation entre la segmentation et la vérité terrain, c'est-à-dire qu'elles sont égales. Un coefficient à 0 indique qu'il n'y a aucune corrélation entre ces deux éléments.

Un autre indice est l'indice de Jaccard, ou *Intersection over Union*, peut-être utilisé pour estimer la qualité d'une segmentation. Cet indice est donné par la formule :

$$Jaccard = \frac{TP}{TP + FN + FP} \quad (1.2)$$

## 1.3 Seuillage et histogramme

Nous venons de définir le cadre de la segmentation. Nous allons nous intéresser maintenant aux méthodes pour effectuer une segmentation. Nous allons d'abord définir la notion de seuillage et d'histogramme ([sous-section 1.3.1](#)), puis nous regarderons quelques méthodes qui reposent sur ces deux outils ([sous-section 1.3.2](#) et [sous-section 1.3.3](#)).

### 1.3.1 Seuillage et histogramme

Seuiller une image consiste à fixer une valeur seuil  $T \in F$  qui va dissocier l'ensemble des pixels de  $E$  en deux régions (binarisation) telles que :

$$\forall (i, j) \in E, \quad \mathcal{L}(i, j) = \begin{cases} 0, & \text{si } I(i, j) \leq T \\ 1, & \text{sinon} \end{cases}$$

Cette notion de seuillage peut facilement s'étendre pour plusieurs niveaux de seuils, on parle alors de multiseuillage. Cette simple définition binaire nous suffira pour présenter la suite des techniques utilisées dans ce manuscrit.

La manière la plus simple de fixer le seuil  $T$  est de procéder de manière empirique ou en se basant sur des *a priori* de l'image. Des méthodes ont été proposées pour déterminer automatiquement la valeur du seuil.

Le seuillage est efficace si une seule image est à traiter, car le seuil peut être adapté pour cette image sans considération d'autres images. Elle peut être efficace si l'objet recherché se différencie des autres par le fait qu'il soit le seul à avoir certains niveaux d'intensité. Un autre avantage du seuillage est sa rapidité en temps de calcul.

Néanmoins, cette pratique est limitée lorsqu'il y a une nécessité d'automatisation de la segmentation et/ou les images présentent une variabilité entre elles, impliquant que l'objet recherché n'est pas toujours de même intensité. Un autre point faible du seuillage est le côté tout ou rien.

Des méthodes plus locales et adaptatives ont été proposées [[Lu et al., 2010](#), [Singh et al., 2012](#)]. Une méthode est locale si elle ne prend en compte que les valeurs dans un voisin du pixel à seuiller. Elle est adaptative si son comportement change selon le contexte. Par exemple, la valeur du seuil est calculée sur la base de certaines statistiques telles que la dynamique ou la variance de l'image (ou du voisinage).

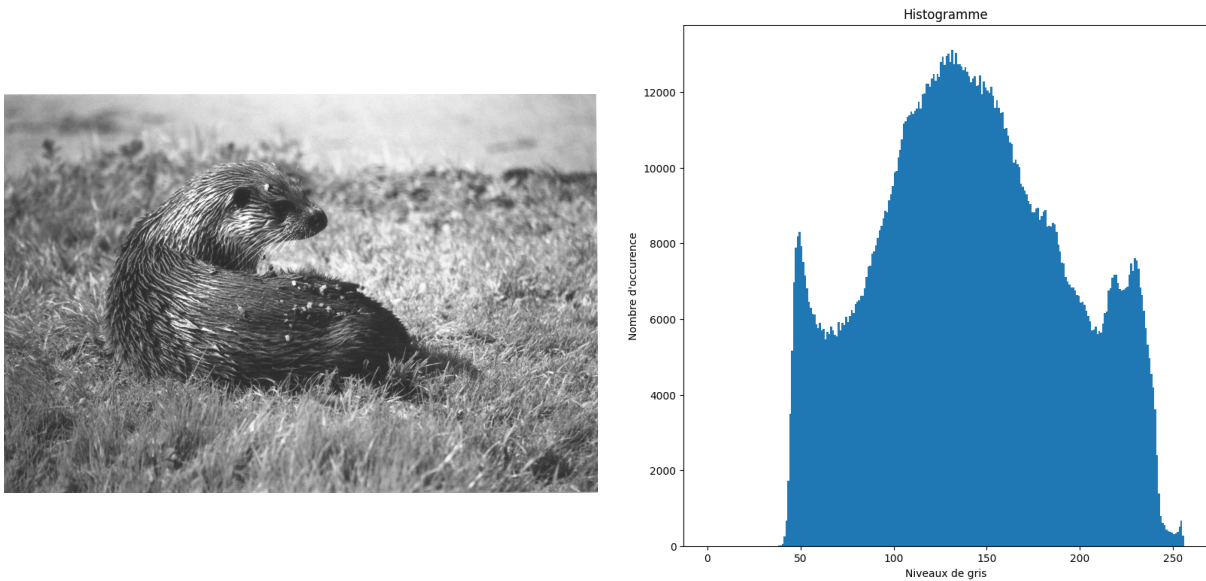


FIGURE 1.5 – L’histogramme (à droite) de l’image en niveaux de gris (à gauche).

Une catégorie de méthodes de segmentation très utilisée est celle employant l’histogramme d’intensité d’une image. Pour chaque niveau de gris, on détermine le nombre d’occurrences de ce dernier.

**Définition 1.3.0.1 (Histogramme)** *L’histogramme  $H$  d’une image  $I : E \leftarrow F$  en niveaux de gris et codée sur  $L$  niveaux de gris est définie par :*

$$\forall k \in [0, L - 1], H(k) = \sum_{(i,j) \in E} \chi_k(I(i, j)), \text{ avec } \chi_k(I(i, j)) = \begin{cases} 1 & \text{si } I(i, j) = k \\ 0 & \text{sinon} \end{cases}$$

Le choix du seuil  $T$  peut se faire visuellement à l’aide de l’histogramme. Puisque que l’histogramme représente la distribution des valeurs de niveaux de gris dans une image, utiliser un pic ou un creux dans l’histogramme peut être intéressant.

La figure 1.5 montre l’histogramme obtenu pour une image en niveaux de gris. Un pic se démarque dans l’histogramme, ce dernier peut être utilisé pour seuiller l’image. L’idéal est de trouver une méthode automatique pour déterminer où se situe ce pic.

### 1.3.2 Méthode d’Otsu

Fixer un seuil  $T$  de manière manuelle, même avec l’aide de l’histogramme, n’est pas facile. Otsu développe en 1979 [Otsu, 1979] une méthode automatique pour déterminer le seuil en fonction de l’histogramme. L’objectif est de minimiser la variance au sein de chaque classe. Pour la binarisation, cela revient à maximiser la variance entre ces deux classes. La méthode consiste à tester l’ensemble des valeurs de seuil possibles et de mesurer pour chacune d’entre elle la variance des deux classes définies par ce seuil. Otsu propose de garder la valeur pour laquelle la somme des deux variances est minimale. De manière

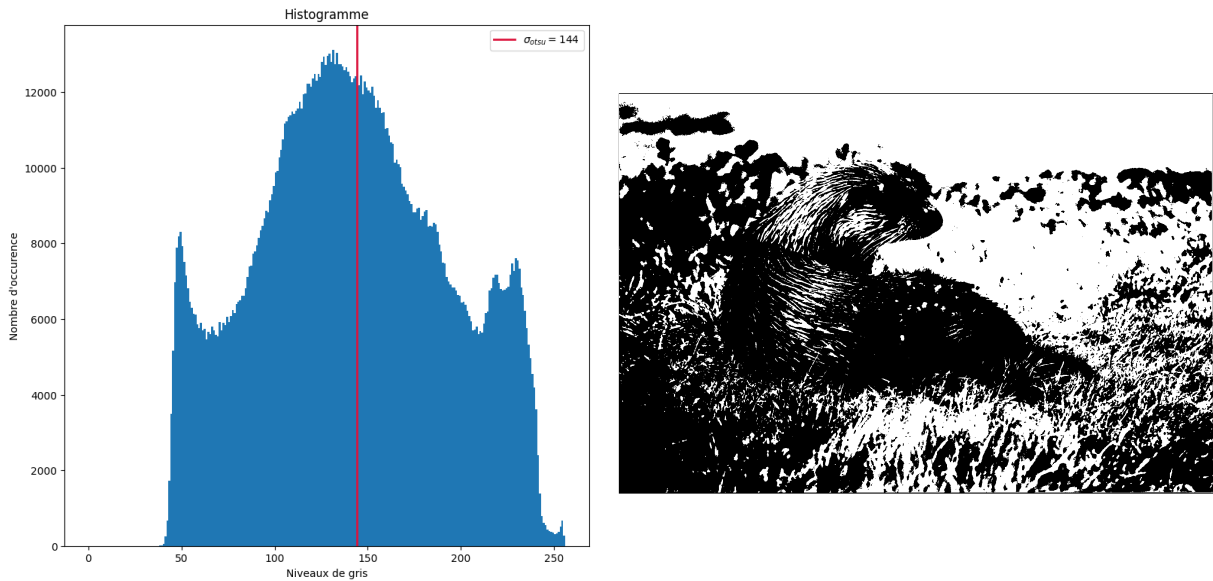


FIGURE 1.6 – Résultat de la méthode d'Otsu pour segmenter une image.

formelle le seuil est défini :

$$\sigma_{otsu} = \operatorname{argmin}_{\sigma} \left\{ (\mu_0 - \mu)^2 \sum_{k < \sigma} H(k) + (\mu_1 - \mu)^2 \sum_{k \geq \sigma} H(k) \right\} \quad (1.3)$$

avec,  $\mu$  la moyenne des valeurs de  $I$ ,  $\mu_0$  la moyenne des valeurs de  $I$  inférieur à  $\sigma$ , et  $\mu_1$  la moyenne des valeurs de  $I$  supérieur ou égale à  $\sigma$ .

La figure 1.6 montre le résultat de la méthode d'Otsu pour une image en niveaux de gris. Le seuil qui minimise la formule 1.3 est de 144, soit proche du pic que l'on observe dans l'histogramme.

Cette méthode est plus pratique qu'un choix manuel de seuil. Cette méthode est rapide en temps de calcul.

Cependant, la méthode est globale et ne prend donc pas en compte le voisinage d'un pixel. Autre inconvénient (voir figure 1.6), la méthode ne fonctionne que pour le cas binaire en supposant que l'histogramme soit bimodal, c'est-à-dire qu'il y ait exactement deux pics. Cela étant, des propositions d'amélioration ont été faites [Moghaddam and Cheriet, 2012, Yuan et al., 2015].

### 1.3.3 Seuillage par hystérésis

La technique de seuillage par hystérésis propose d'utiliser deux seuils. En effet, le principal défaut des méthodes de seuillage est le côté tout-ou-rien : le seuil est une barrière stricte pour définir si un élément de  $E$  appartient à l'objet à seuiller ou non.

La méthode de seuillage par hystérésis consiste à définir non pas un seuil mais deux seuils :  $\sigma_{bas}$  et  $\sigma_{haut}$ . De plus, cette méthode inclut une notion de connexité entre les éléments de  $E$ . Un élément  $x \in E$  est appartient à :

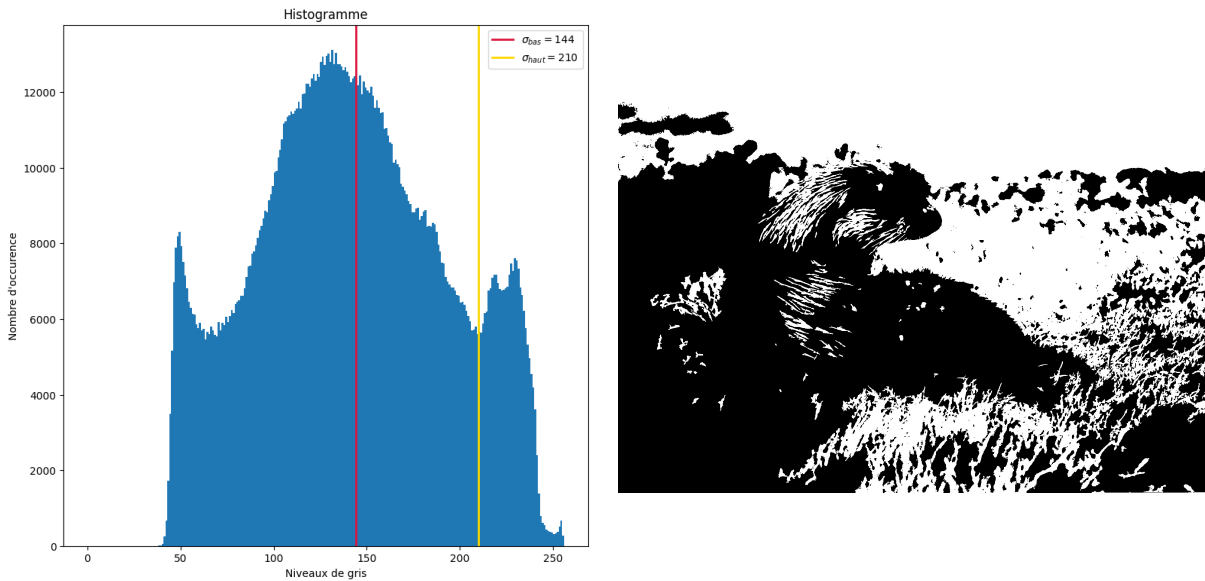


FIGURE 1.7 – Résultat de la méthode par hystérésis pour segmenter une image.

- l'objet si  $I(x) \geq \sigma_{haut}$  ;
- l'objet si  $\sigma_{bas} \leq I(x) < \sigma_{haut}$  et s'il est connexe avec au moins un élément  $y \in E$  tel que  $I(y) \geq \sigma_{haut}$  ;
- au reste de l'image sinon.

Pour le premier seuil, il est possible d'utiliser une méthode de seuillage automatique. Définir une marge de tolérance permet de définir le second seuil. Cela permet de rester robuste aux objets dont les contours apparaissent en dégradé d'intensité sur l'image.

La figure 1.7 montre la segmentation résultante de la méthode par hystérésis avec comme seuil  $\sigma_{bas} = 144$  et  $\sigma_{haut} = 210$ . On observe que la segmentation crée des zones plus grandes (moins de petites zones). Mais le résultat ne suffit pas à identifier la loutre.

L'avantage de cette méthode est de ne pas être plus longue en temps de calcul tout en apportant une notion de localité. En effet, pour un pixel entre les deux seuils, le voisinage du pixel est analysé. Par rapport à un seuillage fixe, cette méthode est plus robuste.

Néanmoins, la recherche de seuil reste le point faible de la méthode. En utilisant un choix de seuillage automatique, la méthode dépendra des limites des méthodes utilisées pour déterminer les seuils. Des études ont essayé de réduire les limites dues au choix du seuillage [Medina-Carnicer et al., 2009, Khan et al., 2020].

## 1.4 Approche par gradient

L'approche par contour (ou par frontière), parfois appelée approche par gradient, où l'on cherche à trouver les contours des classes et non directement celles-ci, repose principalement sur la notion de gradient (local) d'une image. En effet, les maxima locaux de la dérivée première correspondent aux zones très contrastées de l'image, c'est-à-dire là où il

y a un important changement de valeurs. De même, les zéros locaux de la dérivée seconde correspondent aux contours de l'image.

Nous allons dans un premier temps définir la notion de gradient d'une image ([sous-section 1.4.1](#)), puis nous verrons que le calcul peut-être effectué à partir de la convolution que nous définirons ([sous-section 1.4.2](#)). Enfin nous verrons la méthode de Canny, qui est une méthode de segmentation utilisant cette notion de dérivée ([sous-section 1.4.3](#)).

### 1.4.1 Gradient d'une image

En termes mathématiques, le gradient d'une fonction de deux variables, à savoir ici l'intensité en fonction des coordonnées de l'image, est un vecteur de dimension 2 dont les coordonnées sont les dérivées selon les directions horizontale et verticale. Le calcul de la dérivée dans un espace discret passe par une approximation, qui se résume à une différence à l'échelle locale. La dérivée partielle en  $x$  est donnée par l'approximation d'Euler :

$$\forall (i, j) \in E, \quad \frac{\partial I}{\partial x}(i, j) \approx \frac{1}{p_x} [I(i, j) - I(i - 1, j)]$$

où  $p_x$  est le pas de discrétisation et sera considéré comme unitaire. On obtient de la même manière la dérivée partielle en  $y$ , notée  $\frac{\partial I}{\partial y}$ . Le gradient  $\nabla I$  est alors le vecteur composé de  $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ .

L'approximation d'Euler peut être fortement améliorée en considérant que la fonction *penne* est discontinue au niveau du pixel analysé. La formule proposée, pour  $x$ , devient la suivante (la dérivée en  $y$  s'obtient de la même manière) :

$$\forall (i, j) \in E, \quad \frac{\partial I}{\partial x}(i, j) \approx I(i + 1, j) - I(i - 1, j)$$

Cependant, ces deux opérateurs d'approximation de la dérivée sont très sensibles au bruit. Il est préférable de lisser les valeurs par une interpolation. Une façon simple d'interpoler les valeurs est de prendre la moyenne. La moyenne est calculée dans l'axe opposé, i.e. le lissage se fait verticalement pour la dérivée horizontale, et horizontalement pour la dérivée verticale. Ainsi, pour le pixel  $(i, j)$  la dérivée en  $x$  avec lissage est égale à :

$$\begin{aligned} \frac{\partial I}{\partial x}|_{\text{lisse}}(i, j) &= \frac{1}{3} \left[ \frac{\partial I}{\partial x}(i, j - 1) + \frac{\partial I}{\partial x}(i, j) + \frac{\partial I}{\partial x}(i, j + 1) \right] \\ &= \frac{1}{3} \left[ I(i + 1, j - 1) - I(i - 1, j - 1) \right. \\ &\quad \left. + I(i + 1, j) - I(i - 1, j) \right. \\ &\quad \left. + I(i + 1, j + 1) - I(i - 1, j + 1) \right] \end{aligned} \tag{1.4}$$

### 1.4.2 Convolution et dérivée

La formule 1.4 obtenue peut se simplifier à l'aide de l'opérateur mathématique de convolution :





FIGURE 1.8 – Image en niveaux de gris. Dérivées en X et Y à l’aide des masques de Sobel.

**Définition 1.4.0.1 (Convolution)** Soit  $I$  une image et  $\omega : E' \subseteq E \rightarrow F$ , appelée filtre ou masque de convolution. La convolution de  $I$  par  $\omega$ , dénotée par  $I * \omega$ , est égale à :

$$\forall (i, j) \in E, \quad (I * \omega)(i, j) = \sum_m \sum_n \omega(m, n) I(i - m, j - n)$$

La formule 1.4 du calcul du gradient en utilisant la convolution est alors

$$\nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) = (I * \omega_x, I * \omega_y), \quad \text{où } \omega_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \text{ et } \omega_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Les masques  $\omega_x$  et  $\omega_y$  ont été proposés par Prewitt. Il est possible de définir d’autres masques pour calculer le gradient [Hast, 2014]. Par exemple, Sobel [Sobel and Feldman, 1968] propose pour calculer la dérivée, tout en limitant la sensibilité au bruit, les deux masques suivants (appelés filtres de Sobel–Feldman) :

$$s_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \text{et} \quad s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

La figure 1.8 montre les dérivées en  $x$  et en  $y$  de l’image d’origine à l’aide des masques de Sobel. D’autres masques ont été proposés afin d’améliorer l’estimation de la dérivée et/ou de réduire l’influence bruit. On observe que les zones de forts contraste (forte variation d’intensité) sont mises en évidence selon l’axe  $x$  ou  $y$ .

Par ailleurs, l’opérateur de convolution n’est utilisé que pour le calcul de la dérivée. En effet, avec cette opération il est possible de lisser une image (filtre moyenne, filtre gaussien), de calculer une approximation du Laplacien, d’effectuer des gradients directionnels [Kirsch, 1971], etc.

### 1.4.3 Canny

Canny propose un processus pour détecter des contours [Canny, 1986]. D’abord, afin de rendre le calcul du gradient plus robuste au bruit, un pré-traitement est nécessaire. Le pré-traitement est une convolution par un noyau gaussien. Le calcul du gradient est fait par un masque de Sobel. Il convient ensuite de supprimer les non-maxima locaux. Si la



FIGURE 1.9 – Résultat de la détection des contours par la méthode de Canny,  $\sigma_{bas} = 150$  et  $\sigma_{haut} = 220$  ont été choisis pour l'étape 5, le seuillage par hystérésis.

norme du gradient en  $(i, j)$  est inférieure à celle des voisins selon la direction du gradient  $\theta(i, j)$  alors le pixel est retiré de la liste des pixels appartenant à un contour. Enfin, un seuil par hystérésis est fait sur les pixels restants. La méthode de Canny se résume ainsi :

1. Effectuer le pré-traitement ;
2. Calculer une approximation du gradient :  $\nabla I = (G_x, G_y) = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$  ;
3. Calculer la norme et l'orientation du gradient :

$$\|\nabla I\| = \sqrt{G_x^2 + G_y^2} \quad \text{et} \quad \theta(x, y) = \arctan\left(\frac{G_y}{G_x}\right); \quad (1.5)$$

4. Supprimer des non-maxima locaux dans la direction du gradient ;
5. Seuiller les maximums locaux restants.

La [figure 1.9](#) montre un exemple de la méthode de Canny pour détecter les contours. Dans l'exemple  $\sigma_{bas} = 150$  et  $\sigma_{haut} = 220$  ont été choisis pour effectuer le seuillage par hystérésis (étape 5).

Cette méthode de détection des contours est rapide. Elle est plus robuste au bruit que le seuillage de la norme du gradient. De plus, il est facile de modifier une étape du processus pour l'adapter à un contexte donné (changer le pré-traitement, modifier le calcul du gradient, de l'orientation, etc.).

Malheureusement, bien que plus robuste qu'un seuillage, la méthode de Canny reste sensible au bruit. En outre, le paramétrage (choix pour la variance de la gaussienne, choix pour les seuils dans le seuillage par hystérésis) reste à faire manuellement.

## 1.5 Accroissement de régions

Cette famille de segmentation est une approche par classes où l'on cherche l'intégralité de la classe (et non simplement la frontière). Nous allons nous intéresser à une famille de méthodes, à savoir l'accroissement de régions. L'idée principale consiste à décomposer

l'image en régions, puis à les fusionner entre-elles de manière à obtenir la segmentation voulue.

### 1.5.1 Décomposition en régions

La décomposition/fusion (*split and merge* en anglais) se base sur la décomposition de l'image en blocs élémentaires [Horowitz, 1974]. La décomposition la plus naturelle est celle qui considère chaque pixel comme un bloc élémentaire. Une autre décomposition consiste à découper l'image en bloc carré de taille fixe. Des structures plus complexes peuvent être utilisés à l'instar d'un 4-arbres (*quadtree*).

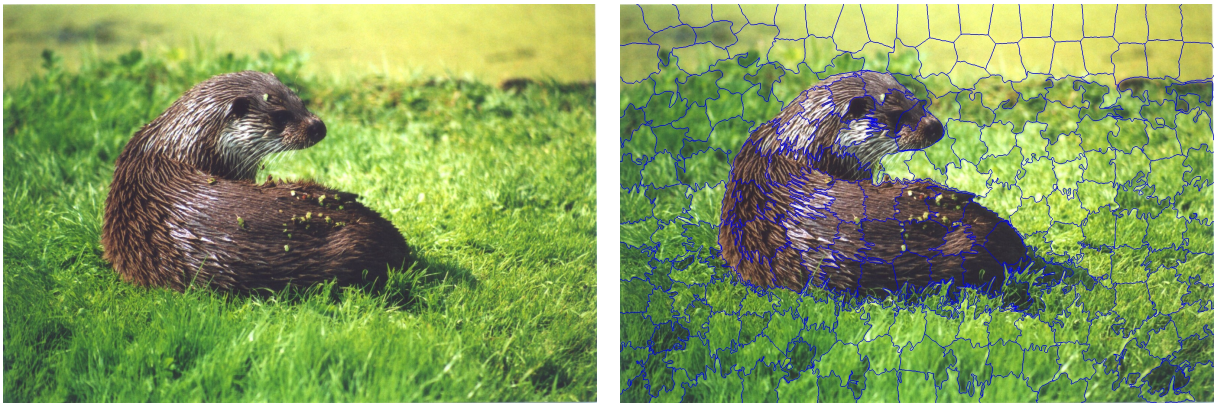
Par exemple, l'algorithme SLIC (*Simple Linear Iterative Clustering*) est une méthode rapide de décomposition qui subdivise une image en  $K$  régions, appelées *superpixels* en fonction du contenu de l'image [Achanta et al., 2010]. La segmentation en *superpixels* est très utile car elle allège la quantité de données à traiter, et permet de réduire le temps nécessaire à la fusion des régions (ou de prendre plus de temps pour mieux fusionner). L'avantage par rapport à la décomposition en blocs carrés de taille fixe est que les superpixels permettent d'avoir un ensemble de régions ayant des propriétés définies. Les superpixels peuvent être utilisés à d'autre fins (segmenter chaque superpixel un à un, etc.).

La suite de la section présente l'algorithme SLIC. L'algorithme repose sur le principe du K-means. Voici les étapes clefs de l'algorithme :

1. Initialiser les  $K$  centres des superpixels. Les centres sont équidistants les uns des autres d'une distance  $S$  (distance calculée d'après la taille de l'image et  $K$ ). Chaque centre est un vecteur en 5 dimensions :  $(i, j, L, a^*, b^*)$  où  $L, a^*$  et  $b^*$  sont les valeurs des 3 composantes de l'image dans la colorimétrie Lab ;
2. Initialiser la carte de superpixels et la carte des distances. Les distances sont initialisées à l'infini et les pixels sont attribués au superpixel numéro 0 ;
3. Pour chaque centre, on calcule la distance des pixels se trouvant dans un rayon de  $2 \times S$  au centre. Si la distance est plus petite que celle en mémoire pour le pixel, elle est modifiée en mémoire et le pixel fait maintenant partie du superpixel ;
4. Mettre à jour la position des centres (5 dimensions) de chaque superpixels (mise à jour selon la valeur moyenne de tous les pixels du superpixel) ;
5. Répéter les étapes 2 à 4 jusqu'à ce que l'algorithme converge, c'est-à-dire, que le déplacement moyen des centres entre deux itérations soit plus petit qu'un certain seuil.

Un post-traitement est effectué pour s'assurer que chaque classe forme un bloc unique et continu. Enfin, l'algorithme SLIC a tendance à créer de petites régions détachées de leurs centres (voir [figure 1.10](#) ou [figure 1.11](#)). D'autres méthodes ont été proposées pour améliorer cette décomposition en superpixels [Wang et al., 2017].

La [figure 1.10](#) montre la décomposition de l'image à l'aide de l'algorithme SLIC. Les contours des superpixels sont en bleus. On observe que les régions homogènes en terme de couleurs forment des formes plus rectangulaires, alors que les régions non homogènes sont plus allongées et ont une forme quelconque.



(a) Image en entrée

(b) Délimitation des superpixels en bleu

FIGURE 1.10 – Résultat de la décomposition d’une image en superpixels à l’aide de l’algorithme SLIC.

### 1.5.2 Fusion des régions

Une fois le processus de décomposition de l’image  $I$  effectué, il faut fusionner pour chaque région obtenue. Cette phase de fusion se fait de manière itérative, c’est-à-dire jusqu’à ce qu’il n’y ait plus la possibilité de fusionner. Avant l’étape de fusion, il est nécessaire de calculer un vecteur descripteur pour chaque bloc. La fusion d’un bloc avec la région courante se fait de la manière suivante : l’ajout du bloc à la nouvelle région se fait si la nouvelle région obtenue possède un vecteur descripteur similaire à l’ancienne région. Il est possible de fusionner un bloc avec la région actuelle.

L’un des vecteurs descripteurs simple est la moyenne (des niveaux de gris) de la région. Il peut cependant être possible de redéfinir ce critère pour l’adapter (avec l’écart-type ou l’entropie par exemple). Il est également possible d’avoir plus de critères au sein du vecteur : combinaison de l’entropie et de la variance, etc. À la fin, la méthode de décomposition/fusion fournit un partitionnement complet de l’image initiale, ce qui correspond bien à une segmentation.

La [figure 1.11](#) montre un résultat possible de la fusion des régions. Dans l’exemple, Les régions initiales choisies sont celles qui touchent le bord de l’image. Le vecteur descripteur est composé de la moyenne pour chaque composante couleur. Enfin, le seuil de tolérance pour la fusion d’une région a été fixé à 22.

L’avantage de cette approche est de créer des classes qui ont des caractéristiques différentes. L’approche est simple, décomposer l’image et fusionner les résultats obtenus à la décomposition. Cela permet de s’adapter à des contextes très variés [[Ma et al., 2017](#), [Karsa and Shmueli, 2018](#)]. Le choix des régions initiales peut s’automatiser. La fusion permet aussi de prendre en compte des critères aussi large et variés que possible (colorimétrie, orientation, etc.).

Les problèmes potentiels de cette méthode sont la sur-segmentation. Cela survient si l’image est très texturé et/ou si les descripteurs sont mal choisis. La segmentation dépend fortement du choix du calcul de la similarité ainsi que des seuils définis pour la fusion.

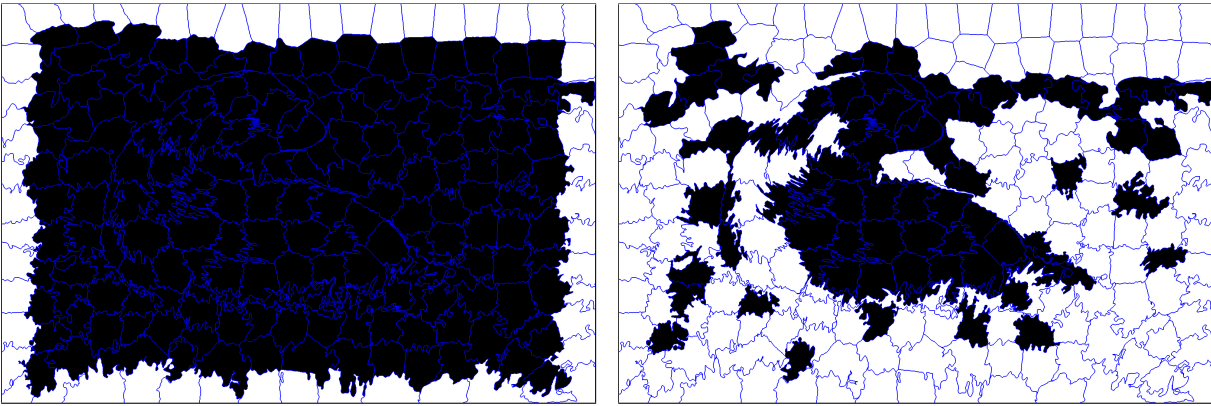


FIGURE 1.11 – À gauche, la région initialisée avec les superpixels du bord de l’image. À droite, la région finale obtenue par fusion avec pour vecteur descripteur la moyenne des composantes RVB et un seuil de 22.

Autrement dit, la méthode peut vite devenir très compliquée à paramétrer. Enfin, en pratique, la méthode est assez coûteuse en temps de calcul. Ce temps de calcul pouvant vite croître selon le critère de similarité et/ou des descripteurs à générer.

## 1.6 Contours actifs (ou Snakes)

Les contours actifs ont été proposés par Kass, Witkins et Terzopoulos en 1988 [Kass et al., 1988]. Il s’agit de courbes mathématiques qui vont se déformer et se déplacer dans l’image pour venir coller les contours d’un objet à segmenter. Segmenter par cette approche permet donc d’identifier un objet précisément mais ne permet pas une explication globale de la scène.

La définition mathématique d’un contour actif  $c$  est la suivante :

**Définition 1.6.0.1** Soit  $I : E \rightarrow F$  une image (avec  $E \subset \mathbb{N}^2$ ). Un contour actif est une courbe  $c : [0, 1] \rightarrow E$  telle que :

$$\forall s \in [0, 1], \quad c(s) = (x(s), y(s))$$

où  $x : [0, 1] \mapsto E$  et  $y : [0, 1] \mapsto E$ .

Le contour a une existence propre (peut être discontinu) et est initialisé dans l’image. La position initiale est en général dans un voisinage assez proche du contour de l’objet à segmenter. Les déformations et déplacements du contour se font sous l’effet de plusieurs forces définies par leurs énergies. Dans la version initiale [Kass et al., 1988], le contour est soumis à trois forces :

- une force interne, correspondant à la résistance à la tension et à la torsion du contour, directement liée à son aspect lissé ;
- une force externe (ou de contrainte) qui est introduite selon les contraintes spécifiques au problème considéré. Par exemple, elle permet de tendre vers une forme donnée (circulaire, ovale, etc.) ;

- et une force de potentiel induite par l'image elle-même.

L'énergie totale du contour actif correspond à la somme des énergies des trois forces :

$$E_{totale}(c) = \int_0^1 \left( E_{interne}(c(s)) + E_{externe}(c(s)) + E_{image}(c(s)) \right) ds$$

Le principe de cette technique est de minimiser l'énergie totale en faisant évoluer la courbe  $c$ . Bien que Kass et al. définissent ces énergies, il est tout à fait possible d'en changer la formulation selon le contexte et l'application voulue [Chan and Vese, 2001b, Zhang et al., 2015].

La force de cette méthode est de s'adapter à tout type de contexte, car il suffit d'ajouter ou de modifier les énergies.

En revanche, deux défauts à cette méthode sont à souligner. Le premier est le temps de calcul qui est important. L'autre inconvénient est le paramétrage difficile car il n'est pas aisé de connaître l'influence exacte des paramètres sur le résultat obtenu.

### 1.6.1 Énergie potentielle

Détaillons rapidement la méthode proposée par Kass et al. [Kass et al., 1988]. L'énergie potentielle d'un contour actif est calculée à partir des informations de l'image. L'information la plus simple à définir est l'information d'intensité. Pour attirer le contour vers les pixels de faible intensité, on définira l'énergie potentielle par :

$$E_{image}(c(s)) = I(c(s)) \quad (1.6)$$

Cette énergie est minimale lorsque l'intensité est minimale. Cependant, dans la plupart des applications, nous cherchons à ce que le contour se dirige vers les lignes de fort gradient. Cette énergie est alors exprimée par :

$$E_{image}(c(s)) = -\nabla I(c(s)) \quad (1.7)$$

Il est tout à fait possible de définir l'énergie potentielle comme une combinaison linéaire des deux propositions précédentes :

$$E_{image}(c(s)) = \delta I(c(s)) + \varepsilon \nabla I(c(s)) \quad (1.8)$$

où  $\delta$  et  $\varepsilon$  permettent de réguler l'influence de chacune d'elle.

### 1.6.2 Énergie interne

L'énergie potentielle semble suffisante pour réussir à déplacer le contour jusqu'aux bords de l'objet ciblé. Cependant, il s'agit d'une énergie locale et le processus de minimisation peut conduire à un minimum local. L'énergie interne va permettre de limiter les déformations du contour pour réduire le risque d'un minimum local :

$$E_{interne}(c(s)) = \alpha(s)c'(s) + \beta(s)c''(s) \quad (1.9)$$

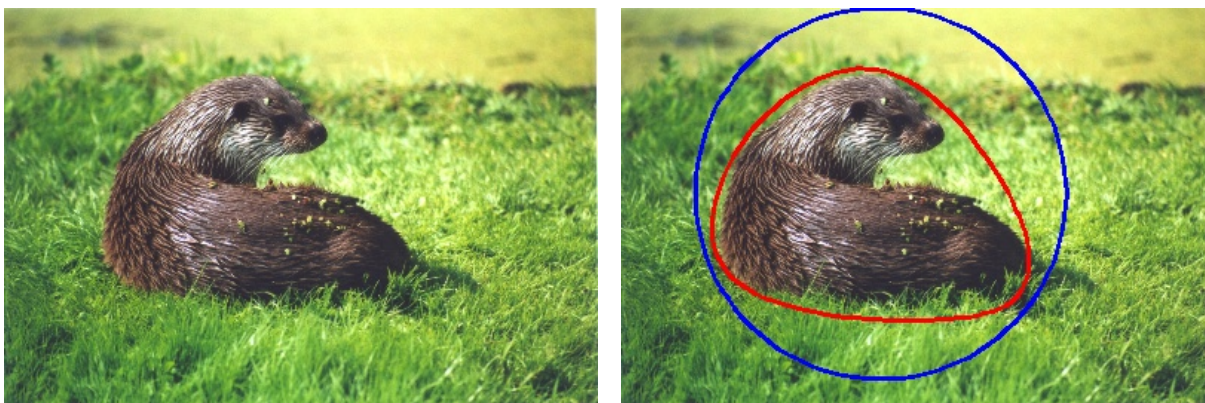


FIGURE 1.12 – À gauche, l'image en entrée (la taille a été divisée par 4 pour avoir une convergence plus rapide du contour). À droite, en bleu le contour initial (100 points) et en rouge le contour final obtenu avec  $\alpha(s) = 0.03$ ,  $\beta(s) = 1.0$ ,  $\delta = 2$  et  $\varepsilon = -4$ .

La dérivée première  $c'(s)$  représente les variations de la longueur de la courbe. Cela permet de paramétrer la résistance à la tension. D'un point de vue de l'utilisation, le coefficient  $\alpha(s)$  régule l'élasticité du contour.

La dérivée seconde  $c''(s)$  exprime quant à elle la courbure du contour. Elle permet d'avoir plus ou moins de déformations locales. Nous pouvons voir le paramètre  $\beta(s)$  comme contrôlant la raideur du contour.

### 1.6.3 Énergie externe

Nous l'avons dit, l'énergie externe d'un contour actif est une énergie décrite dans le cadre d'une application spécifique. Elle peut par exemple représenter une forme (cercle, ellipse, etc.) si elle est connue d'avance. Dans l'article de Kass et al., il est fait mention d'une énergie externe qui agit tel un ressort reliant le contour et à une position-clef définie.

La [figure 1.12](#) montre un exemple de contour actif (celui proposé initialement par Kass et al.) sur l'image de la loutre en entrée. Le gradient d'une image couleur est la somme des gradients de chaque composante. La courbe bleue montre le contour initial choisi. Afin d'accélérer la convergence, les dimensions de l'image ont été divisées par 4. Les paramètres du contour sont les suivants,  $\alpha(s) = 0.03$ ,  $\beta(s) = 1.0$ ,  $\delta = 2$  et  $\varepsilon = -4$ . La courbe rouge montre le contour après convergence.

## 1.7 Apprentissage automatique - Réseaux de neurones

Les réseaux de neurones sont devenus une méthode classique pour de nombreuses tâches. La segmentation en fait partie. Nous allons décrire brièvement les principes d'un réseau de neurones, puis voir certaines classes de neurones spécifiques au traitement d'images et donner quelques exemples de réseaux de neurones largement testés dans la segmentation d'images.

### 1.7.1 Les principes de base

Un réseau de neurones artificiels est composé de neurones qui forment les unités de base. La première définition est due à Lettvin et al. en 1959 [Lettvin et al., 1959]. Un neurone est décrit par un ensemble d'entrées  $\{x_1, \dots, x_n\}$  auxquelles sont associées, pour chacune d'elle, un poids  $w_i$ . Ce poids décrit l'importance de l'entrée  $x_i$  du neurone. La valeur initiale d'entrée du neurone est la somme des entrées  $x_i$  pondéré par les poids  $w_i$ . À cette somme on ajoute un biais  $b$  qui correspond à l'activation du neurone. L'activation, d'un point de vue biologique, est un potentiel d'activation. C'est le seuil qui entraîne une réponse du neurone s'il est atteint. Une fois la sommation calculée, une fonction est appliquée.

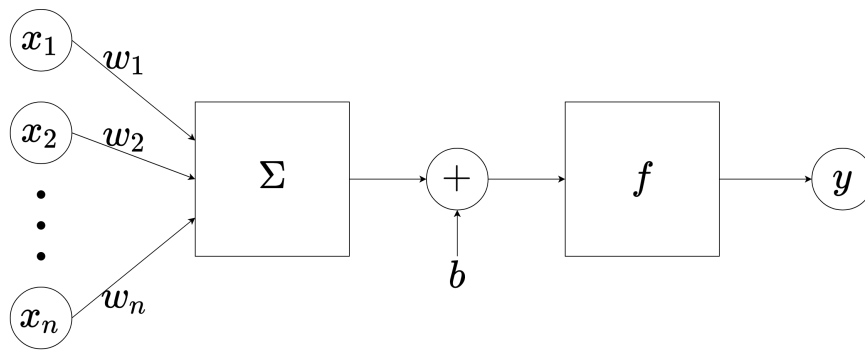


FIGURE 1.13 – Schéma d'un neurone artificiel.

La sortie du neurone, notée  $y$ , est le résultat de la fonction, nous avons :

$$y = f\left(\sum_i^n w_i x_i + b\right) = f(W\mathbf{x} + b) \quad (1.10)$$

où  $f$  est la fonction d'activation dont nous détaillerons le fonctionnement ultérieurement,  $\mathbf{x} = (x_1, \dots, x_n)$  est le vecteur d'entrée et  $W$  le vecteur des poids. Ce processus est illustré dans la [figure 1.13](#).

Le réseau de neurones à propagation (*feedforward neural network* en anglais) est le plus élémentaire des réseaux de neurones. En effet, ce dernier se compose uniquement de couches de neurones, qui sont connectées entre elles. Les sorties d'une couche de neurones forment les entrées de la couche suivante. La [figure 1.14](#) montre un exemple de réseau de neurones à propagation.

Comme le montre la [figure 1.14](#), un tel réseau se décompose en trois grandes familles de neurones :

- les neurones d'entrée, qui représentent les données qui doivent être traitées par le réseau. À noter qu'aucun calcul n'est fait à cette étape du réseau ;
- les neurones *cachés*, qui effectuent les calculs et transfèrent l'information d'une couche à une autre ;
- les neurones de sortie, qui constituent le résultat obtenu par le réseau.



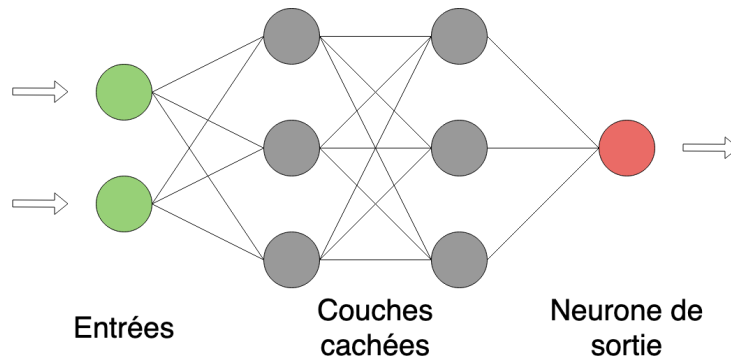


FIGURE 1.14 – Exemple d’un réseau de neurones à propagation.

S’il n’y a qu’une seule couche *cachée*, nous parlons alors de réseau mono-couche. S’il y en a plusieurs, nous parlons de réseau multi-couches.

En se basant sur l’équation 1.10, il est possible de décrire de manière formelle un réseaux multi-couches. Soient :

- $e$  le nombre de neurones d’entrées ;
- $s$  le nombre de neurones de sorties ;
- $m + 1$  le nombre de couches (entrée et sortie incluses) ;
- $n_i$  le nombre de neurones de la couche  $i$  ( $n_0 = e$  et  $n_m = s$ ) ;
- les fonctions affines :  $\forall \mathbf{x} \in \mathbb{R}^{n_{j-1}}, A_i^j(\mathbf{x}) = W_i^j \mathbf{x} + b_i^j$ ,  $W_i^j$  les poids du neurone  $i$  de la couche  $j$ ,  $b_i^j$  le biais numéro  $i$  de la couche  $j$  ;
- $f_i^j$  la  $i$ -ème fonction d’activation de la couche  $j$  ;

Le vecteur de sortie  $\hat{\mathbf{y}} = (y_1, \dots, y_s)$  est alors défini par la relation de récurrence :

$$\forall \mathbf{x} \in \mathbb{R}^e, \begin{cases} \mathbf{a}^0 &= \mathbf{x} \\ a_i^j &= f_i^j(A_i^j(\mathbf{a}^{j-1})) \\ \hat{y}_i &= f_i^m(A_i^m(\mathbf{a}^{m-1})) \end{cases} \quad \begin{array}{l} \text{avec } 1 \leq j \leq m \text{ et } 1 \leq i \leq n_j \\ \text{avec } 1 \leq i \leq s \end{array} \quad (1.11)$$

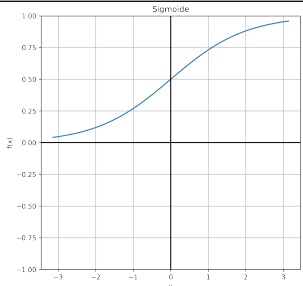
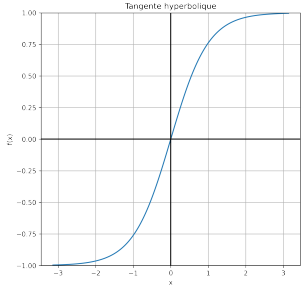
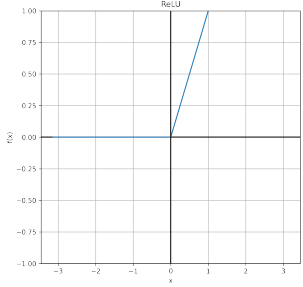
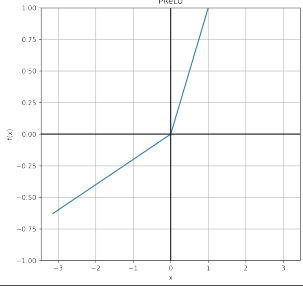
## 1.7.2 Fonctions d’activation

Les fonctions d’activation  $f_i^j$  ont pour but d’introduire de la non-linéarité dans le réseau. En effet, si ces fonctions n’existaient pas, le réseau ne serait qu’une application linéaire, ce qui en limiterait son usage. La non-linéarité des fonctions d’activations permet également de mieux s’ajuster aux données qui ne sont probablement pas linéaires.

Cependant, les fonctions d’activation doivent être différentiables en tout point. C’est une condition nécessaire pour effectuer l’algorithme de rétropropagation qui sera utilisé pour ajuster la valeur des poids et de biais. Si ce n’est pas le cas, la rétropropagation peut induire des erreurs. Certaines propriétés sur les fonctions d’activations permettent des optimisations dans le calcul [Snyman et al., 2005] ou bien d’obtenir certaines propriétés [Wu, 2009].

Il existe beaucoup de fonctions d’activation. Le tableau 1.1 illustre quelques exemples de fonctions d’activations courantes.

TABLE 1.1 – Exemples de fonctions d'activation.

Nom	Équation	Graphique
Sigmoïde	$\frac{1}{1 + e^{-x}}$	
Tangente hyperbolique	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	
Rectified Linear Unit (ReLU)	$\begin{cases} 0, & \text{si } x \leq 0 \\ x, & \text{si } x > 0 \end{cases}$	
Parametric ReLU	$\begin{cases} \alpha x, & \text{si } x \leq 0 \\ x, & \text{si } x > 0 \end{cases}$	

### 1.7.3 Apprentissage d'un neurone

La phase d'apprentissage consiste à modifier la valeur des poids  $w_i^j$  et des biais  $b_i^j$  des couches afin d'obtenir le meilleur résultat (régression, classification, segmentation, etc.). La notion de meilleure fixe alors un objectif à minimiser : minimiser une fonction de coût  $\psi$  (*loss function* en anglais) qui calcule l'erreur entre la sortie du réseau  $\hat{\mathbf{y}}$  et la sortie attendue  $\mathbf{y}$ . Comme pour les fonctions d'activations, de nombreuses fonctions de coût ont été proposées.

Dans le cadre de la segmentation d'images, notamment celui d'une binarisation, nous pouvons citer deux fonctions de coût très utilisées : l'entropie croisée binaire et le coefficient de corrélation de Matthews (MCC). L'entropie binaire croisée se définit par la formule suivante :

$$\mathcal{L}_{BCE} = - \sum_i [y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)]$$

où  $\hat{y}_i$  est la valeur de sortie pour le pixel  $i$  et  $y_i$  est la sortie du réseau pour le pixel  $i$ .

On remarque que si  $y_i$  tends vers 0 alors seul le terme  $\ln(1 - \hat{y}_i)$  compte et tends 0 quand  $\hat{y}_i$  tends vers 0. De même, si  $y_i$  tends vers 1 alors seul le terme  $\ln(\hat{y}_i)$  compte et tends vers 0 quand  $\hat{y}_i$  tends vers 1. Autrement dit, minimiser cette fonction revient à faire en sorte que la sortie  $\mathbf{y}$  soit égale à la vérité terrain  $\hat{\mathbf{y}}$ .

Une autre fonction de coût est le coefficient de corrélation de Matthews [[Abhishek and Hamarneh, 2021](#)] (voir l'équation 1.1). Pour utiliser ce coefficient en tant que fonction de perte il faut rendre la formule différentiable. Cela est fait en introduisant la version différentiable des valeurs de la matrice de confusion :

$$\begin{aligned} \text{TP} &= \sum_i^N y_i \hat{y}_i & \text{FP} &= \sum_i^N y_i (1 - \hat{y}_i) \\ \text{FN} &= \sum_i^N (1 - y_i) \hat{y}_i & \text{TN} &= \sum_i^N (1 - y_i) (1 - \hat{y}_i) \end{aligned}$$

La formule du MCC devient après simplification :

$$\mathcal{L}_{MCC} = 1 - \frac{\sum y_i \hat{y}_i - \frac{\sum y_i \sum \hat{y}_i}{N}}{\sqrt{\sum y_i \hat{y}_i - \frac{\sum y_i (\sum \hat{y}_i)^2}{N} - \frac{(\sum y_i)^2 \sum \hat{y}_i}{N} + \left(\frac{\sum y_i \sum \hat{y}_i}{N}\right)^2}} \quad (1.12)$$

Cette fonction est différentiable en chaque point et il est possible de vérifier que le minimum est atteint quand  $FP = 0$  et  $FN = 0$ , autrement dit quand l'ensemble des pixels est bien classé.

Revenons à la fonction de coût de manière générale. Pour minimiser cette fonction, l'un des algorithmes d'optimisation est celui de rétropropagation qui se base sur la méthode de

descente du gradient. L'algorithme de descente du gradient va permettre d'approcher un minimum local de la fonction en cherchant les points auxquels son gradient est nul. Dans le cas particulier des réseaux de neurones, cela revient à dire qu'à chaque itération, on effectue le calcul du gradient de la fonction par rapport à chaque poids du réseau. Puis, en fonction de la valeur, les poids sont mis à jour. Un paramètre  $\alpha$ , appelé coefficient d'apprentissage (*learning rate* en anglais) est introduit afin de pondérer la mise à jour. Le coefficient d'apprentissage fait partie des hyperparamètres, ce sont les paramètres qui ne sont pas modifiés lors de l'apprentissage comme c'est le cas pour les poids, les biais, etc.

Il existe des problèmes qui peuvent survenir durant le processus de l'apprentissage. Parmi eux, il y a le sur-apprentissage. Il se définit comme un mauvais paramétrage des hyperparamètres, notamment ceux issus de la descente du gradient, ou bien comme une base de données d'apprentissage de *mauvaise* qualité (trop petite, non représentative, etc.). Si le réseau a sur-appris, il est capable de capturer des informations inutiles pour accomplir sa tâche, et il devient souvent incapable de généraliser les caractéristiques des données. Cela limite son utilisation à de nouvelles données (non issues de la base d'apprentissage).

Un autre problème est celui de l'atteinte même du minimum. En effet, il est possible que la descente de gradient atteigne un minimum local de la fonction et non le minimum global. Pour réduire ce risque des méthodes ont été mises en œuvre. Parmi elles nous allons aborder l'algorithme d'optimisation ADAM largement utilisé depuis son introduction [Kingma and Ba, 2014]. Adam est un algorithme d'optimisation qui peut être utilisé à la place de la descente de gradient (stochastique) pour mettre à jour les poids du réseau de manière itérative en fonction des données d'apprentissage. L'un des défauts de la descente de gradient est le suivant : le coefficient d'apprentissage est le même pour toutes les mises à jour de poids et ne change pas au cours de l'apprentissage. ADAM est la fusion de deux autres méthodes qui avaient été proposées *Adaptive Gradient Algorithm* (AdaGrad) et *Root Mean Square Propagation* (RMSProp) [Ruder, 2016]. Au lieu d'adapter le coefficient d'apprentissage  $\alpha$  sur les valeurs de la moyenne des gradients (comme dans RMSProp), ADAM utilise également la moyenne des variances non centrées. Plus précisément, l'algorithme calcule une moyenne exponentielle du gradient et du gradient au carré (sur une fenêtre temporelle glissante). Deux paramètres  $\beta_1$  et  $\beta_2$  sont introduits afin de contrôler les taux de décroissance de ces deux moyennes.

#### 1.7.4 Régularisation d'un réseau de neurones

En employant la descente de gradient pour apprendre les poids des neurones un problème survient. Ce problème est celui de la valeur du gradient qui tends vers 0 au fur et à mesure que l'on remonte le réseau. Cela entraîne que les poids ne se mettent pas à jour. Afin de pallier ou du moins limiter ce problème, des techniques de régularisation ont été proposées. Nous allons nous limiter à trois méthodes : le *shakeout*, la normalisation et l'augmentation de données.

### 1.7.4.1 Dropout et Shakeout

La première technique est le *Dropout*. Cette méthode consiste à éliminer, de manière aléatoire selon une probabilité  $\tau$ , certains neurones à chaque itération afin de diminuer le nombre de paramètres du réseau. Le nombre de poids mis à jour est réduit. Elle peut être interprétée comme un moyen de régulariser un réseau de neurones en ajoutant du bruit à ses unités cachées. Cette idée a déjà été utilisée dans le contexte des auto-encodeurs pour débruiter [Vincent et al., 2008] où du bruit est ajouté aux entrées d'un auto-encodeur et la sortie du réseau n'est pas modifiée. Cette technique peut se formaliser. En reprenant la notion introduite dans la sous-section 1.7.1 et en fixant la couche sur laquelle est appliquée le *dropout* (c'est-à-dire en fixant  $j$  afin d'alléger la notation) nous avons l'opération d'une couche :

$$\begin{aligned}\mathbf{z}^j &= W^j \mathbf{a}^{j-1} + b^j \\ \mathbf{a}^j &= f^j(\mathbf{z}^j)\end{aligned}$$

Maintenant, avec le principe de *dropout*, ces équations sont remplacées par :

$$\begin{aligned}r_i^{j-1} &\sim \text{Bernouilli}(\tau) \\ \tilde{a}_i^{j-1} &= r_i^{j-1} a_i^{j-1} \\ \mathbf{z}^j &= W^j \tilde{\mathbf{a}}^{j-1} + b^j \\ \mathbf{a}^j &= f^j(\mathbf{z}^j)\end{aligned}$$

où  $r_i^{j-1}$  est une variable qui a la probabilité  $\tau$  de valoir  $\frac{1}{1-\tau}$  sinon 0. Plus récemment, une généralisation de cette méthode a été proposée. Elle s'intitule *shakeout* [Kang et al., 2018]. Cette méthode prend en entrée un autre paramètre  $c$ . Les deux premières équations deviennent alors :

$$\begin{aligned}r_i^{j-1} &\sim \text{Bernouilli}(\tau) \\ \tilde{a}_i^{j-1} &= \begin{cases} -c s_i^{j-1} & \text{si } r_i^j = 0 \\ \frac{(a_i^{j-1}) + c\tau s_i^{j-1}}{1 - \tau} & \text{sinon} \end{cases}\end{aligned}$$

où  $s_i^{j-1}$  est le signe de la variable  $a_i^{j-1}$ . La technique du *dropout* se retrouve avec  $c = 0$ .

### 1.7.4.2 Couche de normalisation

La méthode de descente de gradient a quelques inconvénients. Lorsque le réseau calcule le gradient en fonction des entrées de n'importe quelle couche, les poids sont mis à jour selon la direction indiquée par le gradient. Mais comme les couches sont empilées les unes après les autres, une légère mise à jour des poids peut produire des valeurs trop fortes ou trop faibles pour le gradient, rendant la mise à jour dans les couches antérieures difficiles. La normalisation restreint la distribution des valeurs des poids, ce qui aide le réseau à produire des valeurs de gradients plus optimales pour la mise à jour des poids.

Il existe différentes façon de normaliser les entrées [Qiao et al., 2019]. Nous allons nous intéresser uniquement à la normalisation par *batch* (*Batch Normalization* ou BN en anglais). La couche BN se concentre sur la normalisation des entrées d'une couche spécifique (c'est-à-dire le résultats des activations des couches précédentes). La normalisation des entrées signifie que les entrées de n'importe quelle couche du réseau doivent avoir une moyenne et une variance unitaire approximativement nulles. Mathématiquement, cela revient à soustraire la moyenne des entrées puis à diviser par l'écart type. L'algorithme 1 résume les calculs effectués pour cette normalisation.

---

**Algorithme 1** : Algorithme de normalisation (BN)
 

---

**Entrée(s)** :  $\mathbf{x} = \{x_1, \dots, x_n\}$  le vecteur d'entrées

$\gamma$  et  $\beta$  deux paramètres d'apprentissage

**Sortie(s)** :  $y_i = BN_{\gamma, \beta}(x_i)$

$$\mu \leftarrow \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma^2 \leftarrow \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} \quad /* \varepsilon \text{ est une valeur pour éviter la division par } 0 */$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta$$


---

### 1.7.4.3 Augmentation de données

La deuxième technique est l'augmentation de données (*data augmentation* en anglais) [Shorten and Khoshgoftaar, 2019]. L'objectif est d'augmenter artificiellement, le nombre des échantillons d'apprentissage en ajoutant de la variabilité dans les données, mais sans affecter leur sémantique. Dans le cadre d'images, nous pouvons ajouter de nouvelles images en effectuant des rotations, des symétries verticale/horizontale, en changeant leurs dimensions ou l'intensité lumineuse.

### 1.7.5 Réseaux de neurones convolutifs

En traitement d'images, la convolution est une opération importante. Elle permet de détecter des contours comme nous l'avons vu. Elle permet également de lisser l'image d'entrée en appliquant un filtre moyenneur, qui comme son nom l'indique va effectuer une moyenne d'un voisinage du pixel. La convolution par un filtre Gaussien a pour objectif de réduire le bruit, i.e. diminuer les variation d'intensités. Toutes ces applications de la convolution ont poussé à essayer d'effectuer la même opération dans un réseau de neurones. Par ailleurs, d'autres couches ont été introduites dans les réseaux de neurones convolutifs apportant chacune leurs avantages. Nous allons aborder, dans cette sous-section, différentes couches largement utilisées dans des réseaux servant à segmenter une image.

### 1.7.5.1 Couche de convolution

Un réseau de neurones convolutif est un réseau de neurones qui utilise l'opération mathématique de convolution, voir [sous-section 1.4.2](#). Nous allons voir ici des paramètres importants de la convolution. Notons que les réseaux de neurones convolutifs utilisent par abus de langage le terme convolution bien qu'il s'agisse d'une corrélation croisée [[Borsellino and Poggio, 1973](#)]. Mais la propriété de commutativité de la convolution n'est pas utilisée par les réseaux.

Une couche de convolution se caractérise par :

- les dimensions du noyau de convolution,  $n$  et  $m$  ;
- le nombre de filtres de convolution  $k$  ;
- le pas de convolution (*stride* en anglais). Il s'agit du décalage du noyau de convolution. Par exemple, un *stride* = (1, 1) revient à dire que le noyau se déplacera de 1 dans ses mouvements horizontaux et de 1 dans ses mouvements verticaux ;
- enfin le *padding*. Ce paramètre permet de définir le dépassement de l'image pour pouvoir appliquer la convolution. Les pixels hors de l'image sont initialisés selon différentes méthodes, la plus simple étant de mettre des zéros.

Afin de faciliter la compréhension par la suite, nous fixerons les dimensions du noyau à 2 avec une taille carrée ( $n = m$ ), un pas  $s$  et un *padding*  $p$ . Nous allons voir deux types de convolution : la convolution classique et la déconvolution.

La convolution classique, il s'agit tout simplement d'avoir une couche du réseau qui effectue de manière classique la convolution. Les objectifs étant de filtrer l'image, de faire le calcul de gradient (de l'image), etc.

L'utilisation de couches convolutives est pertinente dans le sens où, d'une part les paramètres des noyaux de convolutions sont partagés. C'est-à-dire que que deux entrées  $x_i$  et  $x_j$  auront les mêmes poids de convolutions. Autre avantage, celui de la convolution elle-même qui permet d'effectuer des calculs qu'une couche classique ne pourrait faire. Ainsi, cela permet d'extraire davantage d'informations. Enfin, la convolution est équivariante, c'est-à-dire que si l'entrée change d'une certaine façon, la sortie change de la même façon. Ainsi, si l'image d'entrée est translatée, alors les cartes d'activations seront translatées.

Les couches de convolution sont très majoritairement suivies d'une couche de ReLU, ou des variantes. En effet, la fonction ReLU supprime les valeurs négatives des cartes ce qui simplifie les calculs.

La [figure 1.15](#) montre l'influence des paramètres comme le pas, le *padding* ou bien le nombre de filtres  $k$ . Dans le premier cas, la taille des filtres est de  $3 \times 3$ , avec  $s = 1$ ,  $p = 1$  et  $k = 6$ . Dans le deuxième cas, le *stride* est fixé à  $s = 2$ , réduisant la dimension en sortie. Puis, le *padding* est mis à 0. On remarque que cela produit une sortie de même dimension qu'avec les paramètres précédents. Cependant, les valeurs de sortie diffèrent car la convolution n'est pas appliquée sur les mêmes pixels. Enfin, dans le dernier la cas le nombre de filtres  $k$  a été réduit à 3.

### 1.7.5.2 Couche de normalisation dans le cadre d'un réseau convolutif

Dans un réseau convolutif l'entrée de chaque couche est un vecteur  $\mathbf{X} \in \mathbb{R}^{N \times H \times W \times C}$ , où  $N \in \mathbb{N}$  est la taille du *batch*,  $H \in \mathbb{N}$ ,  $W \in \mathbb{N}$  et  $C \in \mathbb{N}$  sont les dimensions de

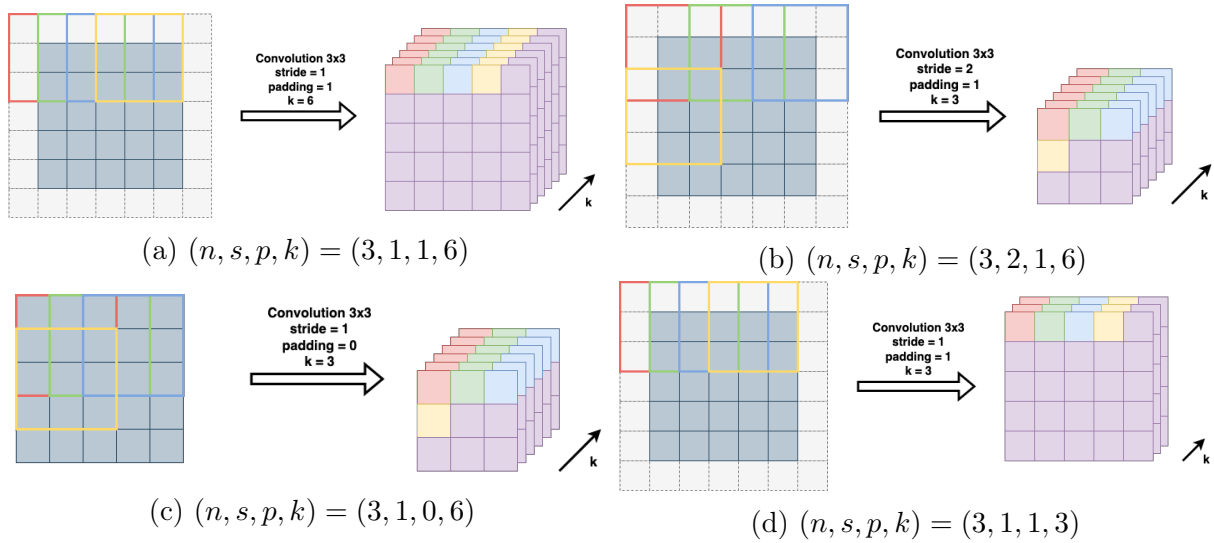


FIGURE 1.15 – Exemples de couches convolutives. Les cases bleutées représentent les entrées, les cases blanches représentent les ajouts dus au *padding*, les cases violettes représentent les sorties après convolution. Celles de couleurs soulignent l'influence des paramètres.

l'*image* (hauteur, largeur et nombre de canaux). Dans ce cadre, la couche de normalisation BN consiste à faire la somme selon les axes  $N, H$  et  $W$ . C'est-à-dire que pour un canal  $c \in [0 \dots C - 1]$ , le canal  $\mathbf{Y}_c$  de la sortie sera obtenue par les calculs suivants :

$$\begin{aligned} \mu_c &= \frac{1}{N \times H \times W} \sum_k \sum_i \sum_j \mathbf{X}_{k,i,j,c} \\ \sigma_c^2 &= \frac{1}{N \times H \times W} \sum_k \sum_i \sum_j (\mathbf{X}_{k,i,j,c} - \mu_c)^2 \\ \hat{\mathbf{X}}_c &= \frac{\mathbf{X}_c - \mu_c}{\sqrt{\sigma_c^2 + \varepsilon}} \\ \mathbf{Y}_c &= \gamma_c \hat{\mathbf{X}}_c + \beta_c \end{aligned}$$

où  $\gamma_c$  et  $\beta_c$  sont deux paramètres appris par le réseau et propres à chaque canal  $c$ .

### 1.7.5.3 Couche d'échantillonnage - *Pooling*

Similaire à la couche de convolution, la couche d'échantillonnage va réduire la taille spatiale des cartes de caractéristiques, mais elle doit conserver les informations les plus importantes [Sun et al., 2017]. Il existe différents types d'échantillonnage [Franchi et al., 2020] : maximum (*max pooling* en anglais), moyen (*average pooling*), etc.

L'échantillonnage consiste à appliquer un noyau de taille  $n \times n$  sur la carte d'activation en le faisant glisser avec un pas (*stride*) préalablement défini (le pas est généralement égal à la taille du noyau pour éviter le phénomène de chevauchement de noyau). Par exemple, le *max pooling* renvoie la valeur maximale de la partie de l'image couverte par le noyau.



Dans un autre cas, nous voudrions prendre la moyenne de tous les éléments couverts par le noyau, cela est fait par l'*average pooling*. Le *max pooling* permet de supprimer le bruit. D'un autre côté, l'*average pooling* effectue une moyenne de l'information, il est possible que l'information qui sort du lot (forte valeur) disparaisse par un tel traitement. Par conséquent, en pratique, le *max pooling* est beaucoup plus utilisé que l'*average pooling*.

La forme la plus courante est une couche d'échantillonnage avec des noyaux de taille  $2 \times 2$  appliqués avec un pas de 2. Cela réduit la dimension de l'entrée par 2, élimine 75% des activations et laisse la dimension de profondeur inchangée.

La [figure 1.16](#) montre deux exemples de couches d'échantillonnage, la première est une couche *max pooling* de taille  $2 \times 2$  avec un pas de 2 pour réduire la dimension par 2. Le deuxième exemple est une couche *average pooling* de taille  $3 \times 3$  avec un pas de 1.

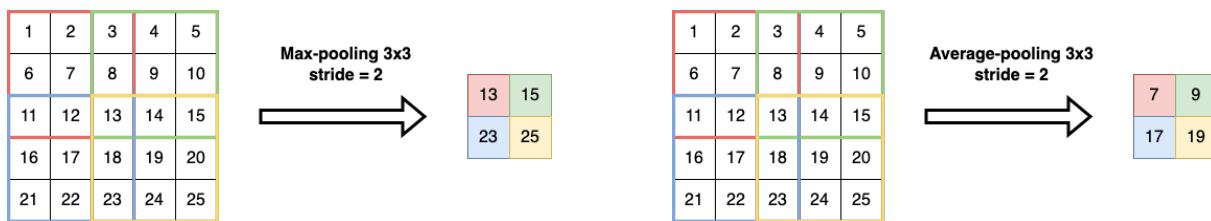


FIGURE 1.16 – Exemple du *max pooling* de taille  $3 \times 3$  avec un pas de 2 et du *average pooling* avec les mêmes paramètres.

#### 1.7.5.4 Couche de suréchantillonnage - *Upsampling*

Cette couche est l'opération inverse de la couche précédente. Cette couche va agrandir la taille spatiale des cartes caractéristiques. Tout comme pour l'échantillonnage, il existe différentes façon de suréchantillonner.

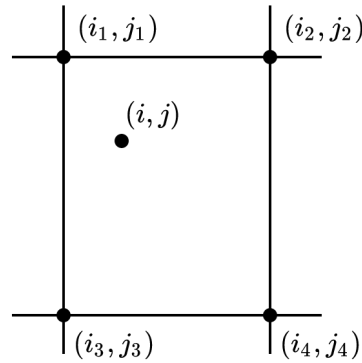
Pour suréchantillonner il est nécessaire de compenser les valeurs de pixels ajoutés. La méthode la plus courante est l'interpolation [[Gonzalez and Woods, 2018](#)]. Il existe différentes manières d'interpoler des valeurs entre elle : au plus proche, bilinéaire, bicubique, etc.

L'interpolation au plus proche, comme son nom l'indique, consiste à affecter à un pixel (ajouté) la valeur du pixel le plus proche. Cette interpolation est simple, mais elle a tendance à produire des artefacts, tels qu'une distorsion importante des bords. Une approche limitant cet effet est l'interpolation bilinéaire. L'idée est de résoudre l'équation suivante :

$$I(i, j) = ai + bj + cij + d$$

où  $a, b, c$  et  $d$  sont les inconnues. Supposons que l'on veuille attribuer une valeur au pixel  $\mathbf{p} = (i, j)$ . Nous connaissons l'équation pour les quatre pixels voisins de  $\mathbf{p}$  (voir [figure 1.17](#)). Nous avons un système d'équations à quatre inconnus à résoudre. Une fois les valeurs de  $a, b, c$  et  $d$  déduites, il est possible d'affecter la valeur au pixel  $\mathbf{p}$ .

D'autres méthodes pour suréchantillonner ont été proposées pour améliorer les performances des réseaux [[Anwar et al., 2020](#)].

FIGURE 1.17 – Interpolation bilinéaire d’un pixel  $(i, j)$ 

### 1.7.5.5 Couche d’attention - *Attention*

Dans le cadre de la segmentation d’images, l’attention est un moyen de ne mettre en évidence que les activations (des neurones) pertinentes pendant l’apprentissage. Cela permet de réduire la quantité de calcul fait sur les activations non pertinentes, ce qui donne au réseau un meilleur pouvoir de généralisation. Essentiellement, le réseau peut prêter *attention* à certaines parties de l’image.

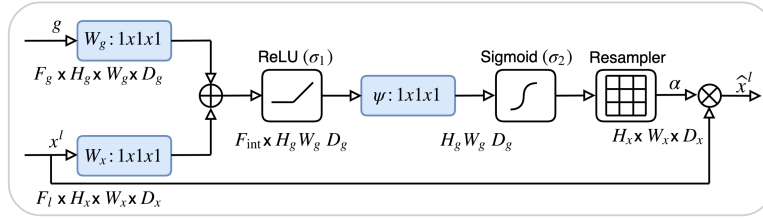
L’attention se présente sous deux formes : *soft* et *hard* [Xu et al., 2015]. La première consiste à calculer l’attention comme une somme pondérée des états cachés de l’encodeur. Les zones de haute pertinence ont des poids plus importants et les zones de faible pertinence ont des poids plus faibles. Au fur et à mesure de l’apprentissage du réseau, l’attention se porte davantage sur les régions ayant un poids plus élevé.

Pour la deuxième forme d’attention, au lieu de la moyenne pondérée de tous les états cachés, on utilise des scores pour sélectionner un seul état caché. Cette sélection pose un problème, parce que la fonction utilisée pour extraire l’état caché n’est pas nécessairement différentiable (par exemple la fonction *argmax*). Des techniques plus complexes sont employées pour pallier à ce souci comme l’extraction automatique de régions pertinentes (*region proposal* en anglais) [Ren et al., 2015].

Les couches d’attention peuvent également avoir un rôle à jouer dans le cadre d’une réduction des paramètres du réseau. En effet, nous l’avons dit, les couches d’attention peuvent allouer les ressources disponibles pour se concentrer de manière sélective sur le traitement d’une partie particulière de l’image au lieu de l’ensemble de la scène. Dans le cadre du traitement d’images, il est possible de définir trois grandes familles d’attentions : celle portée sur les informations spatiales (où faire attention), celle qui se concentre sur les canaux (à quoi faire attention) [Hu et al., 2018, Yang et al., 2020, Zhu et al., 2021a] et la dernière famille étant celle qui combine à la fois l’information spatiale et l’information basée sur les canaux (où et à quoi faire attention).

Le processus d’attention, qui repose sur le mécanisme visuel de discrimination des régions que l’on visualise, peut se résumer par la formule :

$$Attention = F(G(\mathbf{x}), \mathbf{x}) \quad (1.13)$$


 FIGURE 1.18 – Principe de la couche d’attention proposée par [Oktay et al., 2018].

L’entrée donnée à la couche d’attention est représentée par  $\mathbf{x}$ . La fonction  $G$  représente le mécanisme de sélection des régions pertinentes. La fonction  $F$  est la fonction qui modifie l’entrée selon l’attention résultante de  $G$ . La plupart des couches d’attention qui ont été proposées peuvent s’écrire sous la formulation proposée.

Oktay et al. proposent un système d’attention pour la segmentation d’images [Oktay et al., 2018]. Le principe est le suivant :

1. L’entrée est composée des vecteurs  $\mathbf{x}$  et  $\mathbf{g}$  ;
2. le vecteur  $\mathbf{g}$  vient de la couche précédente du réseau. Dans le cas d’un réseau de convolution usuel, ce vecteur a une meilleure représentation des caractéristiques puisqu’il vient d’une couche plus profonde du réseau ;
3. les vecteurs  $\mathbf{x}$  et  $\mathbf{g}$  passent par une convolution (respectivement de noyau  $W_x$  et  $W_g$ ), elle permet que leurs dimensions soient égales si nécessaires ;
4. les deux vecteurs sont additionnés (élément par élément) ;
5. le vecteur résultant passe par une couche d’activation ReLU  $\sigma_1$  et une convolution  $\Psi$   $1 \times 1$  qui ramène aux dimensions d’entrée ;
6. puis il passe par une fonction sigmoïde  $\sigma_2$  qui le met à l’échelle dans la plage de valeurs  $[0, 1]$ , produisant les poids de l’attention (où les coefficients proches de 1 indiquent des caractéristiques plus pertinentes) ;
7. les coefficients d’attention sont rééchantillonnés aux dimensions d’origines de  $\mathbf{x}$  en utilisant une interpolation trilineaire (notée *Interp*).
8. enfin, les poids d’attention sont multipliés (élément par élément) au vecteur  $\mathbf{x}$  d’origine, en mettant à l’échelle le vecteur selon la pertinence. Les poids sont ensuite transmis dans la suite du réseau.

En reprenant la formule générale d’une couche d’attention, la solution proposée par Oktay et al. s’écrit sous la forme suivante :

$$G(\mathbf{x}, \mathbf{g}) = \text{Inter}(\sigma_2(\Psi(\sigma_1(W_g * \mathbf{g} + W_x * \mathbf{x}))))$$

$$F(G(\mathbf{x}, \mathbf{g}), \mathbf{x}) = G(\mathbf{x}, \mathbf{g})\mathbf{x}$$

La [figure 1.18](#) synthétise la couche d’attention présentée ci-dessus.

Woo et al. [Woo et al., 2018] combinent les attentions spatiales et de canaux pour proposer un module d’attention par bloc convolutif (appelé *CBAM* pour *Convolutional Block Attention Module*). Leur module infère, de manière séquentielle, les cartes d’attention selon deux chemins : celui par les canaux et celui spatial. Tout d’abord la carte d’attention

est portée sur les canaux. Deux opérations sont faites sur le vecteur  $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$  : le calcul du maximum  $\mathbf{x}_{max}^c$  et de la moyenne  $\mathbf{x}_{moy}^c$  sur les dimensions spatiales ( $H$  et  $W$ ). Un premier vecteur  $s_c$  est calculé :

$$s_c = \sigma(W_2 * ReLU(W_1 * \mathbf{x}_{max}^c) + W_2 * ReLU(W_1 * \mathbf{x}_{moy}^c))$$

L'attention sur les canaux est donnée :

$$A_c(\mathbf{x}) = s_c \mathbf{x}$$

L'attention spatiale est faite de manière presque similaire. On effectue le calcul du maximum  $\mathbf{x}_{max}^s$  et de la moyenne  $\mathbf{x}_{moy}^s$  sur la dimension  $C$  et on applique une convolution suivie d'une fonction sigmoïde. L'attention spatiale est résumée par :

$$A_s(\mathbf{x}) = \sigma(W_3 * ([\mathbf{x}_{max}^s; \mathbf{x}_{moy}^s]))$$

La sortie du module d'attention est :

$$A_s(A_c(\mathbf{x}))$$

L'un des inconvénients de la couche proposée par Woo et al., mais aussi de Oktay et al., est que l'attention spatiale et l'attention basée sur les canaux sont calculées de manière indépendantes. Misra et al. [Misra et al., 2021] proposent une manière de prendre en compte cela à travers un module ajoutant peu de paramètres. L'idée est d'effectuer des rotations du vecteur d'entrée puis d'effectuer des convolutions de manières usuelles. Derrière cette approche, l'idée est de mettre l'accent sur la dépendance qu'il y a entre chaque dimension du vecteur d'entrée.

Notons  $Z_{pool}$  la fonction qui concatène la moyenne et la valeur maximale sur la première dimension du vecteur  $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ . Notons  $Perm_1$  et  $Perm_2$  la rotation de  $90^\circ$  dans le sens trigonométrique selon les axes respectifs  $H$  et  $W$ . Les rotations inverses sont dénotées  $Perm_1^{-1}$  et  $Perm_2^{-1}$ . De manière formelle la couche d'attention proposée par Misra et al. est :

$$\begin{aligned} \mathbf{x}_1 &= Perm_1(\mathbf{x}) \\ \mathbf{x}_2 &= Perm_2(\mathbf{x}) \\ \mathbf{s}_0 &= \sigma(W_0 * Z_{pool}(\mathbf{x})) \\ \mathbf{s}_1 &= \sigma(W_1 * Z_{pool}(\mathbf{x}_1)) \\ \mathbf{s}_2 &= \sigma(W_2 * Z_{pool}(\mathbf{x}_2)) \\ F(G(\mathbf{x}), \mathbf{x}) &= \frac{1}{3}(\mathbf{s}_0 \mathbf{x} + Perm_1^{-1}(\mathbf{s}_1 \mathbf{x}_1) + Perm_2^{-1}(\mathbf{s}_2 \mathbf{x}_2)) \end{aligned}$$

### 1.7.6 Exemple de réseaux de neurones

Dans cette sous-section, nous allons présenter des réseaux de neurones qui seront, par la suite, utilisés dans ce manuscrit. Il s'agit de l'architecture de base des réseaux, tout apport sera précisé en temps voulu. Une architecture typique des réseaux de convolution peut simplement se voir comme un réseau **encodeur** suivi d'un réseau **décodeur**.

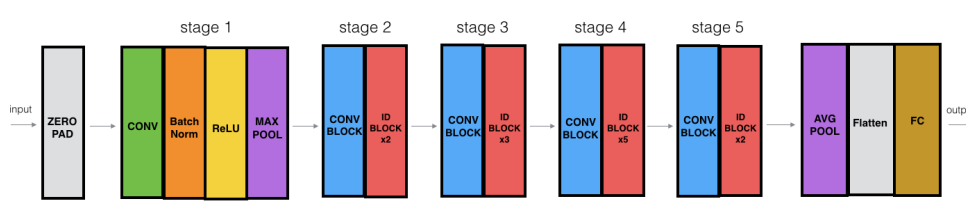


FIGURE 1.19 – Architecture du réseau ResNet-50 d’après [He et al., 2016].

## ResNet [He et al., 2016]

Le réseau de neurones résiduel (*Residual Network* en anglais) ou simplement ResNet, apparaît suite au problème de rétropropagation du gradient et l’augmentation de l’erreur d’apprentissage [He et al., 2016]. En effet, lorsque le réseau de neurones est trop profond, c’est-à-dire que le nombre de couche est grand, le gradient se réduit à zéro au fur et à mesure de la rétropropagation. La mise à jour des poids des différentes couches ne se fait presque plus, d’où la difficulté d’apprendre. Avec ce réseau, l’optimisation est assurée par des connexions résiduelles. Cela permet au gradient de passer dans deux couches de convolution, mais aussi de passer directement à travers un saut aux couches suivantes en reliant l’entrée de la couche  $n$  à la sortie de la couche  $(n + a)$ . ResNet ne segmente pas d’images, il s’agit un réseau **encodeur** (voir figure 1.19) dont la sortie est un vecteur. Néanmoins, il a été utilisé dans d’autres réseaux convolutifs.

## SegNet [Badrinarayanan et al., 2017]

SegNet suit l’architecture typique d’un réseau de convolution [Badrinarayanan et al., 2017]. En effet, il se compose d’un réseau **encodeur** suivi d’un réseau **décodeur** (voir figure 1.20).

La partie encodante est organisée en cinq parties, contenant un ensemble de blocs. Chaque bloc comprend une couche convolutive, une couche de normalisation (les cartes d’activations sont normalisées), une couche ReLU et une couche d’échantillonnage de taille  $2 \times 2$  et un stride de 2 (réduisant les dimensions par 2).

L’architecture du décodeur est organisée de la même manière, en cinq parties composées de blocs. Les couches sont similaires à celles des blocs de l’encodeur, sauf qu’ici chaque bloc comprend une couche de sur-échantillonnage (l’opération inverse du *pooling*). Les couches du décodeur sont connectées à celles de l’encodeur à l’aide de la méthode *pooling indices* où l’idée est de garder en mémoire la position lors de la couche d’échantillonnage *max pooling*. L’image de sortie est multicanaux dont chaque canal  $k$  représente la probabilité pour un pixel d’appartenir à la classe  $k$ .

SegNet est utilisé dans des segmentations multi-classes. Il a été développé dans le cadre d’analyse d’images de scène de route. Il est également très utilisé car le nombre de paramètres est moindre par rapport à d’autres réseaux grâce la couche d’échantillonnage proposée.

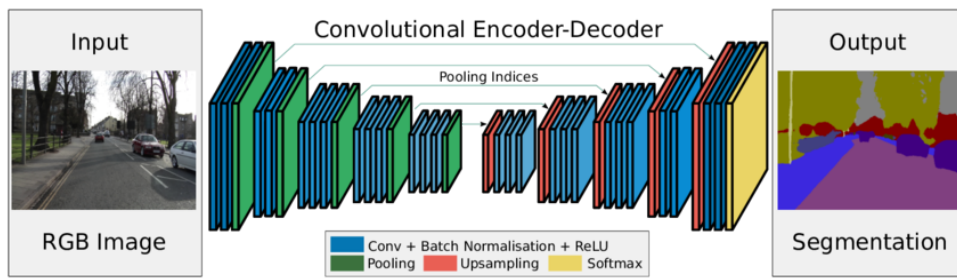


FIGURE 1.20 – Architecture du réseau SegNet d’après [Badrinarayanan et al., 2017].

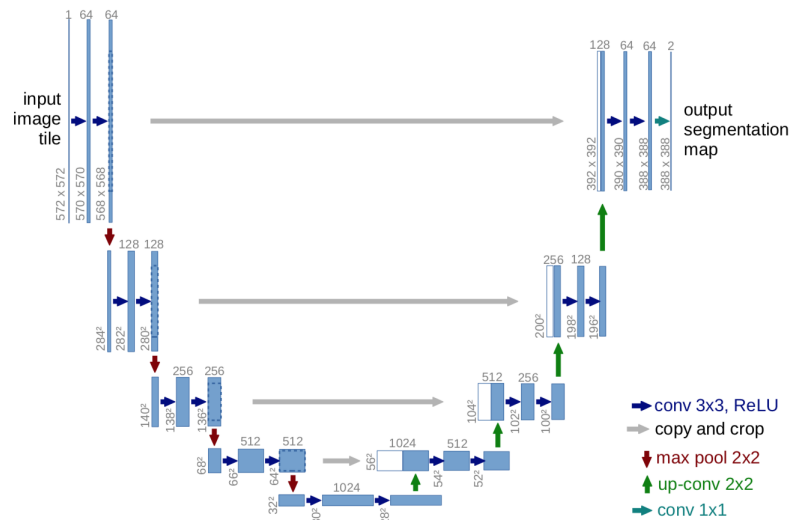


FIGURE 1.21 – Architecture du réseau U-Net, d’après [Ronneberger et al., 2015]

## Unet [Ronneberger et al., 2015]

Ronneberger et al. proposent un réseau baptisé U-Net [Ronneberger et al., 2015]. Son nom provient de l’aspect qu’il a lorsque l’on schématise le réseau. Il est construit de la même façon que SegNet : une partie encodante et une partie décodante composée de blocs qui se répètent. Cependant, dans U-Net la liaison entre les blocs encodants et les blocs décodants est la concaténation de ces blocs entre-eux (voir figure 1.21).

Le réseau encodeur est composé de deux couches de convolutions  $3 \times 3$  avec ReLU comme activation, puis d’une couche d’échantillonnage de taille  $2 \times 2$  et un stride de 2 (réduisant les dimensions par 2).

Le réseau décodeur est composé d’une couche de sur-échantillonnage, suivi d’une concaténation avec la sortie du bloc du réseau encodeur du même niveau, puis de deux couches de convolution de taille  $3 \times 3$ .

U-Net a été proposé pour la segmentation d’images médicales. Sa force réside dans sa faculté à apprendre sur des jeux de données réduits.

## Mask-RCNN [He et al., 2017]

Le réseau Mask R-CNN [He et al., 2017] est plus complexe que les précédents (voir figure 1.22). En effet, ce réseau a un double objectif, il doit à la fois segmenter et à la fois donner une sémantique, le tout en ayant plusieurs objets à traiter en simultanément. L'objectif était de proposer un réseau capable de faire ces deux opérations pour un panel considérable d'objets du quotidien (ballons, piétons, bancs, voitures, etc.). Il s'agit d'une extension du modèle Faster RCNN. Mais contrairement à Faster RCNN [Girshick, 2015], dont l'objectif est de classer sémantiquement les objets et de fournir la boîte englobante associée aux objets, Mask R-CNN va plus loin en proposant une segmentation.

Mask R-CNN se décompose en deux grandes parties (qui sont des réseaux de neurones). La première consiste à créer des régions où il pourrait y avoir un objet à détecter. Cette partie s'appelle *region proposal network* (voir la section sous-sous-section 1.7.5.5). La seconde partie est de classer chacune des régions détectées à la première étape, et proposer une segmentation pour ces régions. Les deux parties sont connectées à l'aide d'un troisième réseau, appelée *backbone structure*. Par exemple, ResNet peut-être utilisé en tant que *backbone*.

De part son apprentissage sur un large jeu de données, il est possible de le faire ré-apprendre très simplement sur un jeu de données spécifique.

## Refine Net [Lin et al., 2017]

Refine Net propose une approche multi-résolution, qui va employer à quatre reprises une architecture avec des unités d'affinement (*Refining units* en anglais) [Lin et al., 2017]. Chaque unité est connectée à un réseau résiduel (voir ResNet) et au bloc d'affinement antérieur, c'est-à-dire celui à une échelle plus grande (voir figure 1.23).

Chaque unité est composée de deux unités de convolution résiduelle (RCU) chacune

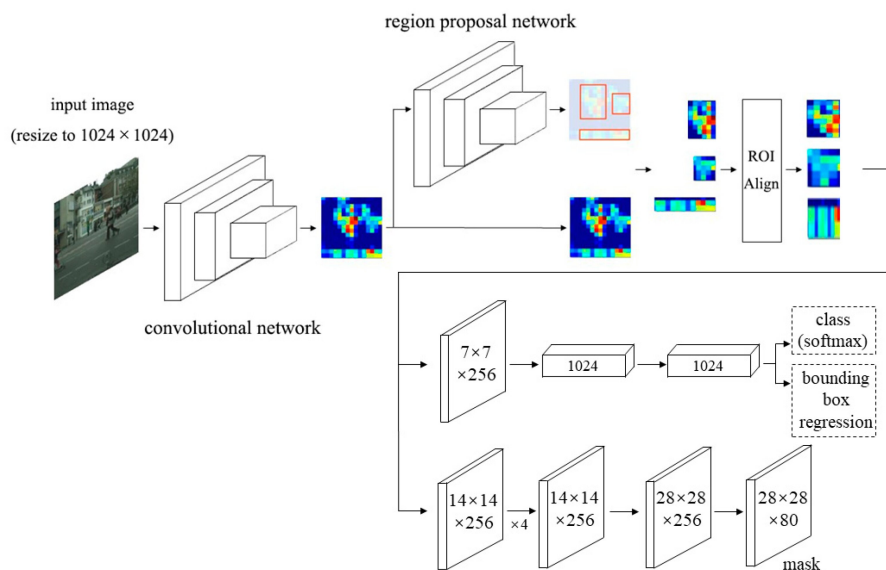


FIGURE 1.22 – Architecture du réseau Mask-RCNN d'après [He et al., 2017].

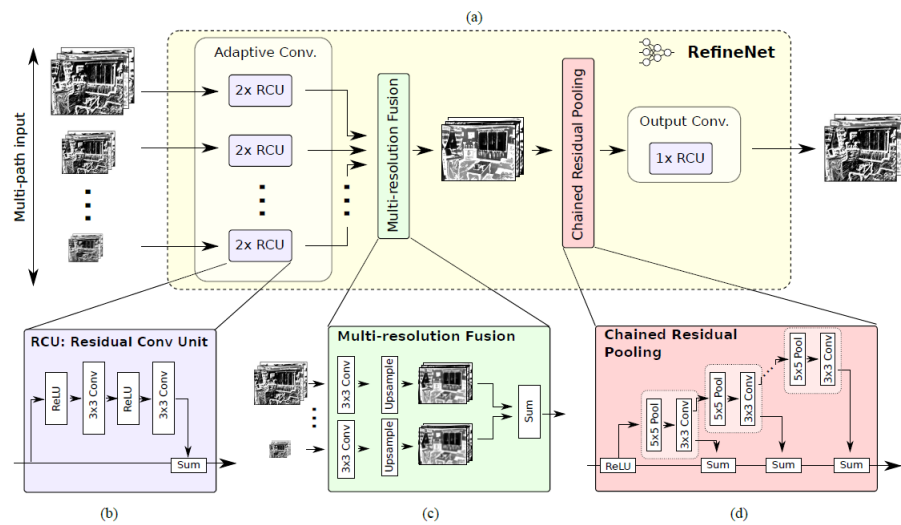


FIGURE 1.23 – Architecture du réseau RefineNet d’après [Lin et al., 2017].

incluant deux couches de convolution. Les sorties des RCU sont fusionnées dans un bloc de fusion multi-résolution composé d’une couche de convolution et d’une couche de sur-échantillonnage. Une chaîne de blocs d’échantillonnage de taille  $5 \times 5$  suivie d’une convolution  $3 \times 3$  est opérée sur les cartes caractéristiques de telle sorte qu’un bloc d’échantillonnage prenne la sortie du bloc précédent en entrée. Ce procédé permet de réutiliser les informations obtenues à l’étape d’avant évitant d’effectuer un sur-échantillonnage trop grand qui ferait perdre plus d’informations qu’autre chose. Enfin, les sorties de tous les blocs d’échantillonnage sont fusionnées avec les cartes de caractéristiques d’entrée par la sommation d’un RCU.

RefineNet a l’avantage de prendre en compte plusieurs résolutions de la même image. En effet, le principe est le même que celui des pyramides d’images utilisés en compression d’images ou en analyse d’images. Le réseau possède alors plus d’informations sur l’image permettant d’améliorer ses performances.

### Yolo V3 [Redmon and Farhadi, 2018]

Pour fini, Yolo V3 (*You Only Look Once* version 3) fait suite à la proposition de Redmon et al. en 2016 [Redmon et al., 2016] inspiré de Resnet [He et al., 2016]. Dans cet article de 2016, ils ont présenté une nouvelle approche pour la détection d’objets. L’extraction de caractéristiques et la localisation des objets ont été réunies en un seul bloc. Leur architecture, en une seule étape, appelée YOLO (*You Only Look Once*), permet un temps d’inférence très rapide. L’idée derrière YOLO est la suivante : il n’y a pas de *region proposal network* comme dans les réseaux RCNN (voir Mask-RCNN). Il s’agit essentiellement de convolutions tout au long du processus. Au lieu de découper les zones selon la probabilité de présence d’un objet puis de les transmettre à un autre réseau qui effectue la classification/segmentation, le réseau YOLO propose de le faire d’une seule traite.



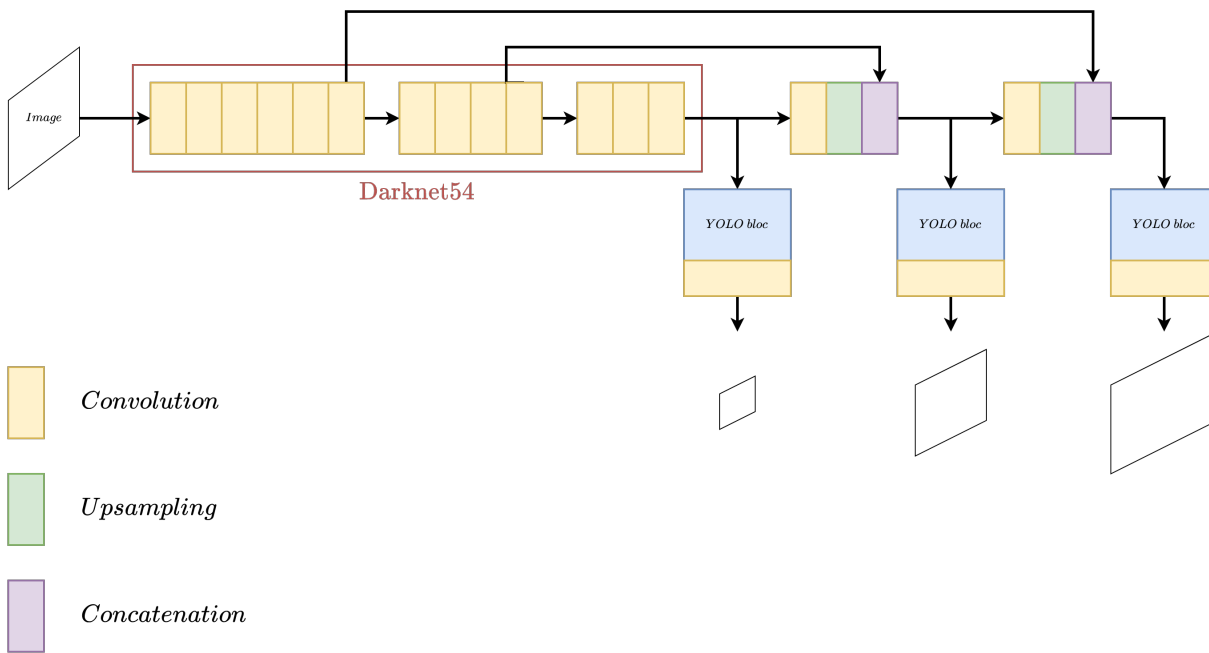


FIGURE 1.24 – Architecture du réseau YOLO V3.

La figure 1.24 montre l’architecture du réseau. Un réseau squelette est utilisé : Darknet54. Il procède des parties encodeur des réseaux auto-encodeurs vus précédemment. La seconde partie effectue la classification.

## 1.8 Résumé

Nous venons de décrire formellement ce qu’est une image, ce qu’est la segmentation. Nous avons pu définir, à partir d’une segmentation, la notion d’image labellisée. Puis nous avons évoqué les objectifs de la segmentation : mesurer des objets, donner une sémantique, etc. Cela nous a amené à parler des méthodes pour mesurer la qualité d’une segmentation, notamment à l’aide d’une vérité terrain donnée par des opérateurs humains.

Par la suite, nous avons abordé différentes techniques de segmentation qui seront utilisées dans les chapitres suivants. Nous avons commencé par les méthodes les plus simples reposant sur l’analyse de l’histogramme de l’image avec la méthode d’Otsu [Otsu, 1979, Yuan et al., 2015]. Puis, nous avons introduit la méthode de l’hystérésis utilisant deux seuils au lieu d’un [Medina-Carnicer et al., 2009, Khan et al., 2020]. S’en est suivi des approches par frontières en commençant par la méthode de Canny [Canny, 1986] après l’introduction du calcul du gradient [Sobel and Feldman, 1968, Kirsch, 1971, Hast, 2014]. Nous avons également introduit la méthode d’accroissement de régions ainsi que des méthodes pour subdiviser une image [Horowitz, 1974, Achanta et al., 2010, Wang et al., 2017]. Dans la suite, nous avons décrit la méthode des contours actifs [Kass et al., 1988, Chan and Vese, 2001b, Zhang et al., 2015]. Pour finir, nous avons détaillé les réseaux de neurones : le calcul au sein d’un neurone, les fonctions d’activations, l’apprentissage des poids du neurones en parlant de la fonction de coût (entropie croisée binaire et le

coefficient de corrélation de Matthews [Matthews, 1975, Abhishek and Hamarneh, 2021]). Nous avons aussi cité les méthodes pour effectuer l'apprentissage, avec la descente du gradient et ADAM [Kingma and Ba, 2014]. Nous avons évoqué des méthodes afin de régulariser le réseau avec la méthode du *dropout* ou de sa forme plus générale le *shakeout* [Vincent et al., 2008, Kang et al., 2018]. Nous avons détaillé des réseaux de neurones spécifiques au traitement d'images : les réseaux convolutifs. Finalement, nous avons donné quelques exemples de réseaux convolutifs ayant montré des résultats probants sur diverses applications [Ronneberger et al., 2015, He et al., 2016, He et al., 2017, Badrinarayanan et al., 2017].

La [tableau 1.2](#) résume les principaux avantages et les inconvénients des différentes méthodes abordées au cours de ce chapitre.

Méthode	Avantages	Inconvénients
Otsu	<ul style="list-style-type: none"> <li>● Rapide</li> <li>● Aucun paramétrage</li> </ul>	<ul style="list-style-type: none"> <li>● Très sensible au bruit</li> <li>● Uniquement bimodal</li> </ul>
Hystéresis	<ul style="list-style-type: none"> <li>● Rapide</li> <li>● Peu (ou aucun) paramétrage</li> </ul>	<ul style="list-style-type: none"> <li>● Sensible au bruit</li> <li>● Seuillage global</li> </ul>
Canny	<ul style="list-style-type: none"> <li>● Rapide</li> <li>● Adaptable</li> </ul>	<ul style="list-style-type: none"> <li>● Sensible (calcul du gradient)</li> <li>● Détection des contours</li> </ul>
Accroissement de régions	<ul style="list-style-type: none"> <li>● Robuste</li> <li>● Adaptable paramétrage</li> </ul>	<ul style="list-style-type: none"> <li>● Choix de la subdivision et de la fusion</li> <li>● Paramétrage difficile et manuel</li> </ul>
Contours actifs	<ul style="list-style-type: none"> <li>● Robuste</li> <li>● Adaptable paramétrage</li> </ul>	<ul style="list-style-type: none"> <li>● Choix des forces du contours</li> <li>● Paramétrage difficile et manuel</li> </ul>
Réseaux de neurones	<ul style="list-style-type: none"> <li>● Robuste</li> <li>● Adaptable</li> <li>● Rapide (à l'exécution)</li> </ul>	<ul style="list-style-type: none"> <li>● N'abouti pas nécessairement à la meilleure solution</li> <li>● Temps d'entraînement</li> </ul>

r

TABLE 1.2 – Résumé des méthodes vues au cours de ce chapitre avec leurs principaux avantages et inconvénients.

# Segmentation : les applications aux bois

## Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>64</b>
<b>2.2</b>	<b>L'anatomie du bois</b>	<b>64</b>
2.2.1	Les cernes	64
2.2.2	Orientation du bois	65
2.2.3	Fonctions du bois	66
2.2.4	L'aubier et le duramen	67
<b>2.3</b>	<b>Qualité du bois</b>	<b>68</b>
<b>2.4</b>	<b>Segmentation de la grume</b>	<b>68</b>
2.4.1	Schraml et Uhl (2014) [Schraml and Uhl, 2014]	69
2.4.2	Wimmer et al. (2021) [Wimmer et al., 2021]	72
<b>2.5</b>	<b>Segmentation du duramen</b>	<b>72</b>
2.5.1	Raatevaara et al. (2020) [Raatevaara et al., 2020]	72
<b>2.6</b>	<b>Détection des cernes annuels</b>	<b>74</b>
2.6.1	Conner et al. (1998) [Conner et al., 1998]	74
2.6.2	Soille et Misson (2001) [Soille and Misson, 2001]	75
2.6.3	Laggoune et al. (2005) [Laggoune et al., 2005]	76
2.6.4	Cerda et al. (2007) [Cerda et al., 2007]	78
2.6.5	Norell (2011) [Norell, 2011]	79
2.6.6	Zhou et al. (2012) [Zhou et al., 2012]	80
2.6.7	Sundari et al. (2014) [Sundari and Kumar, 2014]	81
2.6.8	Kennel et al. (2015) [Kennel et al., 2015]	83
2.6.9	Fabijańska et Danek (2018) [Fabijańska and Danek, 2018]	84
<b>2.7</b>	<b>Conclusion</b>	<b>86</b>

---

## 2.1 Introduction

Nous venons de voir différentes méthodes de segmentation. Dans ce chapitre, nous nous intéressons à leur utilisation dans le cadre d'images RVB de coupes transversales de grumes de bois.

Il conviendra en premier lieu de détailler l'anatomie du bois afin de présenter les éléments qui, dans le cadre du projet TreeTrace, nous intéressent. Les éléments sur lesquels nous porterons notre attention sont ceux qui permettent d'estimer la qualité du bois. Tous les éléments anatomiques qui seront abordés sont présentés dans la [section 2.2](#). Puis, nous aborderons dans la [section 2.4](#) les méthodes de segmentation de la grume en elle-même afin de l'isoler pour mieux analyser les autres éléments anatomiques. Ensuite, nous regarderons la segmentation du duramen dans la [section 2.5](#). Enfin, nous détaillerons les méthodes de segmentation des cernes de croissance dans la [section 2.6](#).

## 2.2 L'anatomie du bois

Le bois est un matériau renouvelable qui possède des propriétés favorables d'un point de vue écologique, mais également pour une utilisation structurelle. Du côté du consommateur, le bois est apprécié pour sa qualité visuelle, haptique ou olfactive. Toutefois, l'origine biologique du bois entraîne une variabilité considérable de ses propriétés, tant au niveau microscopique qu'au niveau macroscopique. Nous parlons ci-après des éléments anatomiques visibles sur une image de section de grumes, comme par exemple les cernes de croissances, l'aubier et le duramen, etc.

### 2.2.1 Les cernes

Chaque année, l'arbre croît en largeur formant une couche supplémentaire de bois nommée cerne annuel de croissance, anneau de croissance ou simplement **cerne**. La partie centrale de l'arbre, ou le centre structurel de l'arbre, est appelée la **moelle**.

Les cernes se distinguent dans les régions tempérées où le bois initial, formé au printemps (vers avril), est plus poreux que le bois final, formé en été et au début de l'automne (jusqu'en octobre). L'hiver marque l'arrêt de croissance et donc l'arrêt de la formation de nouvelles cellules. Cet arrêt est visible à l'échelle microscopique. Il est plus ou moins visible à l'œil sous la forme de limite pour le cerne annuel (voir [figure 2.1](#)). Dans les régions tropicales, des limites de cernes plus feutrées peuvent correspondre à des changements de saisons (l'absence de certains cernes est possible).

Il existe une relation entre la vitesse de croissance et la densité du bois chez les résineux. Les résineux à croissance rapide, c'est-à-dire à cernes larges, sont moins denses que les résineux à croissance lente (cernes étroits). Cette propriété découle du fait que le bois d'été est dense. Mais aussi que la largeur du bois d'été reste plus ou moins constante quelle que soit la vitesse de croissance, à quelques exceptions près (comme le douglas ou le mélèze). La croissance rapide entraîne une formation plus importante de bois de printemps moins dense. D'un point de vue commercial, les bois à croissance lente sont plus recherchés, car

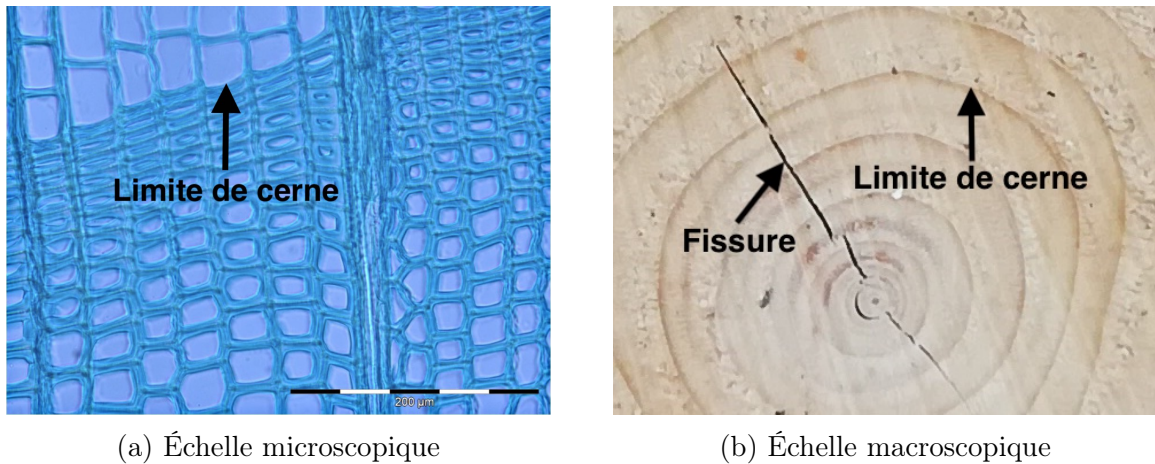


FIGURE 2.1 – Exemple de coupes transversales d'épicéa. La limite de cerne est visible à l'échelle microscope et macroscopique.

leurs propriétés mécaniques sont meilleures. Ils ont également une homogénéité visuelle intéressante pour un ébéniste ou un menuisier [Trouy and Triboulot, 2019].

### 2.2.2 Orientation du bois

Une coupe de **grume** de bois n'est pas la même selon comment elle est coupée. Dans le cadre de la thèse, nous nous intéressons aux coupes transversales. Il nous faut définir la notion de transversalité. La manière dont l'arbre croît permet de définir une orientation pour le bois. Les trois directions (voir **figure 2.2**) sont :

- longitudinale (**L**), dans l'axe du tronc ;
- radiale (**R**), perpendiculaire à **L** et passant par la moelle ;
- tangentielle (**T**), perpendiculaire à **R** et tangente aux cernes.

À partir de ces trois directions, il est possible de débiter une pièce de bois selon trois plans distincts (voir **figure 2.2**) :

- un plan transversal **RT**, selon l'axe radial et tangentiel, autrement dit perpendiculaire à l'axe de l'arbre ;
- un plan (longitudinal) radial **LR**, parallèle à l'axe et passant par la moelle ;
- un plan (longitudinal) tangentiel **LT**, tangent aux cernes ;

**Dans la suite du manuscrit nous ne nous intéresserons qu'aux coupes transversales.**

Le bois est anisotrope, c'est-à-dire que ses propriétés physiques ne sont pas les mêmes selon ces trois directions. Par exemple, la résistance mécanique du bois est beaucoup plus forte dans la direction **L** que dans les directions **R** et **T**. Le bois vert ayant des dimensions maximales, ces changements de dimensions sont appelés retrait du bois. Ainsi, le retrait n'est pas le même dans les trois directions entraînant la formation de fissures

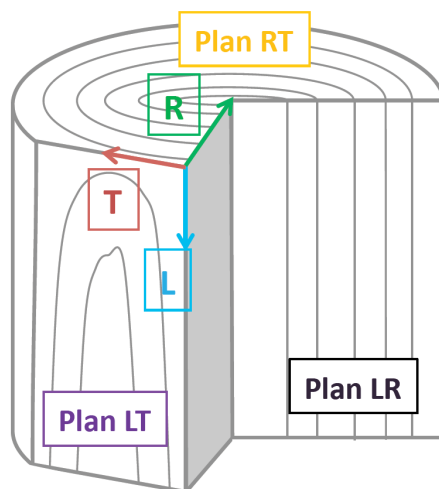


FIGURE 2.2 – Les trois directions et plans possibles dans le bois, d’après [George et al., 2020].

(ou de roulures) lors du séchage de la grume. La [figure 2.1](#) montre un exemple de fissure visible le long de l’axe radial à l’échelle macroscopique. Le retrait est une propriété très anisotrope, le retrait axial est souvent négligé par rapport au retrait tangentiel et au retrait radial [Trouy and Triboulot, 2019]. Afin d’éviter que le bois ne subisse des retraits après mise en œuvre, qui entraînerait des problèmes notamment dans les assemblages, il est indispensable de le sécher avant tout usinage. En plus d’être anisotrope, le matériau bois est aussi hétérogène. C’est-à-dire que ses propriétés physiques ne sont pas les mêmes en tout point.

### 2.2.3 Fonctions du bois

Les propriétés physiques du bois déterminent en partie sa qualité. Or, ces dernières sont liées aux fonctions du bois dans l’arbre. Dans un arbre, le bois a plusieurs fonctions dont les plus importantes sont :

1. la conduction de la sève brute. L’eau et les minéraux puisés dans le sol par les racines sont acheminés jusqu’aux feuilles ;
2. le soutien mécanique ;
3. le stockage des réserves et l’élaboration d’éléments chimiques nécessaires aux cellules.

Chez les résineux, les fonctions de conduction et de soutien mécanique sont assurées par les trachéides, des cellules longitudinales peu spécialisées. Chez les feuillus, des cellules longitudinales plus grosses que les autres sont spécialisées dans la conduction : les vaisseaux. Les fibres, des cellules longitudinales plus étroites, jouent le rôle de soutien mécanique. Enfin, des cellules longitudinales de taille intermédiaire appelées parenchyme assurent les fonctions de stockage ainsi que l’élaboration des besoins vitaux des cellules. Lorsque les cellules de parenchyme s’amassent de façon radiale, elles forment un rayon ligneux. Ces rayons sont présents chez les feuillus et dans une moindre mesure chez les résineux.

### 2.2.4 L'aubier et le duramen

L'aubier est défini comme la partie du tronc dans laquelle les cellules de parenchyme sont encore fonctionnelles<sup>7</sup>. Autrement dit, elles stockent les réserves sous forme d'amidon. Elles sont aussi capables de produire des substances chimiques de défense face aux insectes et/ou aux champignons.

Le duramen, ou bois parfait, est constitué de cellules de parenchyme mortes et non-fonctionnelles. Chez un certain nombre d'espèces, les cellules de parenchyme avant de mourir produisent des substances qui vont imprégner le bois et le protéger chimiquement. Ce processus est la duraminisation. Chez certaines espèces, comme le cas du Douglas, il arrive que le duramen apparaisse coloré à nos yeux selon sa composition chimique. Ainsi, l'aubier peut se différencier, à l'œil, du duramen. On parle alors de bois à aubier différencié (voir la [figure 2.3](#)). C'est cette caractéristique qui sera largement utilisée par la suite pour segmenter le duramen. Les bois dont le duramen ne confère pas de différenciation, en terme de couleur, sont appelés bois blancs comme par exemple le peuplier. L'imprégnation de substances dans le duramen confère à cette partie du bois d'autres propriétés variables : l'odeur, la durabilité naturelle vis-à-vis des champignons et des insectes, une stabilité dimensionnelle, etc.

La duraminisation explique en grande partie certaines caractéristiques importantes, voire essentielles des essences, à savoir la durabilité naturelle et l'imprégnabilité [[Trouy and Triboulot, 2019](#)]. L'imprégnabilité se définit comme la faculté d'un bois à être pénétré par un liquide (comme par exemple des produits de préservation).

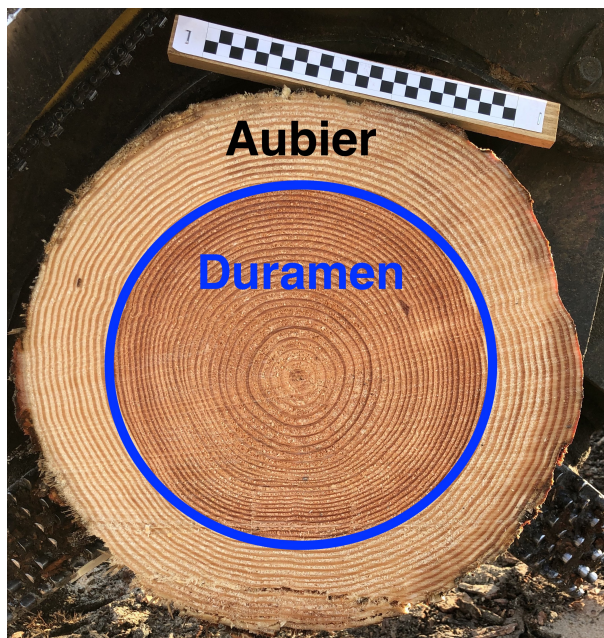


FIGURE 2.3 – Exemple montrant la limite entre le duramen et l'aubier sur le Douglas, un bois à aubier différencié. Sur cette essence, le duramen prend une couleur rouge/rose.

7. d'après la Norme NF B 50-002 - Bois - Vocabulaire



## 2.3 Qualité du bois

Nous venons de voir différentes caractéristiques anatomiques du bois. Il est nécessaire maintenant de les relier à la qualité du bois. Le bois présente une grande variabilité dans ses propriétés. Elle est expliquée par la génétique, la sylviculture, le sol, le climat, etc. Le bois est le résultat de facteurs liés à son anatomie (aubier, duramen, largeur de cernes, excentricité, ...). C'est également le fruit des traitements qui lui sont appliqués (séchage, ...). Cette variabilité n'est pas uniquement interspécifique (entre essences), elle est également intraspécifique (variabilité d'un arbre à l'autre de la même essence) et elle l'est aussi dans un même arbre.

Le classement des bois de construction se fait très souvent à partir d'un examen visuel dans lequel on va chercher des particularités (des nœuds, des poches de résine, ...). Deux classements sont possibles : le classement de l'aspect sur lequel l'esthétisme est pris en compte et le classement structurel.

Le premier classement se limite aux usages non-travaillants (comme le bardage). Les critères de classement diffèrent d'une essence à l'autre et font l'objet de différentes normes.

Le deuxième est le classement structurel. Le bois de construction est réparti dans des classes de résistances définies par la norme EN338<sup>8</sup>. Des classes sont définies à partir d'essai mécanique (flexion, traction). La classe peut être déterminée visuellement selon les critères définis par la norme NF B 52-001<sup>9</sup>. Comme pour le classement d'aspect, le classement visuel pour l'emploi en structure du bois varie selon l'essence, mais des critères sont communs : largeur des cernes, caractéristiques liées aux nœuds (diamètres), présence de poche de résine ou de fente, etc. Par exemple, un bois rond de douglas ne peut être en classe A (meilleure classe) que si le taux de croissance, c'est-à-dire le rayon moyen divisé par le nombre de cernes, est inférieur à 4 mm. Ce classement servira de base pour la recherche dans l'image des éléments pertinents à l'estimation de la qualité.

À noter qu'il est possible d'évaluer la qualité à partir d'une machine : par flexion, par ultrasons, par vibrations, par rayonnements par rayons X, etc. Cette dernière est appelée tomographie par ordinateur à rayons X. Dans le cadre du projet, il s'agit de savoir s'il est possible d'estimer la qualité à partir d'une image RVB d'une coupe transversale.

## 2.4 Segmentation de la grume

Nous venons de voir les éléments anatomiques visibles sur une coupe transversale. Afin d'identifier précisément les éléments nécessaires à l'estimation de la qualité du bois, il faut supprimer tous les éléments superflus de l'image. Autrement dit, il convient de segmenter la grume pour faciliter l'analyse des éléments pertinents pour l'estimation de la qualité.

Les deux seules études, à notre connaissance, à s'intéresser à la segmentation de coupes transversales sont celles proposées par Schraml et Uhl [Schraml and Uhl, 2014] et celle de Wimmer et al. [Wimmer et al., 2021].

---

8. <https://www.boutique.afnor.org/fr-fr/norme/nf-en-338/bois-de-structure-classes-de-resistance/fa170486/57555>

9. <https://www.boutique.afnor.org/fr-fr/norme/nf-b520011/regles-dutilisation-du-bois-dans-la-construction-classement-visuel-pour-lem/fa190902/80578>



FIGURE 2.4 – Exemple des images utilisées dans l'article [Schraml and Uhl, 2014].

### 2.4.1 Schraml et Uhl (2014) [Schraml and Uhl, 2014]

Jusqu'à présent, les images étaient segmentées manuellement ou bien, il n'y avait pas la nécessité d'effectuer une segmentation. En effet, l'utilisation d'images tomodynamétriques est plus fréquente pour l'étude du bois et ne requiert pas un tel procédé, car le fond est noir à l'acquisition de l'image.

Les images utilisées sont de taille  $1024 \times 768$  pixels (jeu de données présenté dans l'article [Schraml and Uhl, 2013], voir figure 2.4). Une conversion en niveaux de gris est effectuée en utilisant un poids identique pour chaque composante couleur.

Schraml et Uhl utilisent l'accroissement de régions combiné à une analyse de texture. En effet, les images RVB de coupes transversales présentent une texture principalement due au sciage. Nous allons détailler la méthode utilisée pour mesurer la texture d'une image. L'image est divisée en blocs. La fusion des blocs se fait par une analyse de texture, notamment avec l'utilisation du motif binaire local. Un post-traitement est fait pour lisser et affiner le contour obtenu par l'accroissement de régions.

#### Motif binaire local - Local Binary Pattern (LBP)

Il existe plusieurs façons pour mesurer la texture d'une image. La mesure que les auteurs ont utilisée est le motif binaire local (**Local Binary Patterns (LBP)** en anglais). Les LBP permettent d'extraire des descripteurs de texture spatiaux pour les images en niveaux de gris.

Soit  $\mathbf{p} = (i, j)$  un pixel de l'image, un voisinage de  $Q \in \mathbb{N}$  points est défini à partir d'un cercle de rayon  $R > 0$  et de centre  $\mathbf{p}$ . L'ensemble des voisins est donné par :

$$\forall q \in \llbracket 0, \dots, Q-1 \rrbracket, \quad v_q = I\left(i - R \sin\left(\frac{2\pi q}{Q}\right), j + R \cos\left(\frac{2\pi q}{Q}\right)\right)$$

La nouvelle valeur en  $\mathbf{p}$  par le calcul du motif binaire local est :

$$I'(\mathbf{p}) = \sum_{q=0}^{Q-1} 2^q \delta(v_q - I(\mathbf{p}))$$

où  $\delta$  est la fonction Heaviside (1 si la valeur est supérieure ou égale à 0, sinon 0).

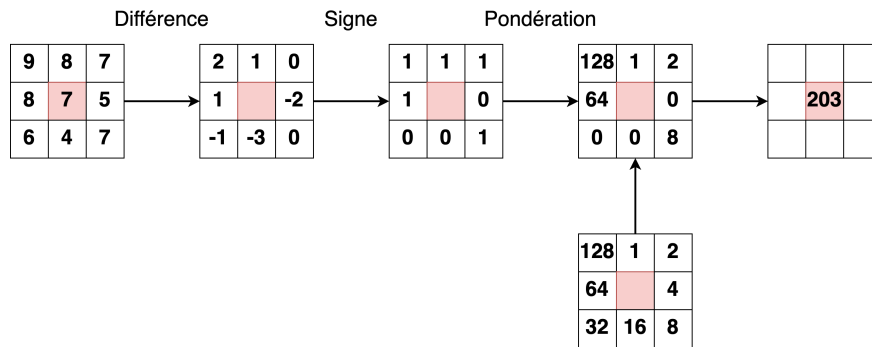


FIGURE 2.5 – Schéma du principe du calcul du motif binaire local (avec  $Q = 8$  et  $R = 1.0$ )

En appliquant ce principe à tous les pixels de l’image on obtient une image de texture. Dans l’article, le principe est appliqué aux blocs dont l’histogramme sera comparé avec celui des blocs voisins.

### Distance d’histogramme

Il est possible de calculer une distance entre deux histogrammes. La méthode la plus simple consiste à calculer la distance ( $L_1$ ,  $L_2$ , distance de Hellinger, etc.) valeur par valeur entre les histogrammes. Une autre mesure entre deux histogrammes est la distance de Wasserstein (**Earth Mover’s Distance (EMD)** en anglais). Cette distance calcule le coût minimum qu’il faut pour passer d’une *signature*  $P$  à une *signature*  $Q$ .

Dans le cas où  $P$  et  $Q$  sont des histogrammes, cette distance revient à calculer la norme  $L_1$  entre leur fonction de distribution cumulative.

La **figure 2.6** montre le calcul du motif binaire local, puis de l’histogramme et de la mesure de distance pour deux images de cernes de bois. La distance est de 1,352, autrement dit ces deux images sont très proches d’un point de vue de la texture.

### Détails de la méthode

Pour l’analyse de texture, les auteurs ont considéré le vecteur descripteur suivant : la valeur moyenne des niveaux de gris rapportée à la variance, l’entropie rapportée à la variance ainsi que l’histogramme d’intensité ou le motif binaire local. Plusieurs décompositions de l’image sont proposées : avec ou sans superposition des blocs.

Pour l’initialisation de la région, la localisation de la **moelle** est nécessaire. Il est fait par l’algorithme proposé dans [Schraml and Uhl, 2013]. Puis, quatre blocs proches de la moelle sont sélectionnés, auxquels sont ajoutés les 4 voisins connexes respectifs, formant la région initiale. Chaque bloc voisin de la région, qui n’est pas déjà dedans, est analysé et fusionné si possible. Les seuils de similarité se basent sur la variance des descripteurs auxquels sont ajoutés des paramètres pour mieux généraliser la méthode.

Un post-traitement est effectué afin de lisser la frontière de la région obtenue. D’abord, les blocs au bord sont gardés (ce sont ceux ayant au moins 1 voisin qui n’appartient pas la région). Puis, un cercle est ajusté par rapport aux blocs restants. Ceux à l’intérieur du cercle sont supprimés. Puis, une ellipse est ajustée. Les blocs à l’extérieur sont supprimés.

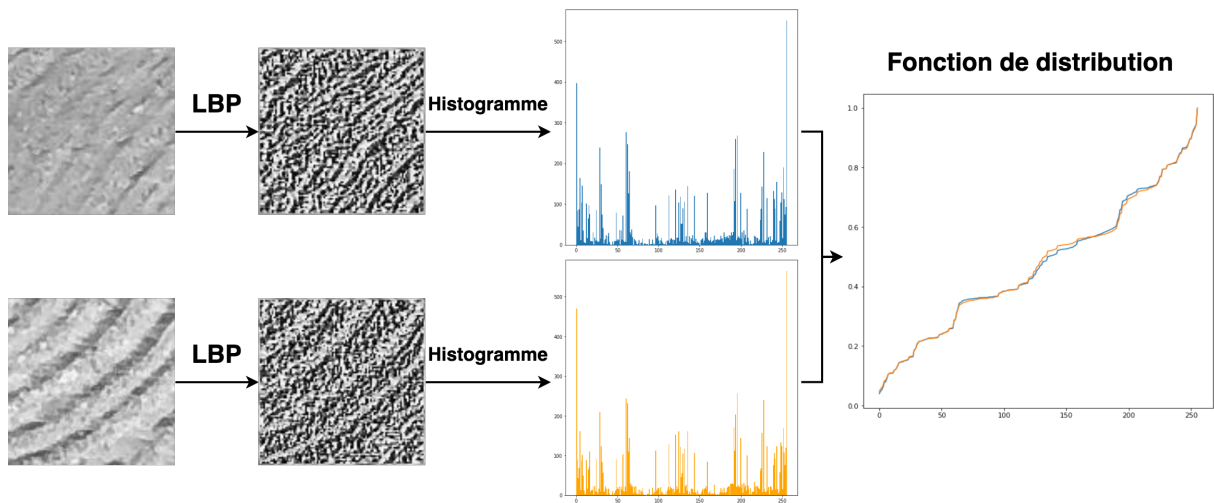


FIGURE 2.6 – Exemple de la méthode sur deux blocs de taille  $100 \times 100$  issus d’une image de coupe transversale. La distance entre ces deux blocs est de 1,352.

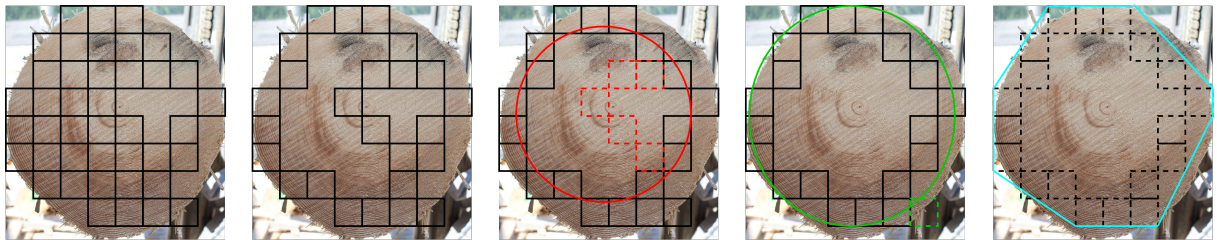


FIGURE 2.7 – Post-traitement proposé par [Schraml and Uhl, 2014]. De gauche à droite : l’image avec les blocs appartenant à la région finale, les blocs au bord, ajustement du cercle (en rouge) et les blocs retirés (blocs rouges), ajustement de l’ellipse (en vert) et les blocs retirés (blocs verts) et approximation par une forme  $\alpha$  (en bleu).

La dernière étape est une approximation par une forme  $\alpha$  de la frontière des blocs restants formant la frontière de la segmentation. La forme  $\alpha$  est équivalente à l’enveloppe convexe si  $\alpha = 0$  sinon la forme tend vers l’arbre de couverture minimale quand  $\alpha \rightarrow \infty$ . La figure 2.7 illustre le post-traitement proposé.

L’absence de post-traitement induit des résultats de faible qualité car un bon nombre de blocs à l’intérieur de la grume ne sont pas fusionnés. L’ajustement par un cercle et une ellipse permet de rectifier ce défaut. Enfin, la forme  $\alpha$  permet de lisser le contour, mais omet toute considération des propriétés de l’image.

Pour mesurer la précision de leur méthode, les auteurs utilisent une valeur issue de la matrice de confusion (voir sous-section 1.2.3) :

$$Err = 1 - \frac{TP + TN}{FN + FP + TN + TP}$$

Ils obtiennent une erreur entre 9% et 5% (avec des blocs de taille  $16 \times 16$  pixels et avec l’histogramme d’intensité). Le motif binaire local n’améliore pas les résultats. Au niveau

temps de calcul, l'algorithme semble quasi-linéaire en la taille de l'image avec des blocs de  $16 \times 16$  pixels mais semble quadratique (en la taille de l'image) avec des blocs de  $32 \times 32$  pixels.

Un problème de ce calcul est le déséquilibre des classes. En effet, dans leur jeu de données la zone couverte par la grume dans la vérité terrain (TP) est beaucoup plus grande que le reste (TN). Ainsi, il est possible qu'avec une image toute blanche, on ait une erreur faible.

### 2.4.2 Wimmer et al. (2021) [Wimmer et al., 2021]

Dans le cadre de la reconnaissance d'une grume à partir d'images RVB de coupe transversale, Wimmer et al. [Wimmer et al., 2021] proposent une segmentation à partir d'un réseau de neurones. Les auteurs ont recours à l'utilisation du réseau de neurones Mask R-CNN (voir la [section 1.7](#) et la [figure 1.22](#)). L'architecture squelette utilisée est le réseau ResNet-50 qui a été pré-entraîné sur le jeu de données COCO dataset. L'entraînement pour la segmentation de grume a été fait par une cross-validation en 4 groupes (un groupe est composé d'environ 279 images) en considérant 30 époques sur le jeu de données de [Schraml and Uhl, 2013]. Une époque correspond à l'apprentissage fait sur l'ensemble du jeu d'apprentissage. Ensuite, le réseau a été appliqué sur un autre jeu de données [3](#). Les images sont seuillées avec une valeur de seuil de 0.5 ou 0.25 selon le jeu de données.

## 2.5 Segmentation du duramen

La segmentation de la grume permet de simplifier la détection d'autres éléments anatomiques. La quantité de duramen est un indicateur de la qualité du bois, car la duramisation change les propriétés physiques et chimiques de cette partie du bois. Dans cette section, nous allons aborder les solutions proposées par la littérature pour effectuer la segmentation du duramen.

### 2.5.1 Raatevaara et al. (2020) [Raatevaara et al., 2020]

En 2020, Raatevaara et al. proposent une méthode pour segmenter le duramen. Tout comme pour la segmentation de la grume précédemment décrite, les auteurs utilisent dans un premier temps l'accroissement de région pour avoir une première segmentation qui sera affinée par un post-traitement.

Les images ont été capturées à l'aide d'un APN (Nikon D7200) muni d'un objectif, mais les dimensions de l'image ne sont pas fournies. Le constructeur précise que la taille des images peuvent aller de  $2400 \times 1344$  pixels à  $6000 \times 3368$  pixels (laissant une large possibilité de dimensions pour les images utilisées).

Une segmentation manuelle de la grume est faite (via le logiciel GIMP, voir [figure 2.8](#)) et la position de la moelle est calculée. La méthode utilisée pour estimer la position de la moelle est proche de celle proposée par Schraml et al. [Schraml and Uhl, 2013] (voir [chapitre 4](#)). L'orientation des cernes est calculée à travers une fenêtre glissante, de taille



FIGURE 2.8 – Exemple d’une image utilisée dans l’article [Raatevaara et al., 2020]. En (a) l’image d’origine, en (b) la segmentation manuelle de la grume, en (c) la vérité terrain (en blanc le duramen, en gris l’aubier).

180 × 180 pixels, à l’aide d’une analyse par composantes principales sur la transformée de Fourier. La moelle est la moyenne des intersections des directions.

L’image est convertie en niveaux de gris en prenant la moyenne entre la composante S de l’espace HSV et la composante  $a^*$  de l’espace CieLAB. Enfin l’image est réduite à une taille de 1500 × 1000 pixels.

Une approche par accroissement de régions est proposée pour segmenter le duramen. Les auteurs supposent que proche de la moelle le duramen est homogène. Plusieurs blocs sont choisis (9 blocs) autour de la moelle dans lesquels des pixels vont être sélectionnés pour servir de départ à l’accroissement de la région. L’accroissement se fait à partir de la moyenne  $\mu$  et de l’écart-type  $\sigma$  des niveaux de gris de la région. Un pixel  $\mathbf{p}$  appartient à la région si :

$$\mu - a\sigma < I(\mathbf{p}) < \mu + a\sigma$$

où  $a$  est un paramètre.

L’accroissement est répété plusieurs fois. Les pixels initiaux diffèrent à chaque itération. L’accumulation des accroissement forme une carte de probabilités. L’accumulation est seuillée par un calcul de maximum d’entropie. Enfin, un post-traitement est effectué à l’aide d’outils morphologiques (dilatation, érosion et fermeture) afin de lisser les contours de la segmentation.

Un deuxième post-traitement est fait afin d’améliorer la segmentation. En partant de la segmentation précédente, une zone d’incertitude (jusqu’à une distance de 50 pixels des pixels segmentés) est définie formant une *trimap* (voir figure 2.9). Le traitement fait à partir de cette carte est l’algorithme de *KNN-matting* [Chen et al., 2013]. Le résultat est une nouvelle segmentation qui est également lissée avec la même méthode que pour la première segmentation.

Pour estimer la qualité de leur segmentation, les auteurs ont utilisés la mesure du diamètre du duramen. Les auteurs ont également utilisé l’indice de Jaccard, ou l’intersection sur l’union (IoU), pour mesurer la qualité des segmentations. L’indice est calculé par la formule :

$$Jaccard = IoU = \frac{TP}{TP + FP + FN}$$

Les résultats donnent un indice supérieur à 0.8 pour plus de 60 images sur les 86 acquises. Les cas où la méthode ne semble pas probante sont ceux où le duramen est très peu visible de manière colorimétrique sur les images.

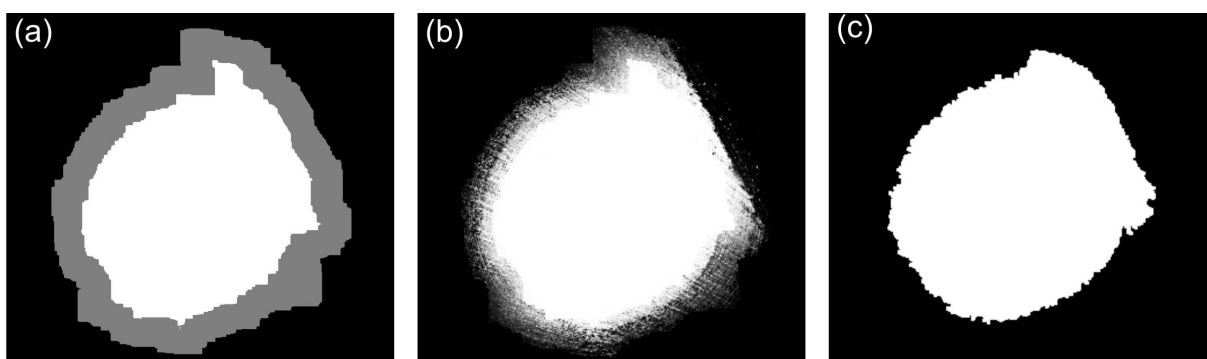


FIGURE 2.9 – Exemple d’une image utilisée dans l’article [Raatevaara et al., 2020]. En (a) la *trimap* obtenue à partir de la première segmentation, en (b) le résultat du *KNN-Matting*, en (c) la segmentation finale.

## 2.6 Détection des cernes annuels

Un autre élément important pour estimer la qualité d’une grume est l’analyse des cernes de croissance. La largeur des cernes et ses variations indiquent comment l’arbre a poussé et restent importantes pour évaluer la qualité du bois. Pour les analyser, il est nécessaire de les mettre en évidence. Nous allons, par la suite, détailler dans l’ordre chronologique les méthodes qui ont été proposées pour détecter les cernes sur des images RVB de coupes transversales de grumes de bois.

### 2.6.1 Conner et al. (1998) [Conner et al., 1998]

L’une des premières études à analyser les cernes est celle de Conner et al. [Conner et al., 1998]. L’étude est menée dans le contexte de la *dendrochronologie*.

Une grande précision dans la largeur des cernes est importante, c’est pourquoi les images traitées sont un assemblage de fenêtres glissantes (avec superposition) partant de la *moelle* vers l’écorce. L’acquisition est faite à partir d’une caméra montée sur un microscope et le déplacement est contrôlé par station de travail Sun Ultra 1. Lors de l’acquisition des images, une normalisation est effectuée afin de s’assurer que la superposition n’induisse pas de changements importants d’intensité.

La méthode proposée repose sur une adaptation du filtre de Canny. La première étape pour détecter les cernes est d’estimer l’orientation des cernes (i.e l’orientation des tangentes aux cernes). Les cernes sont supposés être parallèles les uns aux autres (à l’échelle de fenêtre glissante). Si la direction d’une fenêtre glissante est connue, alors l’orientation est connue. Sinon, le calcul de l’orientation est fait à partir du gradient avec les filtres de Robert ou de Sobel. Puis, l’orientation est quantifiée en 8 directions. Pour chaque direction, la moyenne de la norme du gradient est calculée. L’orientation ayant la moyenne la plus élevée est considérée comme étant la direction des cernes.

L’étape suivante est de supprimer les non maxima de la norme du gradient. La norme du gradient, dans un voisinage de 8-connexe, est étudiée selon l’orientation quantifiée. Si la norme n’est pas plus grande que ses voisins, alors le pixel n’est pas une limite de cerne. Comme le déplacement de la fenêtre glissante est connu (allant de la moelle vers l’écorce),

il est possible de supprimer d'avantage de points car le changement de cerne se fait d'une faible intensité vers une forte intensité. Les maxima qui ne correspondent pas à ce type changement sont aussi supprimés.

Un traitement a posteriori est effectué afin d'avoir des cernes connexes. La première étape consiste à créer l'image labellisée. La technique pour obtenir des cernes entièrement connectés repose sur un seuillage par hystérésis. Le seuil haut permet de détecter les cernes valides. Le seuil bas permet de reconnecter les cernes obtenus avec ceux obtenus par le premier seuil.

### 2.6.2 Soille et Misson (2001) [Soille and Misson, 2001]

En 2001, Soille et Misson proposent une méthode de détection des cernes basée sur la morphologie mathématique [Soille and Misson, 2001].

Les images traitées sont en niveaux de gris. Pour éviter les craquelures, le traitement suivant a été appliqué : les rondelles ont été plongées dans du polyéthylène glycol (PEG 1000) pendant 5 jours. Puis, elles ont été mises à sécher à l'air pendant 5 jours. Ensuite, elles ont été poncées à la ponceuse circulaire avec différents grains : 80, 120, 200 et 400. Enfin, elles ont été scannées avec une résolution de 600 dpi. Les déformations possibles dues à l'acquisition scanner ont été corrigées. Les images finales en niveaux de gris sont d'une taille de  $7625 \times 7531$  pixels. Les images obtenues sont de haute résolution sur lesquelles les limites de cernes sont beaucoup plus marquées. La [figure 2.10](#) montre un exemple de type d'image traitée (non à l'échelle).

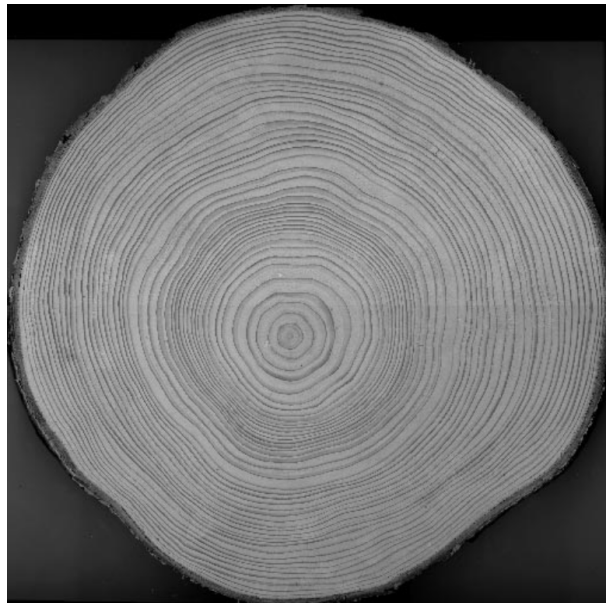


FIGURE 2.10 – Image issue de l'article [Soille and Misson, 2001].

Pour délimiter les cernes, la méthode est la suivante :

- une ouverture morphologique est appliquée (l'élément structurant est un carré de taille 3) ;



- une fermeture morphologique est appliquée (l'élément structurant est un carré de taille 3) ;
- une seconde ouverture est faite par un carré de taille 15 ;
- l'opérateur black-top hat (i.e. différence entre l'image obtenu par fermeture et l'image elle-même) est appliqué avec un élément structurant carré de taille 67 pixels ;
- un seuillage pour extraire les zéros. En pratique une valeur de 5% de la dynamique (différence entre la valeur maximale et minimale) de l'image a été choisie ;

Cette image seuillée sert de marqueur pour le watershed [Vincent and Soille, 1991]. En plus, le gradient, obtenu par différence entre le dilaté et l'érodé de l'image (par un élément structurant carré de taille 3), sert d'image d'entrée pour le watershed.

Cependant, il arrive que la segmentation obtenue ne soit pas parfaite. En effet, la présence de noeuds ou autres défauts peuvent induire une mauvaise segmentation. Un post-traitement est fait pour réduire les erreurs de cette première segmentation. L'idée est de labelliser les classes de la segmentation par rapport à leur distance à la moelle et de fusionner les régions avec le même label. En effet, comme chaque cerne décrit une année de l'arbre, toutes les classes ayant le même âge, i.e. le même label, doivent appartenir au même cerne de croissance. La fonction temporelle géodésique est utilisée pour labelliser [Soille, 1994]. La figure 2.11 résume la méthode et illustre les résultats obtenus pour chaque étape (à l'exception de la fonction géodésique) sur une sous-image issu d'un de nos jeux de données.

### 2.6.3 Laggoune et al. (2005) [Laggoune et al., 2005]

En 2005, Laggoune et al. proposent une méthode basée sur des filtres afin de délimiter les cernes [Laggoune et al., 2005]. Les auteurs ne donnent pas de détails concernant les images utilisées. Néanmoins, elles semblent similaires à celles utilisées par [Soille and Misson, 2001].

Le filtre est un filtre semi-automatique adapté aux images bruitées et floues (mais sans présence de craquelure ou noeuds dans les coupes transversales). Le contour d'une image peut s'estimer par le calcul du Laplacien d'une fonction  $f$  de détection de contours [Sarifuddin et al., 2005]. La fonction  $f$  proposée est :

$$\forall x \in \mathbb{R}, \quad f(x) = k(\lambda + \sin(\beta x) \exp(-\alpha x))$$

où  $k$ ,  $\lambda$ ,  $\alpha$  et  $\beta$  sont des paramètres.

La fonction se simplifie en ajoutant des contraintes aux limites et en 0. Dans le cadre d'une image en deux dimensions, les auteurs constatent que leur filtre est proche du filtre Gaussien. Mais que la formulation du filtre Gaussien est plus avantageuse en terme de calcul. Ainsi, ils appliquent la dérivée seconde d'un filtre Gaussien 2D pour la détection des contours (i.e. la limite des cernes). La norme de la dérivée seconde est l'image résultante.

Une fois le calcul fait, l'image résultante est seuillée. Le seuillage choisi par les auteurs est un seuillage par hystérisis (similaire à celui proposé par Conner [Conner et al., 1998]) afin d'obtenir des cernes 8-connexes et épais d'un pixel.

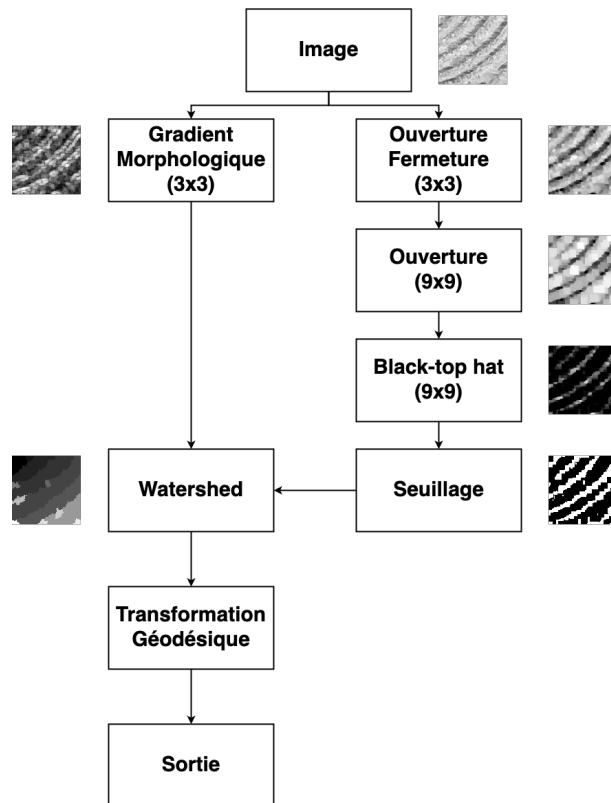


FIGURE 2.11 – Résumé de la méthode de [Soille and Misson, 2001] pour la détection des cernes.

### 2.6.4 Cerda et al. (2007) [Cerda et al., 2007]

Cerda et al. proposent en 2007 une nouvelle méthode pour détecter les cernes sur des images RVB [Cerda et al., 2007].

Le jeu de données est composé de 10 images de coupes transversales âgées d'une dizaine d'années. Les images ont une taille de  $700 \times 700$  pixels et ont été acquises par une caméra Nikon Coolpix 885 (voir figure 2.12).

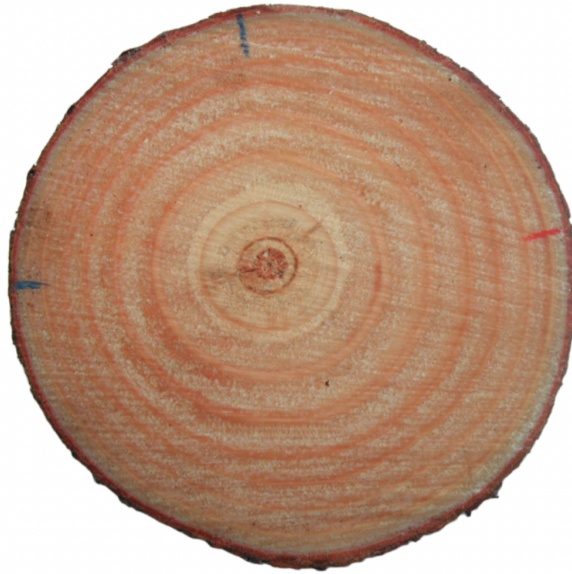


FIGURE 2.12 – Exemple d'image issue de l'article [Cerda et al., 2007].

La première étape de la méthode consiste à garder la composante S (saturation) de l'espace HSV comme image en niveaux de gris. La deuxième étape est une première détection des contours. Cela est fait avec la méthode de Canny. Ensuite, les contours obtenus sont filtrés en utilisant la propriété que les limites de cernes sont des transitions entre basse intensité (bois d'été) et forte intensité (bois de printemps). L'élagage des contours est fait en utilisant l'angle  $\theta$  entre le vecteur moelle-pixel et le vecteur du gradient :

$$\theta(\mathbf{p}) = \arccos \left( \frac{(\mathbf{p} - \mathbf{m}) \cdot \nabla I(\mathbf{p})}{\|\mathbf{p} - \mathbf{m}\| \|\nabla I(\mathbf{p})\|} \right) \quad (2.1)$$

où  $\mathbf{p}$  est le pixel détecté en tant que contour,  $\mathbf{m}$  la position de la moelle et  $\cdot$  le produit scalaire.

Un seuillage par hystérésis est appliqué sur l'image de l'orientation  $\theta$ . En effet, les auteurs indiquent qu'un simple seuillage ne donnent pas des contours suffisamment lisses pour la suite du traitement.

Ensuite, il est nécessaire de connaître le contour externe  $\mathcal{C} = (\mathbf{c}_i)_{i \in \mathbb{N}}$  de la coupe transversale. Toutes les cernes possibles sont simulées avec un pas de  $k$  :

$$\mathbf{r}_i(\mathbf{c}_i, \mathbf{p}, k) = k(\mathbf{c}_i - \mathbf{p}) + \mathbf{p} \quad (2.2)$$

Pour chaque pixel des contours, on incrémente de 1 le cerne simulé le plus proche. Un espace d'accumulation 1D est ainsi obtenu. Les maxima locaux sont considérés comme



FIGURE 2.13 – Exemple des cernes simulés d’après la méthode de [Cerda et al., 2007]. En (a) l’image en entrée, en (b) les cernes simulés d’après l’équation 2.2.

les cernes. Des conditions sur le maximum peuvent être ajoutées (comme celle d’avoir un minimum d’accumulations) afin de réduire le bruit si l’image est de basse résolution.

### 2.6.5 Norell (2011) [Norell, 2011]

Norell propose une méthode basée sur la transformée polaire des distances [Norell, 2011].

Les images ont été capturées en scierie. Elles ont une taille de  $1800 \times 1536$  pixels. Le jeu de données est décomposé en 24 images pour l’entraînement et de 20 images de validation. L’essence étudiée est le pin sylvestre.

La première étape est d’estimer la moelle. Le second objectif est de trouver une *bonne* région, c’est-à-dire une région où les cernes sont suffisamment visibles et marqués. Pour cela, l’orientation  $\theta$  des cernes est nécessaire. La méthode utilisée est la même que celle utilisée pour l’estimation de la moelle par [Norell and Borgefors, 2008]. La région choisie sera celle avec une direction optimale. Pour cela, l’image est décomposée en  $N$  sous-images  $P_i$ , qui sont des secteurs angulaires et qui se superposent par pas de  $\frac{2\pi}{N}$  (voir figure 2.14). Pour chaque région  $P_i$ , un calcul d’erreur est fait :

$$E_i = \sum_{\mathbf{p} \in P_i} |\theta_{gt}(\mathbf{p}) - \theta(\mathbf{p})| \quad (2.3)$$

où  $\theta_{gt}(\mathbf{p})$  est l’angle entre l’axe  $x$  et le vecteur moelle-pixel. La région avec l’erreur minimum est choisie pour la suite des opérations.

L’autrice propose d’estimer les limites de cernes de la manière suivante. À partir de la sous-image  $P_k$  pour laquelle l’erreur est minimale, deux transformées polaires de distances sont faites à partir des sous-images  $P_{k-2}$  et  $P_{k+2}$ . Ces deux cartes sont par la suite

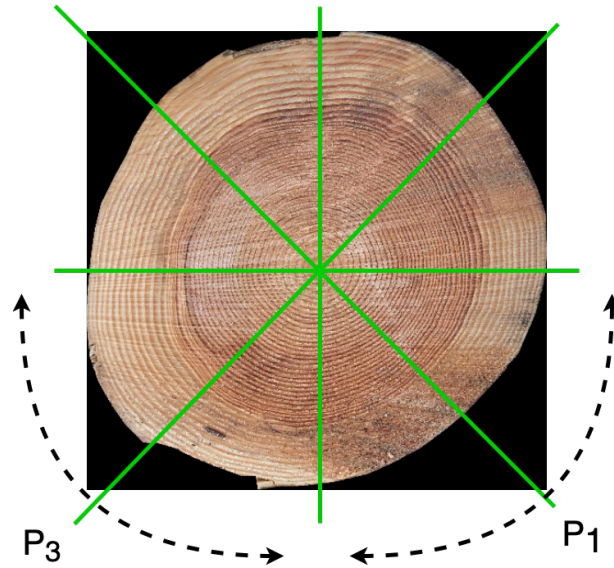


FIGURE 2.14 – Illustration de la décomposition en sous-images proposée par [Norell, 2011].

fusionnées au niveau de l'angle de la sous-image  $P_k$  par une méthode de programmation dynamique. Soient  $S$  et  $R$  deux signaux, alors on a :

$$C(R_1^i, S_1^j) = c(R_i, S_j) + \min \begin{cases} C(R_1^i, S_1^{j-1}) + g \\ C(R_1^{i-1}, S_1^{j-1}) \\ C(R_1^{i-1}, S_1^j) + g \end{cases} \quad (2.4)$$

où  $C(R_1^i, S_1^i)$  est le coût pour calibrer le signal  $R$  depuis l'indice 1 à l'indice  $i$  avec le signal  $S$  allant de l'indice 1 à  $j$ ,  $c(R_i, S_j)$  est une fonction de coût définie par l'écart au carré entre le signal  $R$  et  $S$  à l'indice respectif  $i$  et  $j$  et  $g$  est un paramètre d'ajustement.

Les signaux obtenus par la transformée polaire des distances sont traités en mettant à 0 les valeurs qui ne sont pas des extrema. Deux signaux sont calibrés (voir équation 2.4) à l'aide de la différence entre le signal original et celui avec la mise à 0 des non-extrema. Ces deux signaux calibrés servent alors à estimer les limites de cernes qui correspondent aux minimums locaux.

### 2.6.6 Zhou et al. (2012) [Zhou et al., 2012]

Zhou et al. proposent une méthode pour compter les cernes [Zhou et al., 2012].

Le jeu de données est composé de 5 images d'une taille inférieure à  $600 \times 600$  pixels. Comme pour la plupart des méthodes jusqu'alors développées, les rondelles ont été vraisemblablement traitées à la ponceuse. La figure 2.15 montre les cinq images.

Tout d'abord, la coupe transversale est segmentée dans l'espace HSV. Ensuite, les auteurs séparent l'aubier du duramen (si possible). Pour cela, une conversion en niveau gris est faite. La segmentation est obtenue par une combinaison de plusieurs watershed sur des fenêtres rectangulaires partant du centre de l'image jusqu'au bord de l'image. Les



FIGURE 2.15 – Les images utilisées dans l'article [Zhou et al., 2012].

auteurs comptent combien de fois  $H(i)$  est inférieur à  $H(j)$ , pour  $j$  allant de  $i - 10$  à  $i + 10$ . Si le nombre dépasse 16, alors le creux entre les deux pics est trouvé. La combinaison des watersheds (4 au total, un pour chaque direction) n'est pas précisée.

Pour les cerne, 4 lignes partant du centre vers le bord de l'image (une pour chacune des directions) sont tracées. Un watershed est appliqué sur l'histogramme des intensités des pixels de chaque ligne. Chaque pic obtenu après seuillage selon la valeur résultante du watershed, est considéré comme étant la limite d'un cerne.

Afin de corriger les potentielles erreurs, une normalisation selon la distance est faite pour les 4 directions. Un cerne avec une distance de 100 indique un cerne au bord de la coupe et la distance 0 indique une position proche de la moelle. Les auteurs supposent que les cerne décrivent des cercles concentriques. Ils choisissent la direction avec le plus de cerne qui sera la référence. Les trois autres servent alors de corrections. Chaque valeur est comparée avec la référence dans l'ordre des distances (c'est-à-dire que l'on compare de la distance la plus proche à la distance la plus éloignée). Si la distance est supérieure à un seuil donné, alors il faut ajouter un cerne à cette distance. Cette nouvelle distance étant la nouvelle référence. Une fois les 3 directions comparées, on passe au cerne suivant de la référence. On procède ainsi de suite jusqu'à ce qu'il n'y ait plus de cerne à ajouter. Ce procédé ne permet pas d'obtenir des résultats sous forme d'images, mais sous forme de courbes de  $n$  points (où  $n$  est le nombre de cerne détectés) en fonction de la distance à la moelle.

Cette correction apporte aussi des erreurs, notamment la possibilité d'avoir des cerne très proches. Une deuxième étape de correction est proposée où l'on compare les cerne deux à deux adjacents et l'on supprime l'un des deux cerne si la distance est inférieure à un seuil.

### 2.6.7 Sundari et al. (2014) [Sundari and Kumar, 2014]

Sundari et al. [Sundari and Kumar, 2014] soumettent une nouvelle méthode pour détecter les cerne sur une coupe transversale de bois. Leurs travaux s'inspirent de ceux de Borianne et al. [Borianne et al., 2011] sur des images tomodensitométriques de grumes de bois. Les images traitées sont de taille  $512 \times 512$  pixels.

Les images sont converties en niveaux de gris. Aucun détail n'est donné sur cette conversion. Après la conversion, l'image est lissée par un filtre Gaussien afin de réduire le bruit inter-cerne. Puis, l'image est seuillée selon sa norme du gradient obtenue par les masques Sobel-Feldmann. Pour chaque pixel seuillé, on associe un poids étant la probabilité qu'il s'agisse d'une limite de cerne. Cette probabilité est déterminée par la norme

du gradient.

À partir des cernes détectés, un contour actif est appliqué pour retirer les contours qui ne seraient pas des cernes. Les autrices ne mentionnent qu'une seule force, néanmoins ils s'inspirent des travaux de Borianne et al. [Borianne et al., 2011] pour lesquels la force totale décrivant le contour est :

$$\vec{F}_{totale} = k_0(k_1\vec{F}_A + k_2\vec{F}_B + k_3\vec{F}_R)$$

avec  $\vec{F}_A$  une force d'attraction,  $\vec{F}_B$  une force interne (*bending*) et  $\vec{F}_R$  une force de régularisation.

La force d'attraction permet au contour actif de s'approcher des contours précédemment seuillés. Elle s'exprime de la manière suivante :

$$\vec{F}_A = \frac{1}{m} \sum_{i=1}^m \nu(\mathbf{Q}_j) e^{-\frac{\|\overrightarrow{\mathbf{P}_i \mathbf{Q}_j}\|}{r}} \frac{\overrightarrow{\mathbf{P}_i \mathbf{Q}_j}}{\|\overrightarrow{\mathbf{P}_i \mathbf{Q}_j}\|} \quad (2.5)$$

où  $\nu(\mathbf{Q}_j)$  est la valeur associée au pixel  $\mathbf{Q}_j$ ,  $r$  est le rayon du voisinage et  $m$  est le nombre de pixels seuillés trouvé dans le voisinage. Borianne et al. ont utilisé la même force avec un  $r = 4$ .

La force interne permet de lisser le contour. Les autrices utilisent une force telle que la courbe soit de degré 2, c'est-à-dire :

$$\mathbf{P}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \mathcal{A} \begin{pmatrix} 1 \\ t \\ t^2 \end{pmatrix}$$

Pour chaque point  $\mathbf{P}_i$ , les coefficients de la matrice  $\mathcal{A}$  sont déterminés par la méthode des moindres carrés en utilisant les 7 points suivants et les 7 points précédents du contour. La projection  $\mathbf{P}'_i$  de  $\mathbf{P}_i$  sur le contour est calculée. La force interne est alors égale à :

$$\vec{F}_B = \overrightarrow{\mathbf{P}_i \mathbf{P}'_i} \quad (2.6)$$

Enfin, la force de régularisation permet que les points du contour restent proche entre eux deux à deux. Elle est définie par :

$$\vec{F}_R = |\alpha| \frac{\vec{X}}{\|\vec{X}\|} \quad (2.7)$$

où

$$\alpha = \frac{\|\overrightarrow{\mathbf{P}_{i-1} \mathbf{P}_i}\| - \|\overrightarrow{\mathbf{P}_i \mathbf{P}_{i+1}}\|}{2}$$

$$\vec{X} = \begin{cases} \overrightarrow{\mathbf{P}_{i-1} \mathbf{P}_i}, & \text{si } \alpha > 0 \\ \overrightarrow{\mathbf{P}_i \mathbf{P}_{i+1}}, & \text{si } \alpha < 0 \\ 0 & \text{sinon} \end{cases}$$

Contrairement à Borianne et al., les auteurs ne détaillent pas les valeurs des paramètres.

### 2.6.8 Kennel et al. (2015) [Kennel et al., 2015]

En 2015, Kennel et al. proposent une nouvelle approche pour délimiter les cerne [Kennel et al., 2015]. Cette approche combine deux autres approches : *Q-Shift Dual Tree Complex Wavelet Transform* (DT-CWT) et les contours actifs.

Les images utilisées dans l'article concernent des rondelles poncées. La taille des images n'est pas fournie, mais semble importante (voir [figure 2.16](#)).

La DT-CWT a été proposée par Kingsbury en 2001 [Kingsbury, 2001]. L'idée était de proposer une transformée en ondelettes complexes ayant des propriétés pour l'extraction d'informations. Parmi les propriétés il y a : une invariance par rotation, prise en compte des angles de  $\frac{\pi}{10}$ ,  $\frac{\pi}{4}$ ,  $\frac{2\pi}{5}$ ,  $\frac{3\pi}{5}$ ,  $\frac{3\pi}{4}$  et  $\frac{9\pi}{10}$  contre les angles 0,  $\frac{\pi}{2}$  et  $\frac{3\pi}{4}$  pour la transformée en ondelettes, une reconstruction parfaite avec une bonne localisation des fréquences. La DT-CWT serait meilleure que d'autres ensembles de filtres séparables (ondelettes de Gabor par exemple [De Rivaz, 2001]) car il y a moins de redondances. Le temps d'exécution est aussi plus rapide.

À partir de cette transformée, les auteurs proposent d'extraire les informations issues *Inter Coefficient Products* [Anderson et al., 2005]. Ces dernières sont regroupées en une carte d'orientation de textures. Une fois les informations extraites, Kennel et al. utilisent les contours actifs combinés aux cartes d'orientation de textures pour détecter les cerne.

Un contour  $c$  est composé  $n$  points. À chaque itération, les points  $\mathbf{P}_i$  sont mis à jour par la formule :

$$\mathbf{P}_i = \mathbf{P}_i + (1 - \gamma)(\mathbf{P}_i - \mathbf{P}'_i) + \alpha_i \vec{F}_{int} + \beta_i \vec{F}_{ext} \quad (2.8)$$

où  $\alpha_i$  et  $\beta_i$  sont des poids pour pondérer les forces internes et externes du contour,  $\gamma$  un paramètre global pour éviter les oscillations et  $\mathbf{P}'_i$  est la position du point obtenu à l'itération précédente.

La force interne, qui régule la position d'un point par rapport à ses voisins, est décrite par la formule suivante :

$$\vec{F}_{int} = \varepsilon_i \mathbf{P}_{i-1} + (1 - \varepsilon_i) \mathbf{P}_{i+1} + \mathcal{L}(r_i, \phi_i^*, \varepsilon_i) \vec{\mathbf{n}}_i \quad (2.9)$$

où

- $\varepsilon_i$  permet d'avoir la projection de  $\mathbf{P}_i$  sur la droite  $(\mathbf{P}_{i-1} \mathbf{P}_{i+1})$  ;
- $r_i$  est la demi-distance entre  $\mathbf{P}_{i-1}$  et  $\mathbf{P}_{i+1}$  ;
- $\phi_i^*$  est un angle qui définit une contrainte sur la continuité de la courbe à partir de l'angle  $\phi_i$  entre les deux segments  $\overrightarrow{\mathbf{P}_{i-1} \mathbf{P}_i}$  et  $\overrightarrow{\mathbf{P}_i \mathbf{P}_{i+1}}$  ;
- $\vec{\mathbf{n}}_i$  est le vecteur normal local de la courbe orienté vers l'extérieur ;
- $\mathcal{L}$  une fonction décrivant l'élasticité interne de la courbe.

La force extérieur  $\vec{F}_{ext}$  du point  $\mathbf{P}_i$  est une force attractive vers les points dits d'attraction  $\mathbf{A}_i$ . Les auteurs utilisent la DT-CWT, notamment le gradient, pour déterminer quels sont les points d'attractions. La recherche du point d'attraction est fait dans un voisinage local autour du point  $\mathbf{P}_i$  en prenant le point ayant la plus forte norme de gradient. Une fois ces points déterminés, il est possible de définir des couples  $(\mathbf{P}_i, \mathbf{A}_i)$  et ainsi de définir la force extérieure pour ce couple. Elle est égale à :



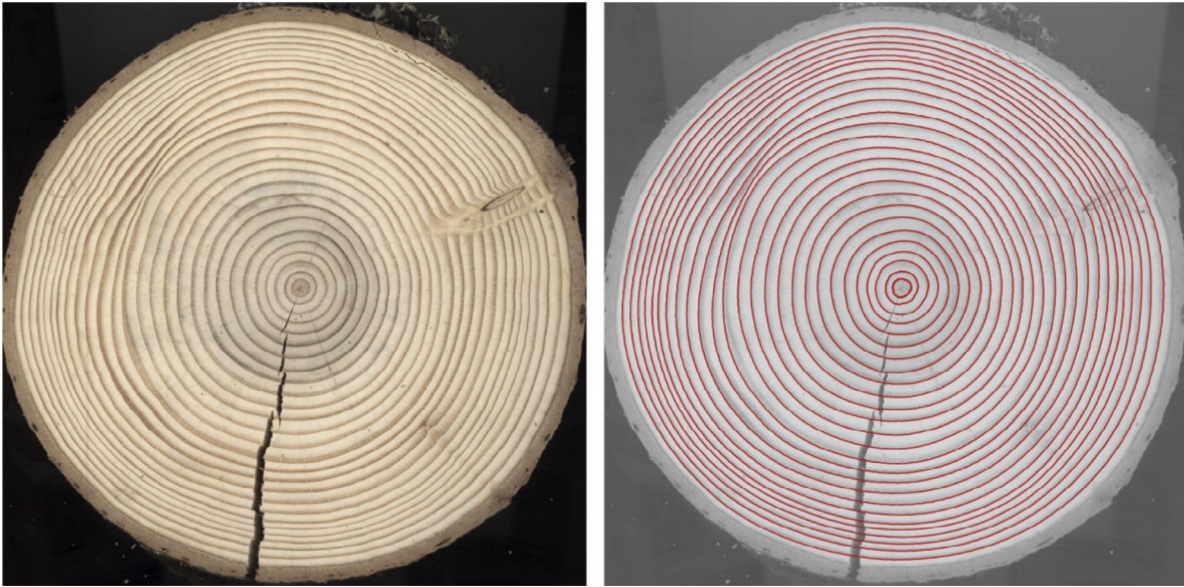


FIGURE 2.16 – Image extraite de [Kennel et al., 2015]. À gauche l'image en entrée, à droite la superposition avec les cernes détectés.

$$\vec{F}_{ext} = \frac{1}{2}(\Delta\nu_i + \frac{|d_i|}{\delta})\overrightarrow{\mathbf{P}_i\mathbf{A}_i} \quad (2.10)$$

où  $\Delta\nu_i$  est la différence de gradient (basé du la DT-CWT) entre le point  $\mathbf{P}_i$  et son attracteur  $\mathbf{A}_i$ ,  $d_i$  est la distance signée entre ces deux points et  $\delta$  est un paramètre de régularisation de l'attraction.

L'un des inconvénients de cette méthode est le temps de calcul. L'estimation se fait en 5 minutes et dépend de la qualité de l'image.

### 2.6.9 Fabijańska et Danek (2018) [Fabijańska and Danek, 2018]

L'article écrit en 2018 par Fabijańska et Danek ne traite pas directement d'images de coupes transversales mais de carottes extraites par une tarière de Pressler [Fabijańska and Danek, 2018]. Une fois la carotte prélevée, elle est scannée avec une résolution de 1200dpi. Les images sont en RVB et ont une taille allant de  $2281 \times 172$  pixels jusqu'à  $13975 \times 172$  pixels (voir figure 2.17). Au total 2500 images sont obtenues de cette manière-ci. Les images sont redécoupés sous formes de petites images de tailles  $128 \times 128$  pixels, ce qui a pour résultat d'avoir 80 000 images servant d'entraînement, de validation et de test.

Les autrices utilisent le réseau de neurones U-Net auquel elles ont apporté quelques modifications. Le nombre initial de filtres est modifié et le dropout est appliqué à quelques couches convolutives. Les images d'origine sont décomposées en sous-images de taille  $600 \times 172$  pixels (les sous-images ne se superposent pas), elle-même redécomposée en petit images de taille  $128 \times 128$  pixels. La taille des images en entrée du réseau est de  $128 \times 128$  pixels. Les 80000 images qui servent d'entraînement sont réparties à 80% pour

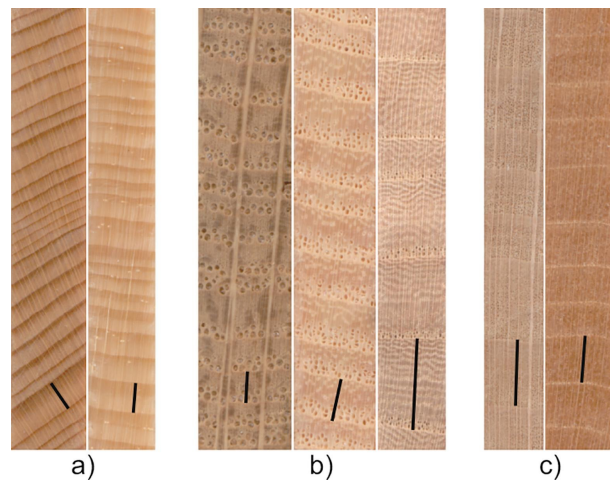


FIGURE 2.17 – Image extraite de [Fabijańska and Danek, 2018]. Le trait indique la largeur de cerne. De gauche à droite, les essences sont sapin blanc européen, épicéa commun, chêne, frêne européen, hêtre européen et tilleul à petites feuilles orme.

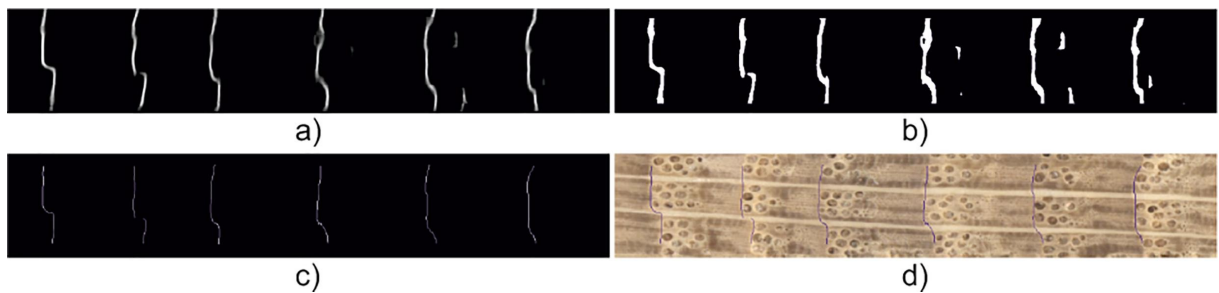


FIGURE 2.18 – Image extraite de [Fabijańska and Danek, 2018]. En a) la sortie du réseau de neurones, en b) le seuillage appliqué, en c) résultat après la squeletisation et en d) la superposition du résultat sur l'image d'entrée.

l'entraînement et 20% pour la validation. Les sous-images sont converties dans l'espace **HSV** et seule la composante **V** est gardée en guise d'entrée du réseau.

Un post-traitement est appliqué. En effet, le réseau de neurones donne une image qui est interprétée comme une carte de probabilité. L'image en sortie est seuillée par une méthode locale et adaptative [Singh et al., 2012]. Une fois le seuillage fait, une squeletisation est faite afin d'avoir des limites de cernes épaisses d'un pixel [Abu-Ain et al., 2013].

Pour montrer les performances, les autrices utilisent la précision et la sensibilité. D'après leurs résultats, le réseau a une sensibilité de 92% à 97% (dépendant de l'essence du bois) et une précision de 94% à 97% (voir figure 2.18). Aucun détail n'est fourni sur la taille du jeu de test ayant servi à obtenir ces chiffres.

Bien que leurs images soient de nature différentes des coupes transversales, il est envisageable d'appliquer le réseau sur une sous-image rectangulaire allant de la moelle jusqu'à l'écorce (la sous-image étant alors similaire aux leurs).

## 2.7 Conclusion

Le [tableau 2.1](#) synthétise brièvement les méthodes vues au cours de ce chapitre.

Pour la segmentation de la grume, la plupart des études n'ont pas la nécessité de le faire de manière automatique. Souvent la segmentation est manuelle. Les deux articles présentés ont deux approches différentes. Le premier repose sur l'accroissement de régions combiné à une analyse de texture et le deuxième utilise des réseaux de neurones. À noter que la segmentation de la grume est, dans les deux cas, utilisée pour identifier des caractéristiques afin de procéder à la traçabilité de la grume. Les images sont très similaires à celle du projet (voir [chapitre 3](#)).

Les articles sur l'aubier et le duramen sont très rares. La seule étude, à notre connaissance, repose sur une approche par accroissement de régions à laquelle un post-traitement est appliqué afin d'affiner le résultat. Comme pour la segmentation de la grume, cet article a des images similaires aux nôtres.

Enfin, toutes les méthodes pour la délimitation des cernes s'inspirent de diverses approches issues du traitement d'images. Nous pouvons souligner que le seuillage par hystérésis a été très utilisé (en combinaison avec d'autres méthodes). De même, le calcul de l'orientation des cernes semble avoir été très sollicité de la part des chercheuses et des chercheurs pour délimiter les cernes.

Cependant, une grande partie des articles qui abordent ce dernier sujet le font dans le cadre de la dendrochronologie. Cela implique que les images de coupes transversales ne sont pas de même qualité que celles capturées en scierie, sur le parc à grumes ou en forêt. L'étude qui est très proche de nos conditions d'acquisition (et de type d'images) est celle proposée par Norell [[Norell, 2011](#)]. Mais le temps de calcul est coûteux et cela peut être un frein pour une réelle application.

	Techniques principales	Types d'images	Résolution (en pixels)	Avantages	Inconvénients
<b>Grume</b>					
[Schraml and Uhl, 2014]	Accroissement régions	RVB APN	1024 × 768	Rapide	Faible résolution
[Wimmer et al., 2021]	Réseaux de neurones	RVB APN	Non communiqué	Bons résultats	Apprentissage à faire
<b>Duramen</b>					
[Raatevaara et al., 2020]	Accroissement de régions	RVB APN	2400 × 1344 6000 × 3368		Temps fournis Moelle
<b>Cernes</b>					
[Conner et al., 1998]	Seuillage	Scanner	Non fournis	Simple Rapide	Haute qualité d'image
[Soille and Misson, 2001]	Morphologie mathématique	Scanner	7625 × 7531		Traitement manuel
[Laggoune et al., 2005]	Filtre détection contour	Non fourni	Non fournie	Simple	Peu de détails
[Cerde et al., 2007]	Simulation de cernes	RVB	700 × 700	Rapide	Sections circulaires
[Norell, 2011]	Transformée polaire des distances	RVB APN	1800 × 1536	Très bons résultats	Temps
[Zhou et al., 2012]	Seuillage	RGB Scanner	600 × 600	Simple Rapide	Traitement manuel
[Sundari and Kumar, 2014]	Contours actifs		512 × 512	Cernes complets	Temps
[Kennel et al., 2015]	Contours actifs	Non fourni	Non fournie	Cernes complets	Temps
[Fabijańska and Danek, 2018]	Réseaux de neurones	RVB Scanner	2281 × 172 13975 × 172	Précis Robuste	Limité à leur images

TABLE 2.1 – Résumé des méthodes vues au cours de ce chapitre avec des avantages et limites pour notre cas d'études.



# Nouvelles approches du traitement d'images appliquées au bois

## Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>90</b>
<b>3.2</b>	<b>Jeux de données</b>	<b>90</b>
3.2.1	Capteurs utilisés	91
3.2.2	Corcieux	92
3.2.3	FVA	93
3.2.4	Besle	93
3.2.5	BBF	93
3.2.6	Logyard	94
3.2.7	Sawmill	95
<b>3.3</b>	<b>Segmentation de la grume</b>	<b>96</b>
3.3.1	K-means	97
3.3.2	Contours actifs	100
3.3.3	Réseaux de neurones convolutifs	101
3.3.4	Résultats	102
<b>3.4</b>	<b>Segmentation de l'aubier et du duramen</b>	<b>106</b>
3.4.1	Réseaux de neurones convolutifs	107
3.4.2	Résultats	110
<b>3.5</b>	<b>Détection des cernes</b>	<b>111</b>
3.5.1	Filtrage directionnel	113
3.5.2	Filtrage directionnel : application au bois	115
3.5.3	Seuillage	117
3.5.4	Affinement du seuillage	117
3.5.5	Calcul de la largeur moyenne	122
3.5.6	Résultats	127
<b>3.6</b>	<b>Synthèse</b>	<b>128</b>

---

## 3.1 Introduction

Nous venons de voir un ensemble de méthodes issues du traitement d'images qui ont été mises en œuvre pour détecter des éléments visibles sur une coupe transversale d'une grume de bois. Rappelons que les principaux éléments visibles sur ce type de coupe, qui peuvent servir à avoir des informations sur la qualité du bois, sont : la position de la moelle et du centre géométrique de la grume, la quantité d'aubier et de duramen et les informations relatives aux cernes (notamment leur largeur). Les applications présentées ont été conçues principalement dans le cadre de la **dendrochronologie**, à l'exception de quelques unes qui ont été appliquées dans un cadre industriel sur des images brutes de coupes transversales, comme dans notre cas. Puisque les conditions, dans le cadre du projet, ne sont pas similaires à celles du cadre de la dendrochronologie, il est nécessaire de développer des méthodes, ou d'adapter les méthodes existantes, afin d'obtenir les résultats escomptés.

Dans ce chapitre nous proposons des nouvelles méthodes pour le traitement d'images du bois. Nous présenterons tout d'abord, dans la **section 3.2**, les jeux d'images qui ont été acquis durant le projet ANR TreeTrace. Ces images ont été capturées directement sur le terrain, aussi bien en forêt qu'en scierie (parc à grumes et avant la découpe). Ces images sont brutes, c'est-à-dire qu'elles ont été directement après la découpe sans traitement postérieur (présence d'autres grumes, de l'abatteuse, ...), il faut retirer les éléments non pertinents pour la suite de notre étude. Ainsi, nous présenterons les travaux proposés pour la segmentation de la grume dans la **section 3.3**. Cette segmentation permettra aussi d'en déduire la position du centre géométrique. Une fois la segmentation faite, les éléments visibles et utiles pour estimer la qualité du bois peuvent être analysés plus aisément. Dans la **section 3.4**, nous aborderons les méthodes développées pour la segmentation du duramen. En combinant la première segmentation, nous en déduisons la quantité d'aubier. Enfin, nous détaillerons les approches proposées pour détecter les cernes dans la **section 3.5**.

## 3.2 Jeux de données

Les images utilisées dans ce manuscrit et acquises durant le projet TreeTrace ont été prises avec différents appareils, principalement des smartphones. Ces images sont toutes des coupes transversales de grumes de Douglas, mais également d'épicéa suite à la demande de nos partenaires autrichiens du projet. Les images ont été prises dans différents environnements : en forêt, sur le parc à grumes et en scierie. L'intérêt est double. D'abord, celui d'avoir des conditions réelles d'acquisition afin d'avoir des images au plus proche de celles observables dans un cadre industriel. Ces conditions empêchent de faire des traitements, comme c'est le cas dans de nombreuses études (voir le **chapitre 2**), afin d'améliorer la qualité des images. Il n'y a donc pas eu de ponçage de la grume, de traitement chimique, etc. L'autre intérêt d'être dans ces conditions, est de tester la robustesse des méthodes développées. L'inconvénient est d'ajouter une difficulté supplémentaire aux traitements.

Nous décrirons d'abord les capteurs employés pour l'acquisition. Finalement, les différents jeux d'images, triés par type de capteur, seront présentés.

Toutes les données qui sont présentées dans ce chapitre font l'objet de deux articles

Capteur	Capteur	Hauteur $\times$ Largeur (en pixel)
Huawei PRA-LX1	CMOS	3968 $\times$ 2976
Huawei ANE-LX1	CMOS	4608 $\times$ 3456
Iphone X	CMOS	3024 $\times$ 4032
Panasonic Lumix DMC-FZ45	CMOS	4320 $\times$ 3240
Sony IMX183	CMOS	5472 $\times$ 3648

TABLE 3.1 – Spécifications des capteurs utilisés pour les jeux de données

Jeu d’images	Type de capteur	Nombre d’images
Corcieux	Huawei PRA-LX1 & Panasonic Lumix DMC-FZ45	1013 & 1009
FVA	Huawei PRA-LX1	760
Besle	Panasonic Lumix DMC-FZ45	65
BBF	Huawei 45 ANE-LX1	45
Logyard	Iphone X	208
Sawmill	Sony IMX183	150

TABLE 3.2 – Jeux de données, capteurs correspondant et nombre d’images.

dont je suis co-auteur : un sur l’épicéa et l’autre sur le Douglas (voir [section 6.2](#)). L’objectif de ces articles est de mettre les données à disposition de la communauté scientifique qui développe des méthodes pour le traitement des images de bois.

### 3.2.1 Capteurs utilisés

Les images acquises pour le projet TreeTrace, et qui seront utilisées dans la suite de ce manuscrit, ont été prises avec différents appareils, principalement des smartphones. La diversité des appareils de capture implique une taille et une résolution d’images différentes pour les jeux de données. Le [tableau 3.1](#) résume les différents capteurs qui ont été utilisés. Les trois premiers sont des capteurs issus de smartphones (Huawei et IphoneX). L’avant-dernier est issu d’un appareil photo numérique et le dernier est une caméra de type industrielle.

Les images composant les jeux ont été prises dans différents lieux et à des dates différentes. Les images représentent des coupes transversales d’extrémités de grumes, acquises dans des conditions réelles : en forêt, sur le parc à grumes et en scierie. De telles images permettront également de déterminer la robustesse des outils de traitement d’images mis en œuvre. Un sommaire des jeux de données est visible dans le [tableau 3.2](#). Le jeu de données Corcieux est divisé en deux parties. En effet, pour ce jeu d’images, les images ont été acquises deux fois. La première fois avec le capteur Huawei PRA-LX1 et la deuxième fois avec le capteur Panasonic LumixDMC-FZ45. Chaque jeu d’images va être détaillé un par un afin de préciser la méthode d’acquisition et d’exposer le type d’images acquises. Si aucune méthode d’acquisition n’est donnée, alors s’il s’agit simplement d’une photo prise à la main.





FIGURE 3.1 – Exemples issus du jeu de données Corcieux avec le capteur Huawei PRA-LX1.



FIGURE 3.2 – Exemples issus du jeu de données Corcieux avec le capteur Panasonic Lumix DMC-FZ45.

### 3.2.2 Corcieux

Ce jeu d'images est uniquement composé de coupes transversales d'épicéa. Les images ont été acquises en septembre 2018. Les images ont été prises à l'aide de l'appareil Huawei PRA-LX1, directement après l'abattage des arbres en forêt. Afin de savoir s'il y avait une différence entre capteurs, les images ont aussi été prises par le capteur Panasonic Lumix DMC-FZ45. Avec le premier capteur, les images ont une taille de  $3968 \times 2976$  pixels. Alors qu'avec le second capteur, les images ont une taille de  $4320 \times 3240$  pixels.

Le jeu de données est composé de 100 grumes. Les deux extrémités de chaque grume ont été photographiées. Chaque extrémité a été prise en photo quatre fois en inclinant légèrement l'appareil de quelques degrés entre chaque. L'objectif de ces rotations est d'augmenter le volume de données et de tester la robustesse des algorithmes. À cela s'ajoute une image avec le numéro de la grume agrafé dessus. Cela donne 5 images par extrémité, soit 10 images par grume. Au total, 1013 images ont été prises avec l'appareil Huawei et 1009 images avec l'appareil Panasonic. Il y a plus d'images que prévu, car certaines extrémités ont été prises plus de 4 fois.

Sur chaque image, un damier a été posé au-dessus de la grume. Chaque carreau du damier fait 1 cm. Il a pour but de calibrer spatialement les images. Cela servira pour estimer la précision des algorithmes par rapport aux vérités terraines faites, comme les largeurs de cernes mesurées sur le bois. La [figure 3.1](#) montre quelques exemples issus de ce jeu avec l'appareil Huawei, et la [figure 3.2](#) montre les mêmes grumes prises avec le Panasonic.



FIGURE 3.3 – Exemples issus du jeu de données FVA. Il s’agit des mêmes grumes qu’aux figure 3.1 et figure 3.2.

### 3.2.3 FVA

Tout comme le jeu de données précédent, celui-ci a vu le jour à la demande des partenaires autrichiens. Ce jeu de données a été fait quelques jours après le précédent. Les images sont celles des mêmes grumes que le jeu de données de Corcieux, à la différence qu’une rondelle d’environ 5 cm a été prélevée à chaque extrémité. Autre différence, les images ont été prises sur un parc à grumes et non en forêt. Les photos ont été prises à l’aide du capteur Huawei PRA-LX1. Les images sont de taille  $3968 \times 2976$  pixels. Les deux extrémités ont été prises 4 fois : la première photo a été prise avec une étiquette puis la rondelle a été sciée et 3 photos ont ensuite été prise de la face rafraîchie (avec une légère inclinaison entre chaque prise). Le jeu possède 760 images. Certaines extrémités ont été prises moins de quatre fois, ce qui explique un nombre plus bas qu’attendu.

Comme pour le jeu d’images précédent, un damier est présent afin calibrer spatialement les images. La figure 3.3 montre quelques exemples de ce jeu de données. Ce sont les mêmes grumes qu’aux figure 3.1 et figure 3.2 afin de visualiser le rafraîchissement qui a été opéré.

### 3.2.4 Besle

Dans le cadre du projet Treetrace, des photos de douglas ont été prises. Ce jeu a été fait en novembre 2018. Les photos ont été faites en forêt directement après la découpe de l’arbre par l’abatteuse. Toutes les images ont été acquises par le capteur Panasonic Lumix DMC-FZ45, ce qui implique que les images ont une taille de  $4320 \times 3240$  pixels. Le jeu est composé de 3 grumes, qui ont été sectionnées plusieurs fois. Pour chaque nouvelle coupe, plusieurs images ont été photographiées avec des angles différents, toujours dans l’objectif d’augmenter le jeu de données et de mesurer la robustesse des méthodes. De plus, des photos de la souche ont été faites. En effet, un objectif du projet est de pouvoir effectuer une traçabilité de la grume de la forêt à la scierie. Ainsi, il a été question de savoir s’il était possible d’identifier une grume à partir de l’image de la souche et celle de la première section. Au total, le jeu comporte 65 images. Dans la même démarche, un damier est présent sur chaque photo afin de pouvoir effectuer la conversion pixel/millimètre. La figure 3.4 montre des exemples issus de ce jeu d’images.

### 3.2.5 BBF

Ce jeu de données est composé des mêmes grumes que le jeu Besle, seulement les photos ont été prises sur le parc à grumes quelques semaines plus tard (le temps de transférer les



FIGURE 3.4 – Exemples issus du jeu de données Besle. La dernière image est celle de la souche.



FIGURE 3.5 – Exemples issus du jeu de données BBF. Les mêmes grumes que celles montrées pour le jeu Besle (à l'exception de la dernière).

grumes de la forêt au parc). Les images ont été acquises par le capteur Huawei ANE-LX1 et ont une taille de  $4608 \times 3456$  pixels. Ces images ont le même objectif que celles des souches : est-il possible d'identifier la grume quelques semaines après leur découpe à partir d'une photo d'une coupe transversale ? Le damier est toujours présent sur ces images. La [figure 3.5](#) montre quelques photos du jeu de données. Nous pouvons voir des problèmes que posent ce type d'images : salissures sur la section durant le transport, présence de résine qui a coulé au niveau de l'aubier, ...

### 3.2.6 Logyard

En janvier 2020, une campagne a été faite afin d'acquérir davantage de photos de douglas. Les photos sont celles de 52 arbres provenant de 4 sites différents (soit 13 arbres par site). Les images ont été acquises avec un iPhone X et ont une taille de  $3024 \times 4032$  pixels. Les photos ont été prises sur le parc à grumes d'une scierie. Pour chaque grume, plusieurs images ont été prises à des hauteurs différentes dans la grume lors de la découpe de celle-ci en 3 billons de 4 m de longueur. Les sections sont identifiées à partir du site (une lettre : A, B, C ou D), du numéro de la grume (1 à 13) et de la hauteur dans l'arbre (a,b,c ou d, avec  $a$  proche du sol et  $d$  proche de la cime par tranche de 4 mètres). La [figure 3.6](#) synthétise la méthode de prise des photos pour un arbre. La provenance des arbres est la suivante :

- Sémelay : A
- Chaulgnes Bertranges : B
- Iboy le pré : C
- Anost : D

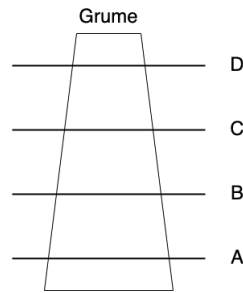


FIGURE 3.6 – Schéma représentant les parties de la grume prises en photo aussi bien pour le jeu **Logyard** que pour le jeu **Sawmill**.



FIGURE 3.7 – Exemples issus du jeu de données Logyard. Il s'agit des quatre section la même grume : D02a, D02b, D02c et D02d.

Au total 208 images ont été acquises. La [figure 3.7](#) montre des exemples de ce jeu de données. Les images ont pour identifiant : D02a, D02b, D02c et D02d. Autrement dit, il s'agit des images de la grume numéro 2 (02) provenant du site Anost (D) et des 4 hauteurs (a, b, c et d) de la grume. Nous pouvons constater qu'en général plus la coupe est proche de la souche, moins elle est circulaire. Comme pour les jeux précédents, un damier est présent sur les photos pour effectuer la conversion pixel/millimètre.

### 3.2.7 Sawmill

Dans la campagne de janvier 2020, les mêmes arbres ont été photographiés en scierie sur la chaîne de traitement juste avant la découpe. Sur la chaîne de traitement de la scierie une caméra a été installée de façon à prendre une photo des billons gros bout uniquement (i.e. le côté ayant le plus gros diamètre, principalement celui le plus proche de la souche). Il s'agit des mêmes 52 arbres que pour le jeu d'images précédent. Un rafraîchissement a été opéré pour chaque billon. Pour chaque site, deux billons ont été retirés pour faire des mesures de résistance physique. De même la hauteur  $d$  ne peut être prise en photo s'agissant de la fin de l'arbre. Le jeu est donc composé de 150 images, deux images n'ont pas été prises (bien qu'il soit possible d'apercevoir le gros bout du billon sur d'autres images). Le capteur est Sony IMX183, les images ont une taille de  $5472 \times 3648$  pixels.

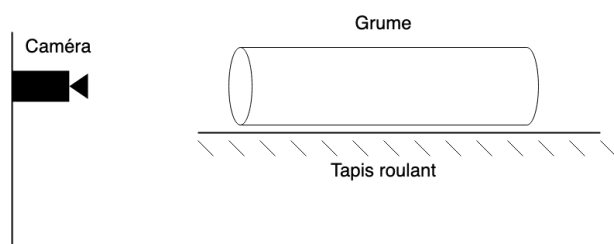


FIGURE 3.8 – Schéma représentant l'installation du capteur en scierie pour le jeu de données de la scierie.



FIGURE 3.9 – Exemples issus du jeu de données Sawmill. De gauche à droite : D02a, D02b, D02c et A01b.

La [figure 3.8](#) montre l'installation de la caméra qui a servi pour la capture des images en scierie.

En scierie, aucun damier n'a pu être placé pour des raisons de sécurité. Néanmoins, un télémètre a été positionné afin de connaître la distance entre la caméra et la grume. Cela permet de faire la conversion pixel/millimètre étant donné que l'on connaît la résolution des images et la focale de l'appareil. La [figure 3.9](#) montre quelques exemples d'images de ce jeu de données. Les images ont pour identifiant : D02a, D02b, D02c et A01b.

### 3.3 Segmentation de la grume

La présentation des jeux d'images permet de se rendre compte que les images ne sont pas similaires à celles fréquemment utilisées dans l'état de l'art (voir [chapitre 2](#)). En effet, de nombreux éléments viennent rendre difficile l'analyse de ces images. Parmi ces éléments, nous pouvons citer la présence de salissures dû au transport, la résine qui a coulé entre l'abattage de l'arbre et son passage sur la chaîne de traitement en scierie, les coups de sciages, ... Tous ces éléments ne sont pas présents lorsque le contexte est la [dendrochronologie](#) puisque le contexte permet d'effectuer un certain nombre de traitements physiques (ponçage) et/ou chimiques.

Dans cette section, nous abordons les méthodes mises en œuvre pour segmenter la section transversale de bois. Cette segmentation est une étape importante, elle permet de simplifier la détection des autres éléments importants dans l'estimation de la qualité du bois. Elle permet également de déduire le centre géométrique de la grume (i.e. le barycentre des points composant la grume). Toute cette section a fait l'objet d'un article publié [[Decelle and Jalilian, 2020](#)] (voir [section 6.2](#)).

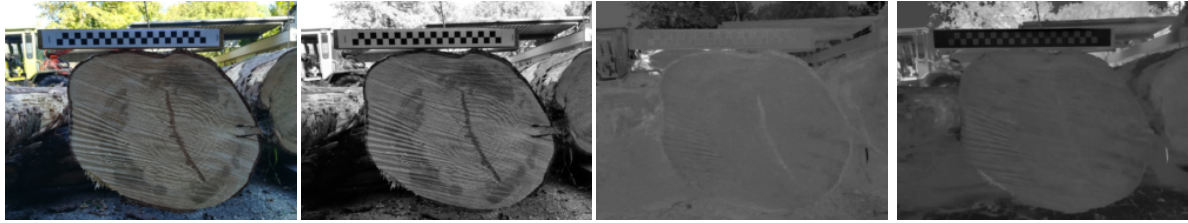


FIGURE 3.10 – Image d'exemple et les trois composantes de l'espace **CieLAB** dans l'ordre  $L$ ,  $a^*$  et  $b^*$ .

### 3.3.1 K-means

À la vue des images des différents jeux de données, il semble normal de se demander s'il n'existerait pas une méthode rapide pour segmenter la grume dans l'image. En effet, deux caractéristiques des grumes sont à souligner. La première est la colorimétrie de la grume qui se démarque souvent du reste de l'image, quand celle-ci est seule. La deuxième est que la grume est proche d'un cercle. Il est pertinent de regarder dans ces deux directions.

La première approche proposée se base sur l'algorithme K-Means. Celui-ci va permettre de regrouper les pixels de couleurs similaires dans un même cluster, le nombre de clusters  $n$  étant un paramètre. L'idée est d'avoir un cluster formé uniquement des pixels de la section transversale de la grume. Le K-Means est une méthode proche de l'algorithme SLIC détaillé au [chapitre 1](#), à ce détail près que les coordonnées  $(x, y)$  des centres des clusters ne sont pas prises en compte. Nous avons utilisé la version proposée par Arthur et al. [[Arthur and Vassilvitskii, 2006](#)] qui développent un algorithme plus performant que l'algorithme initial : la solution optimale est trouvée et le temps de calcul est réduit.

La méthode est appliquée sur l'espace de couleur **CieLAB** (voir [figure 3.10](#)). Le choix s'est porté sur cette représentation car les pixels de section transversale sont souvent dans des teintes proches (Hue). De plus, le bord de la section est généralement foncé, ce qui se transcrit bien dans cet espace (faible saturation, teinte unie, faible brillance). De plus, la taille de l'image est divisée par 4 afin de réduire le bruit et limiter le temps de calcul.

En appliquant K-Means avec  $n = 6$  (le nombre de clusters), nous obtenons les clusters visibles sur la [figure 3.11](#). Le choix de  $n$  s'est fait de manière empirique sur un ensemble d'images. Nous avons observé que pour  $n = 6$ , la grume semble appartenir à un unique cluster tout en se distinguant des éléments non pertinents pour cette segmentation. Le regroupement des pixels ne se fait pas spatialement ; c'est-à-dire, que deux pixels à l'opposé l'un de l'autre peuvent appartenir au même cluster même s'ils ne définissent pas la même chose. Dans la [figure 3.11](#), nous pouvons observer que le blanc de la mire appartient au même cluster que le sol. Pour limiter ce problème, nous avons décomposé chaque cluster en plus petits groupes en cherchant les composantes connexes. Ces composantes connexes sont illustrées pour chaque cluster dans les [figure 3.11](#).

Une fois les pixels regroupés, il faut sélectionner la composante correspondant à la section transversale de la grume. Nous allons utiliser la deuxième caractéristique de la grume que nous avons mentionnée, à savoir la circularité de la grume. Nous définissons la circularité, notée  $C$ , comme un rapport entre l'aire  $\mathcal{A}$ , c'est-à-dire le nombre de pixels, et le périmètre  $\mathcal{P}$ , i.e. le nombre de pixels appartenant au contour, d'une composante :

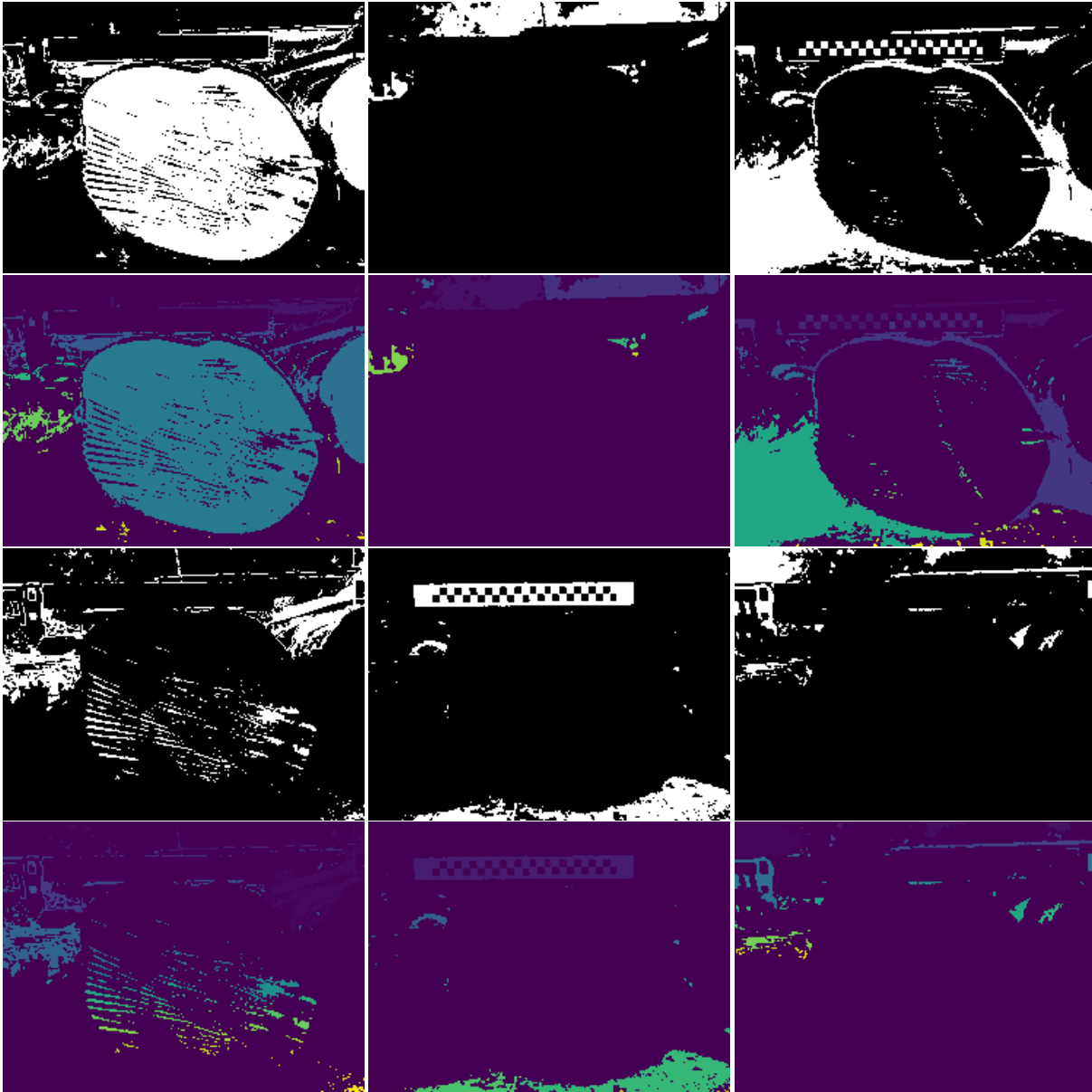


FIGURE 3.11 – Les 6 clusters (image en noir et blanc) trouvés par la méthode du K-Means [Arthur and Vassilvitskii, 2006] et les composantes connexes associées (sous chaque clusters, une couleur par composante).

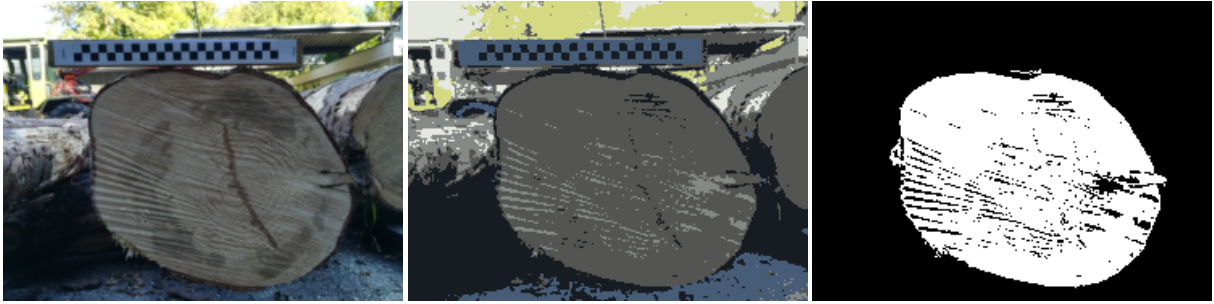


FIGURE 3.12 – Exemple de l’algorithme proposée pour la segmentation de la grume en utilisant l’algorithme de KMeans [Arthur and Vassilvitskii, 2006]

$$C = \frac{4\pi\mathcal{A}}{\mathcal{P}^2} \quad (3.1)$$

Par exemple, pour un cercle parfait  $C = 1$  et pour un rectangle de largeur 1 et longueur 2 la circularité vaut environ 0,698.

Pour sélectionner la composante connexe qui représente la section transversale, nous avons tout d’abord choisi de sélectionner la composante connexe la plus grande pour chaque cluster (ce qui donne 6 composantes connexes). Puis, parmi ces composantes, nous prenons celle ayant la circularité la plus proche de 1. La [figure 3.12](#) montre l’image d’origine, le regroupement par cluster, c’est-à-dire que pour chaque pixel d’un cluster nous lui associons la couleur du barycentre, et enfin la composante connexe retenue selon les critères ci-dessus. Nous pouvons observer que la grume est bien segmentée malgré la présence de trous dans la segmentation. Ces défauts peuvent être corrigés par un traitement a posteriori.

Les avantages de cette méthode sont sa rapidité et sa simplicité. Par ailleurs, elle est facilement adaptable; c’est-à-dire que si l’espace de couleur **CieLAB** n’est pas le plus pertinent pour effectuer l’algorithme K-Means, il est possible d’utiliser une autre représentation de couleur. De même, que le critère  $C$  de sélection peut-être ajusté aux données à traiter.

Le défaut majeur de cette méthode est principalement dans le critère de sélection de la composante connexe. En effet, le critère de circularité n’est pas toujours pertinent. Par exemple, dans la [figure 3.13](#) la grume ne semble pas très circulaire. Si l’on calcule la circularité  $C$  pour cette grume, dont la vérité terrain est montrée à droite dans la figure, on obtient une valeur d’environ 0,506, plus petite que celle d’un carré. C’est la taille qui fait, dans ce cas précis, le premier critère de sélection et non la circularité. Autre limite, nous avons décidé de prendre la plus circulaire des plus grandes des composantes connexes. Ce choix s’explique par le fait que le grume est notre centre d’intérêt et que l’on suppose que l’élément est le plus visible dans l’image (si d’autres extrémités de grumes sont présentes dans l’image).

Des détails sur les performances de cette méthode, ainsi qu’une discussion vis-à-vis des autres méthodes détaillées dans les sections suivantes, sont présentés dans le [tableau 3.3](#).





FIGURE 3.13 – Contre-exemple montrant une limite du calcul de circularité.

### 3.3.2 Contours actifs

Nous avons vu dans le [chapitre 2](#) que les contours actifs ont été plusieurs fois utilisés pour à l'analyse d'images de coupes transversales.

Cependant, l'utilisation du contour telle que décrite dans le [chapitre 1](#) présente un inconvénient dans la segmentation de la coupe transversale. En effet, les cernes présentent une norme du gradient potentiellement élevée, conduisant l'une des forces à être très importante. Une façon de résoudre cet inconvénient est de ne pas prendre en compte la norme du gradient dans le contour actif.

La formulation s'inspire du modèle de Mumford-Shah [[Mumford and Shah, 1989](#)]. Mumford et Shah approximent l'image  $I$  par une fonction continue par morceaux  $U$ , l'objectif est de trouver le minimum l'équation :

$$\mathcal{E}(U, C) = \mu \mathcal{P}(C) + \lambda \int_E (I(x) - U(x))^2 dx + \int_{E \setminus C} \|\nabla U(x)\|^2 dx$$

où  $C$  est le contour actif,  $\mathcal{P}(C)$  est la longueur du contour qui permet sa régularité, et  $\mu$  et  $\lambda$  deux paramètres. Le deuxième terme permet de s'assurer que  $U$  s'approche de  $I$  et le troisième terme assure que  $U$  est différentiable en tout point du domaine  $E$  à l'exception des points du contour (due à l'approximation par une fonction continue par morceaux). Malheureusement, cette formulation est coûteuse en terme de calcul.

Chan et Vese proposent de simplifier la fonction  $U$  [[Chan and Vese, 2001a](#)]. Cette dernière ne peut avoir que deux valeurs possibles qui sont les suivantes :

$$\forall x \in E, \quad U(x) = \begin{cases} c_1 & \text{quand } x \text{ est à l'intérieur de } C \\ c_2 & \text{quand } x \text{ est à l'extérieur de } C \end{cases}$$

Le deuxième apport de Chan et Vese est d'ajouter une contrainte supplémentaire sur le contour  $C$  qui prends en compte l'aire qu'occupe l'intérieur de la courbe. La formule à minimiser est :

$$\mu \mathcal{P}(C) + \nu \mathcal{A}(C) + \lambda_1 \int_{\text{interieur}} |I(x) - c_1|^2 dx + \lambda_2 \int_{\text{exterieur}} |I(x) - c_2|^2 dx$$

où  $\mathcal{A}(C)$  est l'aire contenue à l'intérieur du contour et  $\nu$  un nouveau paramètre.

La paramètre  $\nu$  définit une contrainte sur la zone couverte par le contour  $C$ . Ce paramètre n'a de sens que si le contour doit atteindre les bords de l'image [Getreuer, 2012]. En effet, pour un contour fermé si l'on cherche à minimiser l'aire avec  $\nu > 0$ , alors le contour vide minimise l'aire.

Dans le cadre de l'application à la segmentation de la coupe transversale, l'algorithme du contour actif avec la formulation ci-dessous, a été appliqué sur la composante  $L$  de l'espace CieLAB. Puis, les images ont été re-dimensionnées avec une taille de  $512 \times 512$  pixels par interpolation bilinéaire afin de réduire le temps de calcul. Les paramètres ont été fixés aux valeurs suivantes :  $\mu = 0.5$ ,  $\nu = 0$  et  $\lambda_1 = \lambda_2 = 1$ . Le contour initial était un cercle centré sur l'image d'un rayon de 128 pixels. Le nombre d'itérations a été fixé à 400. Les images résultats sont présentés dans les figure 3.15h et figure 3.16h. Les résultats quantitatifs sont présentés dans la tableau 3.3. Brièvement, les résultats sont moyens. Bien que les contours actifs arrivent à localiser la coupe et à avoir une forme approximatif, la segmentation reste très imprécise avec parfois des éléments non souhaités qui sont segmentés.

Les avantages des contours actifs résident dans les forces utilisées. Elles sont intuitives à comprendre et il est facile d'ajouter des forces pour s'adapter aux problèmes.

Néanmoins, la convergence vers le minimum peut être longue et n'est pas forcément garantie. Le temps de calcul peut varier de quelques secondes avec un contour avec des forces simples à quelques minutes dès que l'on ajoute des forces plus précises et adaptées au problème. Ce dernier point est essentiel, car le temps est un facteur important dans le cadre industriel.

### 3.3.3 Réseaux de neurones convolutifs

Il existe beaucoup de réseaux de neurones convolutifs pour la segmentation d'images. Dans le cadre du projet, nous avons évalué 4 réseaux qui offrent de bons résultats pour la segmentation de manière générale. Ces quatre réseaux sont : U-Net [Ronneberger et al., 2015], SegNet [Badrinarayanan et al., 2017], RefineNet [Lin et al., 2017] et Mask R-CNN [He et al., 2017]. L'architecture de chacun de ces réseaux a été décrite dans la sous-section 1.7.1.

Nous avons proposé un modèle légèrement différent pour U-Net qui s'inspire de celui proposé par [Fabijanska et al., 2017]. En effet, depuis l'architecture proposée de U-Net en 2015, de nombreux apports ont été faits. Celui qui a retenu notre attention concerne la régularisation des réseaux : le *dropout* (voir le chapitre 1). L'architecture proposée est la suivante.

La partie encodeur, la première convolution est de taille  $11 \times 11$  avec *dropout*, suivi d'une seconde convolution de taille  $11 \times 11$ . Chaque convolution est suivie par la fonction d'activation ReLU (voir la section 1.7). L'échantillonnage est de taille  $2 \times 2$  avec un stride de 2 et en prenant le maximum. Le premier bloc est composé de 16 filtres. À chaque réduction, le nombre de filtres est doublé et la taille des filtres est diminuée de 2 jusqu'à la taille minimale de  $3 \times 3$ , c'est-à-dire que l'on commence par des filtres de taille  $11 \times 11$ , puis  $9 \times 9$ ,  $7 \times 7$ ,  $5 \times 5$  et finalement  $3 \times 3$ . La partie décodeur est identique au réseau d'origine (U-Net [Ronneberger et al., 2015]), avec simplement l'ajout de couche de *dropout*

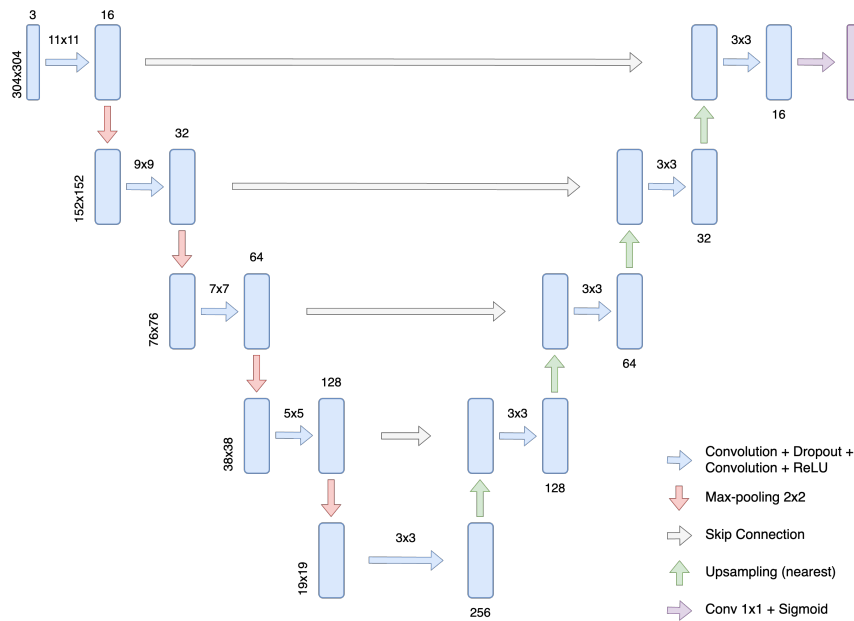


FIGURE 3.14 – Architecture de U-Net proposée pour la segmentation de la coupe transversale. La taille des convolutions est marquée au-dessus des flèches correspondantes.

après la première convolution. La [figure 3.14](#) synthétise la description de l'architecture. L'entraînement a été fait par la méthode ADAM, avec les paramètres par défaut et un coefficient d'apprentissage de  $10^{-4}$ . La fonction de perte est l'entropie croisée binaire.

Pour Mask R-CNN le réseau squelette est le réseau ResNet-101-FPN avec un pré-entraînement sur le jeu de données MS COCO. Les mêmes paramètres que pour le réseau U-Net précédemment présenté ont été choisis : ADAM avec paramètres par défaut, coefficient d'apprentissage de  $10^{-4}$  et l'entropie croisée binaire pour la fonction de perte.

Pour le réseau RefineNet aucune modification n'a été faite. L'entraînement s'est fait avec ADAM (paramètres par défaut) et un coefficient d'apprentissage de  $\alpha = 10^{-4}$ . Le nombre d'époques est de 40000.

Pour SegNet aucune modification n'a été faite. L'entraînement s'est fait avec la descente de gradient stochastique et un coefficient d'apprentissage de  $\alpha = 0,003$  sur un total de 30000 époques d'entraînement.

Une augmentation de données a été faite afin d'améliorer les performances des réseaux. Cette augmentation de données consiste en un certain nombre de déformations : des rotations, des décalages verticaux et horizontaux, des zooms et des transvections. L'augmentation se fait à la volée, c'est-à-dire qu'à chaque batch les images subissent ces transformations de manières aléatoires.

### 3.3.4 Résultats

Pour l'évaluation des méthodes proposées, nous comparons les valeurs du MCC (voir le [chapitre 1](#)). Cet indicateur est l'un des meilleurs car il prend en compte la répartition des différentes classes. Pour chaque jeu, chaque réseau a été entraîné et a fait l'objet d'une validation croisée en 2 groupes. La moitié des images servent d'entraînement, avec

MCC	U-Net	Mask-RCNN	RefineNet	SegNet	K-means	Contours Actifs
<b>BBF</b>	0.864	0.892	<b>0.958</b>	0.897	0.801	0.788
<b>Corcieux</b>	0.917	<b>0.930</b>	0.894	0.869	0.827	0.826
<b>Besle</b>	<b>0.911</b>	0.910	0.773	0.689	0.902	0.879
<b>Sawmill</b>	0.832	0.946	<b>0.961</b>	0.944	0.352	0.246
<b>sbgTS3</b>	0.916	0.784	<b>0.968</b>	0.909	0.668	0.698
<b>sbgTS12</b>	0.873	0.912	<b>0.960</b>	0.938	0.753	0.781

TABLE 3.3 – MCC pour les différents modèles sur les jeux de données utilisés. En caractères gras, le MCC le plus élevé.

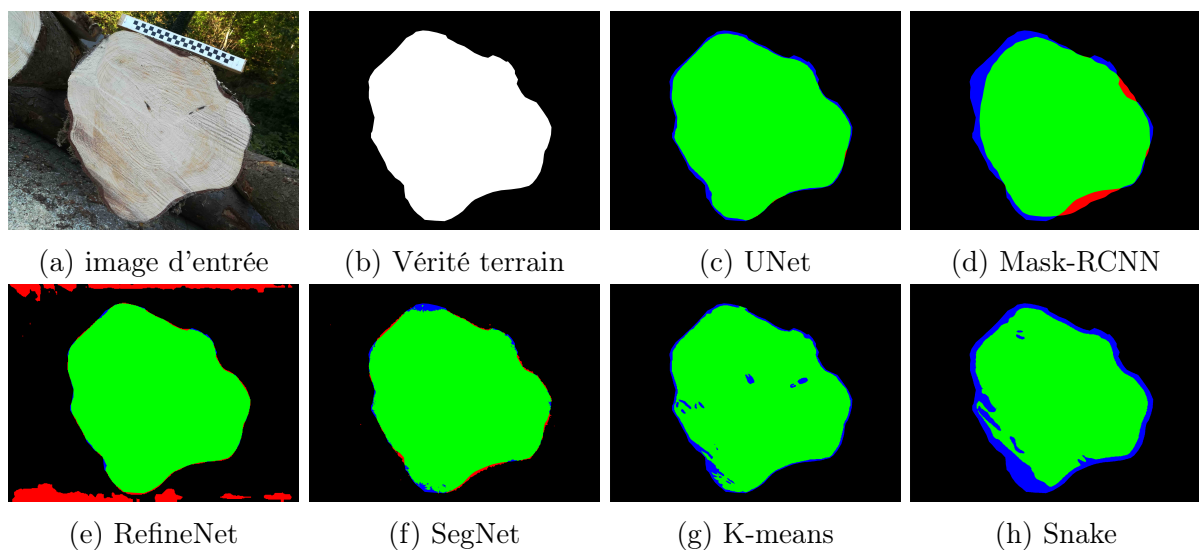


FIGURE 3.15 – Prédiction des modèles sur une image du jeu Corcieux. En vert, les pixels TP. En bleu, les pixels FN et en rouge les pixels FP.

60% pour l'apprentissage et 40% pour la validation, et l'autre moitié pour le jeu de test. Un deuxième entraînement est fait en inversant les jeux ; la moitié du jeu ayant servi de test devient la moitié d'entraînement (avec 60% pour l'apprentissage et 40% pour la validation) et la moitié ayant servi d'entraînement devient celle de test. Le MCC présenté est la moyenne obtenue pour chaque groupe. Ce score est calculé pour chaque jeu d'images suivant : BBF, Corcieux (Huawei), Besle, Sawmill ainsi que deux autres jeux de données (sbgTS3 et sbgT12) issus des données collectées par les collaborateurs autrichiens du projet TreeTrace. La [figure 3.15](#) montre les sorties des différentes méthodes pour une image issue du jeu *Corcieux*. De même, la [figure 3.16](#) montre les sorties des différentes méthodes pour une image issue du jeu *sbgTS3*.

Le [tableau 3.3](#) montre les scores MCC obtenus. Les valeurs en caractères gras sont le résultat le plus élevé, et donc le meilleur, pour chaque jeu d'images. Tout d'abord, le MCC indique que RefineNet est le plus performant pour les jeux *BBF*, *Sawmill*, *sbgTS12* et *sbgTS3*, U-Net est plus le plus performant sur le jeu *Besle* et Mask R-CNN est le plus approprié pour le jeu *Corcieux*. Les méthodes n'utilisant pas de réseaux donnent de bons

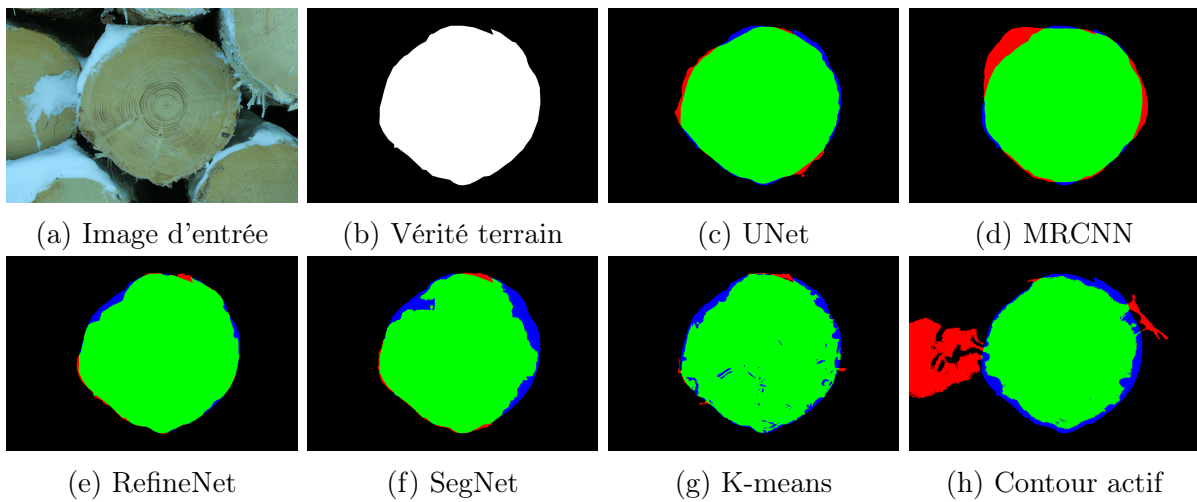


FIGURE 3.16 – Prédictions des modèles sur une image du jeu sbgTS3. En vert, les pixels TP. En bleu, les pixels FN et en rouge les pixels FP.

résultats pour les jeux *BBF*, *Corcieux* et *Besle*. Mais, elles ont des défauts notamment lorsqu'il y a d'autres grumes présentes dans l'image (voir [figure 3.15](#) et [figure 3.16](#)). De plus, elles sont moins performantes pour le jeu d'images *Sawmill*. Ces moins bonnes performances peuvent s'expliquer par plusieurs raisons. D'abord, la section transversale est petite dans ces images. Ce qui implique que la recherche de la plus grande composante circulaire, dans le cas de la méthode K-Means, n'est pas probante. De plus, les images sont peu contrastées et puisque ces deux méthodes utilisent l'espace de couleur **CieLAB** cela implique une moins bonne valeur de MCC. En effet, dans certaines images la présence d'autres grumes doit induire en erreur les méthodes de détection. Ceci est un inconvénient supplémentaire pour la méthode K-Means, et dans une moindre mesure la méthode des contours actifs. La [figure 3.17](#) illustre, à travers une image issue de l'ensemble, ces problèmes liés à ce jeu d'images. Dans l'image la présence d'une deuxième grume est visible, les composantes  $a^*$  et  $b^*$  de l'espace **CieLAB** ne permettent pas de démarquer suffisamment les grumes. Mais l'espace **CieLAB** semble être meilleur pour les autres jeux d'images. Il faudrait, pour ce jeu d'images, changer de représentation des couleurs.

Mask-RCNN est plus performant que U-Net dans tous les jeux à l'exception du jeu *sbgTS3*. U-Net a un score très faible sur le sous-ensemble *Sawmill*. Cela peut s'expliquer par le fait d'avoir utilisé l'entropie croisée binaire pour la fonction de perte, alors que les grumes sont très petites entraînant un déséquilibre dans les classes. Ce déséquilibre peut fausser l'entraînement du réseau. U-Net obtient de meilleurs résultats lorsque les coupes transversales sont plus grandes dans les images, comme dans le sous-ensemble *Besle*.

RefineNet donne en général les meilleurs résultats. Il surpasse les autres pour les jeux *sbgTS3* et *sbgTS12*. Néanmoins, lorsque le jeu de données est plus petit, RefineNet a du mal à fournir une segmentation précise. Cela peut expliquer les faibles résultats pour le jeu **Besle** composé de 65 images.

SegNet n'est jamais le meilleur réseau, mais il n'est jamais non plus le pire. Tout comme pour SegNet, le réseau a probablement trop de paramètres pour les petits jeux de données, notamment **Besle**, ce qui peut expliquer les faibles résultats pour ce jeu d'images.



FIGURE 3.17 – Image issue du jeu d’images *Sawmill* montrant les limites liées à l’image : présence d’une deuxième grume, en première ligne les composantes de l’espace CieLAB et en deuxième ligne les composantes de l’espace HSV.

U-Net	MRCNN	RefineNet	SegNet	K-means	Contour Actif
466	1245	1143	911	341	2052

TABLE 3.4 – Temps de calcul moyen (en ms) pour une image pour les différents modèles.

En plus du score MCC, les temps de calcul moyen pour une image à l’exécution ont été pris en compte. En effet, il est possible de sacrifier un peu de performance pour un gain de temps de calcul, car l’objectif est d’avoir un outil applicable dans un cadre industriel. Si une méthode possède un temps de calcul plus rapide, mais avec une faible perte de performance, par rapport à la méthode la plus performante, alors il est possible de considérer cette méthode. Le [tableau 3.4](#) montre le temps de calcul pour toutes les méthodes. Pour le benchmarking, nous avons utilisé 16GB de RAM à 2133 MHz (LPDDR3), un processeur Intel Core i7 et une carte graphique Intel Iris Plus Graphics 640 1536 Mo. Aucun GPU n’a été utilisé. La première chose que l’on remarque est que K-Means est la méthode la plus rapide. A contrario, le contour actif est le plus lent. Nous pouvons, au vu des résultats sur le MCC du contour actif, considérer que ce n’est pas la méthode offrant les meilleurs résultats. U-Net a un temps d’exécution (466 ms) à peine plus important que K-Means (341 ms), mais avec un MCC plus élevé. U-Net semble donc être une bonne approche. MRCNN est le réseau de neurones le plus lent avec une exécution en 1245 ms. Les résultats de ce réseau sont très légèrement meilleurs que U-Net. Enfin, bien que SegNet ait un temps d’exécution inférieur à une seconde, ses performances sont les moins bonnes des réseaux. Pour à peine 200 ms de plus, le choix de RefineNet est plus pertinent.

Cela étant dit, il est nécessaire de souligner quelques points. Tout d’abord, l’utilisation de l’entropie croisée binaire peut induire des erreurs au niveau de l’apprentissage des réseaux de neurones. Il est envisageable de refaire un entraînement, mais avec la fonction de perte MCC (voir l’équation 1.12). Autre point à noter, certains jeux ont très peu d’images.

Or, certains réseaux ont beaucoup de paramètres nécessitant soit un jeu d'apprentissage très grand, soit d'effectuer un *transfer learning*, soit d'apprendre une partie de manière non supervisée, avec les *Barlow Twins* par exemple [Zbontar et al., 2021]. Nous n'avons pas testé cette méthode, ou d'autres réseaux de neurones plus récents car les travaux sur la segmentation de la grume ont été en 2020. Il est envisageable de faire apprendre sur des réseaux de neurones pour la segmentation de la grume. Bien que l'augmentation de données, à travers des transformations des images, ait été faite, il est possible que cela ait un effet sur l'apprentissage des réseaux. Notamment MRCNN est un réseau qui a été conçu pour la segmentation sémantique multi-classe. Pourtant, il a été utilisé pour une segmentation avec une seule classe. Cet usage peut limiter ses performances.

Il serait possible de concevoir des contours actifs plus adaptés à la segmentation de la grume. Par exemple, il serait pertinent d'inclure des critères de régularité supplémentaires afin de prendre en compte la forme très régulière des grumes (dans la grande majorité des cas). Toutefois, le temps de calcul, déjà important par rapport aux autres méthodes, du contour actif n'incite pas à approfondir cette méthode dans un cadre applicatif. Pour la méthode K-Means, il faudrait adapter la méthode de sélection afin de pallier au problème de la taille de la grume. Effectuer des pré-traitements et des post-traitements avant de faire la sélection de la plus grande composante circulaire pourrait augmenter les performances. Par exemple, l'utilisation de morphologie mathématique permettrait d'avoir des composantes plus circulaires. L'utilisation de la composante  $S$  de l'espace HSV peut être, dans certains cas, plus adéquate.

Dans la suite du manuscrit, afin de ne pas induire davantage d'erreurs dans les autres méthodes, nous avons décidé de prendre la segmentation manuelle de la coupe transversale et non l'une des segmentations présentées ci-dessus.

### 3.4 Segmentation de l'aubier et du duramen

Une fois la segmentation de la grume effectuée, l'étude des autres éléments clefs dans l'estimation de la qualité du bois devient plus aisée. Parmi ces éléments, la quantité de duramen par rapport à la quantité d'aubier est un indicateur de qualité du bois. Afin d'éviter d'induire davantage d'erreurs dans la segmentation du duramen, les images qui sont utilisées dans cette section sont celles des coupes transversales segmentées manuellement.

Nous avons présenté, dans le chapitre précédent, la seule étude [Raatevaara et al., 2020], à notre connaissance, portant sur la segmentation du duramen sur des images RVB brutes de coupes transversales de grumes. L'étude traite des images de même nature que les nôtres. Cependant, les résultats de l'étude sont très disparates : l'indice de Jaccard allant de 0.3 à 0.9 [Raatevaara et al., 2020]. Nous proposons d'effectuer la segmentation du duramen à l'aide de réseaux de neurones convolutifs avec peu de paramètres.

Toute cette section a fait l'objet d'un article qui a été soumis (voir section 6.2). Le code source est disponible <sup>10</sup>.

---

10. <https://gitlab.com/Ryukhaan/treetrace>

### 3.4.1 Réseaux de neurones convolutifs

La deuxième approche pour effectuer la segmentation du duramen est l'utilisation de réseaux de neurones ayant peu de paramètres. En effet, nous avons souligné quelques défauts des réseaux pour la segmentation de la grume. Notamment, nous avons utilisé des réseaux de neurones avec beaucoup de paramètres alors que nos jeux d'images sont restreints. Autre point, le Douglas possède un duramen très coloré (rose/rouge) dans la majorité des cas. Étant donné qu'il s'agit d'identifier ces pixels colorés, il semble pertinent de réduire les nombres de paramètres des réseaux de neurones.

Une opération que l'on peut faire pour diminuer le nombre de paramètres est de remplacer les convolutions d'un CNN par des convolutions séparables en profondeur, ou en anglais *depthwise separable convolutions* (DSC) [Mamalet and Garcia, 2012]. L'idée d'une convolution séparable est de remplacer la matrice de convolution par le produit de deux vecteurs. Par exemple, une matrice de taille  $5 \times 5$ , soit 25 paramètres, peut être remplacée par le produit de deux vecteurs de taille 5, soit 10 paramètres. La notion de profondeur permet d'éviter de multiplier le nombre de paramètres de la convolution par le nombre de composantes *couleurs*, il s'agit d'une addition pondérée des résultats de la convolution sur chaque composante. Avec une image RVB, soit 3 composantes, une convolution classique avec un filtre de taille  $5 \times 5$  donne 75 paramètres. Avec une couche DSC, nous avons  $5 + 5 + 3 = 13$  paramètres.

Malheureusement, réduire le nombre de paramètres va avoir une conséquence sur l'apprentissage. Celui-ci sera plus long et la possibilité de tomber dans un minimum local est plus importante. Afin de compenser cet effet, nous proposons d'ajouter des couches d'attention, avec peu de paramètres, définies dans les sous-sections suivantes.

#### 3.4.1.1 Couche d'attention morphologique

Nous proposons une nouvelle couche d'attention. Nous allons d'abord rappeler l'objectif des couches d'attention. Ces couches utilisent des opérateurs qui mettent en évidence les caractéristiques importantes. L'idée étant de diminuer les informations peu utiles pour la segmentation et d'augmenter l'importance des informations pertinentes. Les outils de morphologie mathématique (MM) permettent d'effectuer opérations spécifiques aux images pour récupérer ou filtrer différentes structures. Ils utilisés dans de nombreuses tâches de vision par ordinateur et/ou traitement d'images, telles que le filtrage, la segmentation, l'extraction de caractéristiques, etc. C'est la raison pour laquelle nous avons introduit une couche d'attention qui utilise les outils de MM.

Les opérations de base de la MM sont la dilatation et l'érosion. D'autres filtrages morphologiques peuvent être définis par une combinaison de ces deux opérateurs. Dans ce travail, nous empruntons les couches morphologiques introduites dans [Mondal et al., 2019].

Les couches introduites sont présentées de la manière suivante. Soit  $I$  une image avec une seule composante. Les opérations de dilatation  $\oplus$  et d'érosion  $\ominus$  sur un pixel  $(i, j)$  sont définies comme :



$$(I \oplus \mathbf{W}_d)(i, j) = \max_{x \in U, y \in V} (I(i - x, j - y) + \mathbf{W}_d(x, y)) \quad (3.2)$$

$$(I \ominus \mathbf{W}_e)(i, j) = \min_{x \in U, y \in V} (I(i - x, j - y) + \mathbf{W}_e(x, y)) \quad (3.3)$$

où  $\mathbf{W}_d \in \mathbb{R}^{a \times b}$ ,  $\mathbf{W}_e \in \mathbb{R}^{a \times b}$ ,  $U = \{1, 2, \dots, a\}$ ,  $V = \{1, 2, \dots, b\}$  et  $a, b \in \mathbb{N}$ . Les deux matrices  $\mathbf{W}_d$  et  $\mathbf{W}_e$  sont les éléments structurants (matrice) de la dilatation et de l'érosion et sont de taille  $a \times b$ .

La couche d'attention proposée (MAG) va utiliser ces couches morphologique. Elle se focalise sur l'information spatiale et non sur l'information des composantes, c'est-à-dire sur les composantes *couleur*. Redisons-le, le duramen pour le Douglas est dans les teintes rouge/rose. Ainsi, d'une image à l'autre nous cherchons à segmenter cette partie de la grume. Ainsi, se focaliser sur les composantes ne semble pas très pertinent, car cela n'apportera pas davantage d'informations.

La couche d'attention morphologique (MAG) que nous proposons est la suivante. Soit la carte de caractéristiques d'entrée  $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$ , où  $H$ ,  $W$  et  $C$  sont des entiers. Notre couche effectue un premier calcul spatial dont la carte 2D  $\mathbf{F}' \in \mathbb{R}^{H \times W \times 1}$  en est le résultat. Cette carte est donnée par la formule :

$$F' = W_s * F$$

où  $W_s$  contient les poids d'une convolution  $1 \times 1$ .

Pour la suite, nous avons considéré deux chemins à l'intérieur de la couche. Le premier chemin utilise  $k \in \mathbb{N}$  couches de dilatation, et le second chemin utilise  $k$  couches d'érosion. Nous avons introduit une séquence d'érosion (ou de dilatation) avec différents éléments structurants afin d'éliminer le bruit ou d'améliorer l'information. Cette séquence est utile, car une seule opération n'est pas nécessairement suffisante pour effectuer les tâches voulues (réduction du bruit et augmentation de l'information pertinente).

Le chemin de dilatations se résume de la manière suivante :

$$\forall i \in \llbracket 0, \dots, k - 1 \rrbracket, \quad \begin{cases} \mathbf{D}^0 = & \mathbf{F}' \\ \mathbf{D}^i = & (\mathbf{D}^{i-1} \oplus \mathbf{W}_d^i) + \mathbf{F}' \end{cases} \quad (3.4)$$

De la même façon, le chemin d'érosions s'écrit :

$$\forall i \in \llbracket 0, \dots, k - 1 \rrbracket, \quad \begin{cases} \mathbf{E}^0 = & \mathbf{F}' \\ \mathbf{E}^i = & (\mathbf{E}^{i-1} \ominus \mathbf{W}_e^i) + \mathbf{F}' \end{cases} \quad (3.5)$$

Comme nous ne pouvons pas savoir quel chemin est le plus efficace pour la suppression du bruit dans un cas particulier, nous combinons les deux sorties en une seule carte de caractéristiques en utilisant une addition pixel par pixel suivie d'une fonction d'activation (sigmoïde)  $\sigma$ . Le résultat est une carte 2D  $\alpha$ . Ensuite, cette dernière est multipliée pixel par pixel ( $\odot$ ) par la carte d'entrée  $\mathbf{F}$  canal par canal :

$$\mathbf{F}'' = \underbrace{\sigma(\mathbf{D}^{k-1} + \mathbf{E}^{k-1})}_{\alpha} \odot \mathbf{F}$$

La [figure 3.18](#) résume la couche d'attention morphologique présentée ci-dessus.

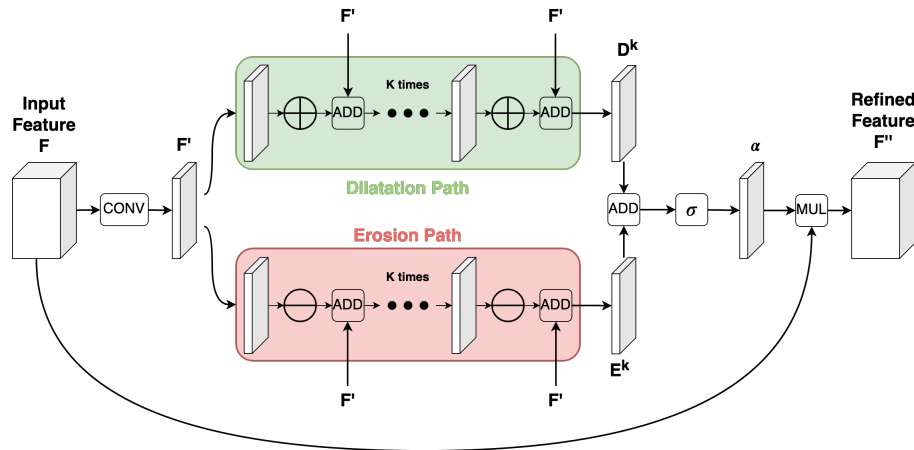


FIGURE 3.18 – Couche d’attention morphologique proposée.

### 3.4.1.2 Architecture proposée

L’architecture proposée afin de réduire les paramètres est une architecture similaire à U-Net. Mais, au lieu de procéder deux fois à des convolutions (à chaque niveau du réseau), nous décidons de ne faire qu’une seule convolution. Cela permet de diminuer de moitié le nombre de paramètres. De plus, chaque convolution est remplacée par une DSC. Comme nous l’avons vu, ce type de convolution permet de réduire le nombre de paramètres. Les fonctions d’activations ReLU sont remplacées par Leaky ReLU. Nous ajoutons à chaque convolution une couche de *shakeout*. La description de cette architecture est illustrée dans la [figure 3.19](#). Ce réseau est intitulé LU-Net (pour *Light U-Net*).

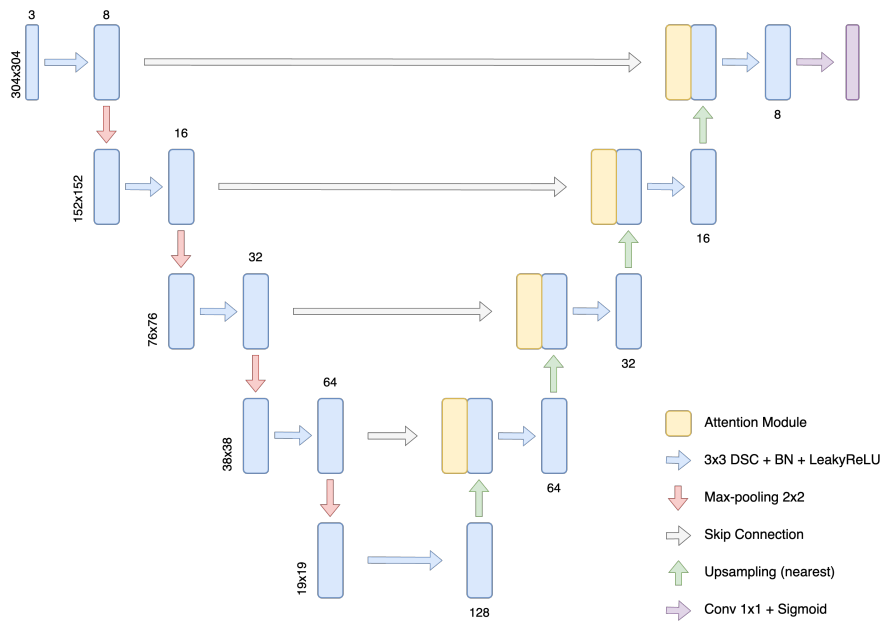


FIGURE 3.19 – Architecture du réseau LU-Net proposée.

### 3.4.2 Résultats

En plus du réseau LU-NET tel que nous l'avons présenté, nous avons entraîné d'autres CNN pour comparer. Notamment, nous avons entraîné le modèle U-Net sans modifications et le modèle LU-Net sans couches d'attention. Nous avons entraîné LU-Net avec la couche d'attention proposée (MAG). Nous avons comparé notre couche d'attention avec deux autres couches d'attention : CBAM et AAG (voir [chapitre 1](#)). Tous les réseaux ont 8 filtres pour la première convolution.

Tous les modèles ont été entraînés sur 100 époques. La taille d'entrée a été fixée à  $304 \times 304$  pixels (interpolation bilinéaire pour le sous-échantillonnage) et les images sont restées dans l'espace RVB. Nos jeux de données étant réduits, nous avons appliqué l'augmentation de données. Les transformations appliquées ont été les mêmes que pour la segmentation de la grume : rotation, transvection, décalage horizontal ou vertical et le zoom. Le taux d'apprentissage initial a été fixé à 0.001. Nous avons utilisé la méthode Adam pour l'apprentissage avec les paramètres suivants  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  et  $\epsilon = 10^{-7}$ . Les paramètres des couches de *shakeout* ont été fixés à  $\tau = 0.1$  et  $c = 0.1$ . La fonction de coût choisie a été le MCC (voir [sous-section 1.7.3](#)) pour tous les réseaux.

L'apprentissage s'est fait sur une validation croisée avec 8 groupes. Les meilleurs résultats ont été obtenus avec  $k = 3$ , le nombre d'érosions/dilatations effectué (voir [équation 3.4](#)), et un élément structurant de taille  $7 \times 7$  aussi bien pour le chemin de dilatations que le chemin d'érosions. La comparaison entre modèles s'est faite en utilisant le MCC.

Contrairement à la segmentation de la grume, il ne nous est pas possible d'utiliser les jeux de données avec de l'épicéa (voir la [figure 3.1](#) ou la [figure 3.3](#)). Cette essence est un bois blanc, c'est-à-dire que le duramen n'est pas visible à l'œil nu, contrairement au Douglas. Par ailleurs, les deux jeux d'images *Besle* et *BBF* ayant très peu d'images ont été exclus de l'apprentissage. Les deux jeux d'images qui ont été utilisés sont le jeu *Sawmill* et *Logyard*. Chaque jeu d'images est traité séparément, afin de prendre en compte la spécificité des jeux. Rappelons également que les images d'entrée ne sont pas les images brutes, mais les images préalablement segmentées manuellement et recadrées. Cela permet de réduire l'influence d'une potentielle mauvaise segmentation de la coupe transversale.

Les résultats des différentes validations croisées pour les deux jeux d'images et chaque réseau sont dans la [tableau 3.5](#). Quelques exemples des contours de la segmentation, pour les deux jeux d'images, sont présentés dans la [figure 3.20](#). Pour le jeu de données *Logyard*, le LU-Net est moins précis que U-Net. Cependant, lorsque des couches d'attention sont ajoutées, le LU-Net montre de meilleurs résultats. U-Net est plus stable que les autres modèles, il fournit l'écart-type le plus faible. Notre couche d'attention donne le meilleur score moyen et a les meilleures performances sur 3 des validations croisées.

Pour le jeu de données *Sawmill*, nous observons que U-Net donne de faibles performances (voir [figure 3.20](#)). LU-Net est plus performant que U-Net. Une explication possible de ces résultats est que les images sont très peu contrastées (voir [figure 3.20](#)). De plus, dans les images RVB, chaque canal est corrélé à l'autre. Par conséquent, les filtres de convolution ne parviennent pas à mettre en évidence le duramen. Au lieu de cela, les DSCs appliquent un filtre de convolution sur chaque canal puis les fusionnent. Ils agissent comme une déconvolution des couleurs [[Ruifrok et al., 2001](#)].

Tout comme pour la détection de la grume, nous avons calculé le temps moyen d'exé-

	Modèle	#	Taille relative	F1	F2	F3	F4	F5	F6	F7	F8	Moyenne	Écart-type
<i>logyard</i>	U-Net	378,321	1 ×	0.931	0.911	0.932	<b>0.926</b>	0.915	0.900	0.933	<b>0.925</b>	0.921	<b>0.011</b>
	LU-Net	33,428	0.08 ×	0.883	0.906	0.926	0.881	0.927	0.926	0.888	0.920	0.907	0.019
	LU-Net+AAG	99,672	0.26 ×	0.903	<b>0.930</b>	0.939	0.911	0.906	0.942	<b>0.936</b>	0.911	0.922	0.015
	LU-Net+CBAM	44,940	0.12 ×	0.897	0.913	<b>0.949</b>	0.915	0.899	0.941	0.904	0.916	0.917	0.018
	LU-Net+MAG	34,332	0.09 ×	<b>0.936</b>	0.923	0.924	0.925	<b>0.939</b>	<b>0.952</b>	0.934	0.907	<b>0.930</b>	0.013
<i>sauvill</i>	U-Net	378,321	1 ×	0.892	0.882	0.907	0.943	0.790	0.699	0.888	0.894	0.862	0.074
	LU-Net	33,428	0.08 ×	0.932	0.928	<b>0.926</b>	0.931	0.890	0.931	0.929	0.939	0.926	0.014
	LU-Net+AAG	99,672	0.26 ×	0.934	0.925	0.920	0.929	<b>0.931</b>	0.917	0.941	0.938	0.929	<b>0.008</b>
	LU-Net+CBAM	44,940	0.12 ×	0.932	<b>0.932</b>	0.916	0.931	0.928	<b>0.932</b>	<b>0.945</b>	0.920	0.930	<b>0.008</b>
	LU-Net+MAG	34,332	0.09 ×	<b>0.941</b>	0.925	0.921	<b>0.946</b>	0.926	0.928	0.942	<b>0.940</b>	<b>0.934</b>	0.009

TABLE 3.5 – Valeur du MCC sur une validation croisée avec 8 groupes pour chaque modèle et les deux jeux d’images utilisés. La deuxième colonne (#) est le nombre de paramètres du réseau.

U-Net	LU-Net	LU-Net+AAG	LU-Net+CBAM	LU-Net+MAG
88	67	76	97	89

TABLE 3.6 – Temps de calcul moyen pour une image pour les différents modèles en ms.

cution pour image (hors temps d’apprentissage). En effet, les couches de convolutions ajoutent des paramètres, mais également du temps de calcul. Ainsi, il est nécessaire de savoir si les bons résultats de notre couche d’attention ne se font pas au détriment d’un temps de calcul trop élevé.

Le [tableau 3.6](#) montre le temps de calcul pour toutes les méthodes. Pour le benchmarking, nous avons utilisé 16GB de RAM à 2133 MHz (LPDDR3), un processeur Intel Core i7 et une carte graphique Intel Iris Plus Graphics 640 1536 Mo. Aucun GPU n’a été utilisé.

La première chose que l’on remarque, c’est que la version allégée de U-Net (LU-Net) est plus rapide. Cela est normal puisque que le nombre de paramètres a diminué. A contrario, le temps de calcul augmente lorsqu’une couche d’attention est ajoutée. C’est le réseau LU-Net avec la couche d’attention CBAM qui prend le plus de temps d’exécution. Notre couche d’attention a le même temps que la version classique de U-Net. Au final, en prenant en compte les résultats précédents, notre couche d’attention offre de meilleurs résultats, pour un ajout minime de paramètres et un temps de calcul très faiblement augmenté.

## 3.5 Détection des cernes

Nous venons de segmenter le duramen et nous avons segmenté la grume, nous pouvons en déduire la zone d’aubier. La zone d’aubier et de duramen ne sont pas les seuls éléments utiles pour déterminer la qualité du bois. Les informations concernant les **cernes** jouent un rôle dans la qualité du bois. Nous avons vu dans le [chapitre 2](#) que la détection des cernes a été très étudiée. Particulièrement, de nombreuses méthodes ont été réalisées dans le cadre de la [dendrochronologie](#). Or, soulignons-le encore une fois, le projet TreeTrace se place dans un cadre industriel et avec des images capturées, en scierie ou en forêt, à partir de surfaces brutes non préparées. Aucun traitement, physique ou chimique, ne peut être

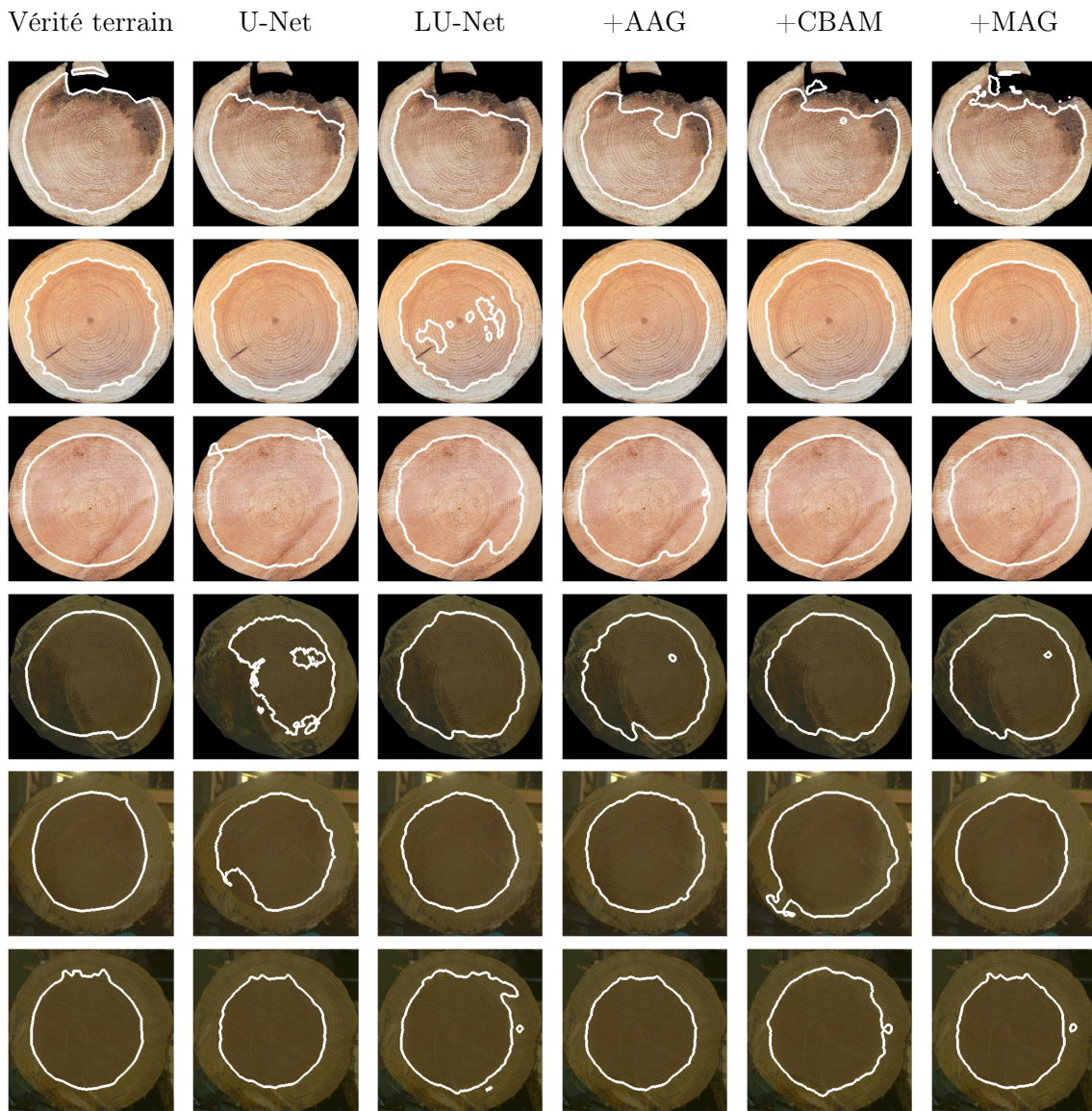


FIGURE 3.20 – Contour de la segmentation du duramen par les différents réseaux de neurones proposés.

réalisé à la différence de la dendrochronologie. Cela implique, comme nous l'avons montré dans les différents jeux d'images acquis (voir [section 3.2](#)), que les images ne mettent pas nécessairement en valeur les cernes. La présence de coups de scies, ou d'arrachements de bois sont encore très présents dans les images rendant l'analyse sur ce type d'image plus difficile (voir [figure 3.21](#)). Avant de procéder à tester dans des conditions réelles nous abordons la détection des cernes dans un cadre plus idéal pour développer de nouvelles approches. C'est la raison pour laquelle, dans la suite de cette section, nous utiliserons les images de grumes segmentées manuellement (par un expert).

Dans la suite de cette section, nous allons détailler la méthode proposée afin de seg-



FIGURE 3.21 – Zoom d’images montrant les difficultés liées au type d’images acquises.

menter les cernes. Notre méthode se décompose en trois parties : un filtrage, un seuillage et un affinement de l’image seuillée.

### 3.5.1 Filtrage directionnel

Avant tout chose, les premières expérimentations qui ont été menées ont été faites sur des images segmentées manuellement. L’idée derrière ça est de simplifier, dans un premier temps, le cadre de travail afin de construire un outil de détection des cernes pour ce type d’images. En effet, la segmentation de la grume pourrait donner des images où les cernes visibles seraient manquantes. Une position de la moelle légèrement décalée pourraient entraîner un filtrage directionnel non opérationnel. Cette segmentation est un cadre plus idéal pour tester un nouvel outil. D’autant plus que les images possèdent déjà beaucoup d’informations qui ne sont pas nécessaires à l’analyse, parfois même peuvent gêner cette analyse.

En effet, la présence de coups de scie, d’arrachements de bois et tout autre forme de bruit, posent un problème dans l’analyse de cernes. La [figure 3.21](#) illustre ce problème. Afin de réduire les erreurs induites par ces défauts, nous proposons d’effectuer un filtrage. Afin d’être le meilleur possible, le filtrage doit se faire dans la direction des tangentes aux cernes. Cela devrait permettre de réduire le bruit tout en rehaussant les limites de cernes. C’est pourquoi, nous proposons d’utiliser des filtres orientables. L’objectif d’un tel filtre est d’appliquer le même filtre mais avec des orientations différentes selon l’orientation des tangentes aux cernes.

**Dans la suite du manuscrit, nous parlerons de l’orientation des cernes pour désigner l’orientation des tangentes aux cernes.**

Le lissage proposé est celui du filtre gaussien orientable [[Freeman et al., 1991](#)] (*steerable filters*). La fonction gaussienne en 2D d’écart-type  $\sigma$  s’écrit :

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.6)$$

l’écart-type est fixé à 1 pour des raisons de simplicité. Notons  $G^\theta$  la fonction gaussienne 2D dépendant d’un angle  $\theta$  donnée et notons  $G_n$  la dérivée  $n$ -ième de la fonction  $G$ . Nous

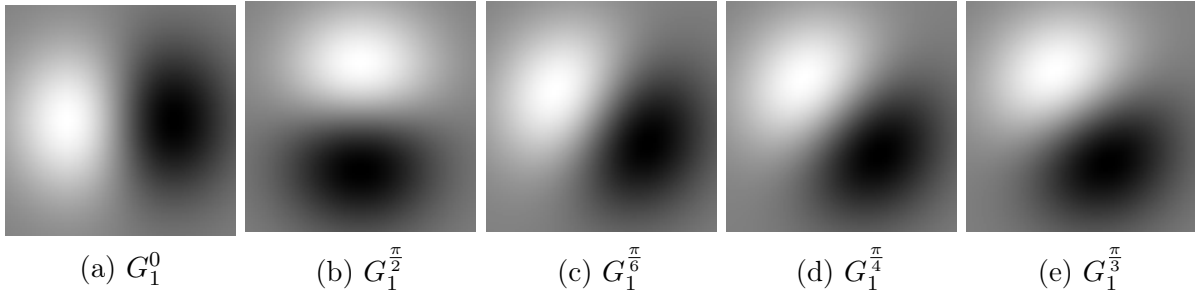


FIGURE 3.22 – Illustration des masques pour un filtre dérivatif gaussien orientable.

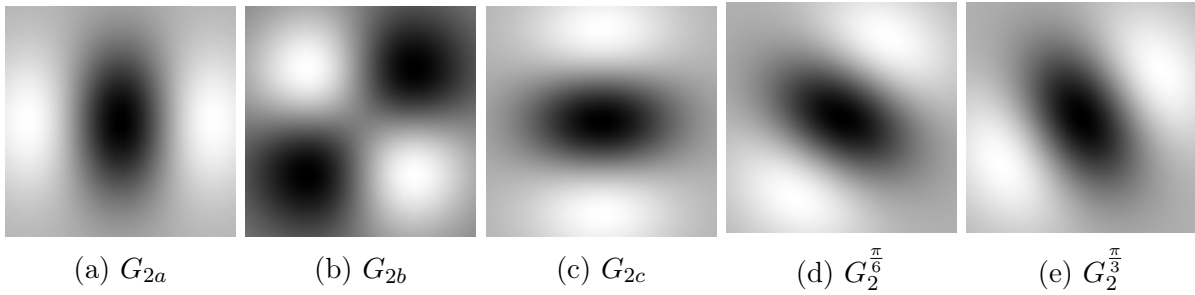


FIGURE 3.23 – Illustration de masques pour un filtre orientable de la dérivée seconde d'une gaussienne.

avons :

$$G_1^0 = \frac{\partial}{\partial x}G(x, y) = -2xG(x, y) \quad (3.7)$$

$$G_1^{\frac{\pi}{2}} = \frac{\partial}{\partial y}G(x, y) = -2yG(x, y) \quad (3.8)$$

La [figure 3.22](#) montre deux masques de taille  $100 \times 100$  pixels représentant la fonction gaussienne 2D orientable. Il est facile de montrer que pour un angle  $\theta$  quelconque, la fonction  $G_1^\theta$  est une combinaison linéaire des deux fonctions ci-dessus (voir l'[équation 3.7](#)). Ainsi,

$$G_1^\theta = \cos(\theta)G_1^0 + \sin(\theta)G_1^{\frac{\pi}{2}} \quad (3.9)$$

En utilisant la linéarité de l'opérateur de convolution, il est possible d'effectuer un lissage et une détection de contour directionnel :

$$I * G_1^\theta = \cos(\theta)(I * G_1^0) + \sin(\theta)(I * G_1^{\frac{\pi}{2}}) \quad (3.10)$$

Il est possible de généraliser cela à d'autres fonctions. C'est le cas de la dérivée seconde de la fonction gaussienne. Pour un angle  $\theta$  le filtre orientable s'écrit :

$$G_2^\theta = k_a(\theta)G_{2a} + k_b(\theta)G_{2b} + k_c(\theta)G_{2c} \quad (3.11)$$

avec

$$k_a(\theta) = \cos^2(\theta), \quad k_b(\theta) = -2 \cos(\theta) \sin(\theta), \quad k_c(\theta) = \sin^2(\theta)$$

et

$$G_{2a} = (2x^2 - 1)e^{-(x^2+y^2)}, \quad G_{2b} = 2xye^{-(x^2+y^2)}, \quad G_{2c} = (2y^2 - 1)e^{-(x^2+y^2)}$$

Les fonctions  $G_{2a}$ ,  $G_{2b}$  et  $G_{2c}$  sont illustrées dans la [figure 3.23](#) ainsi que deux exemples de la fonction  $G_2^\theta$  pour les angles  $\frac{\pi}{6}$  et  $\frac{\pi}{3}$ . De la même façon que précédemment, la linéarité de l'opérateur de convolution permet de convoluer l'image de manière efficace.

### 3.5.2 Filtrage directionnel : application au bois

Le filtre présenté ci-dessus est appliqué pour rehausser les cernes et réduire le bruit intracerne. En effet, un cerne est un changement de transition haute intensité (bois de printemps), faible intensité (bois d'été), puis de nouveau de haute intensité (bois de printemps de l'année suivante). La dérivée seconde de la gaussienne approxime ces transitions.

Cependant, le filtrage par une dérivée seconde de la gaussienne risque de ne pas préserver les contours. Cela va poser un soucis dans la détection des cernes, puisque ces derniers sont des contours : transition de faible intensité vers une forte intensité. Pour corriger cela, nous nous inspirons des filtres bilatéraux [[Tomasi and Manduchi, 1998](#), [Paris et al., 2009](#)]. Un filtre bilatéral est un filtre qui suggèrent de pondérer la valeur d'un pixel par rapport à des distances à ses voisins : d'un point de vue spatiale et d'un point de vue colorimétrique (distance entre les couleurs ou niveaux de gris). Des recherches ont été faites pour faire en sorte que les filtres bilatéraux soient orientables [[Hu and Li, 2012](#), [Le et al., 2014](#), [Venkatesh and Seelamantula, 2015](#)]. Nous proposons ici notre propre filtre bilatéral directionnel.

La valeur d'une image  $I$  au pixel  $\mathbf{p}$  après application d'un filtre bilatéral s'exprime de la manière suivante :

$$I^{\text{filtrée}}(\mathbf{p}) = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{V}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I(\mathbf{p}) - I(\mathbf{q})|) I(\mathbf{q}) \quad (3.12)$$

$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{V}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I(\mathbf{p}) - I(\mathbf{q})|) \quad (3.13)$$

où  $\mathcal{V}$  désigne un voisinage du pixel  $\mathbf{p}$  (souvent un masque carré de taille fixé),  $\sigma_s$  désigne l'écart-type de la gaussienne spatiale et  $\sigma_r$  désigne l'écart-type de la gaussienne colorimétrique (distance en terme de couleurs). Nous décidons de changer la formulation ci-dessus en remplaçant le filtre gaussien spatial par le filtre directionnel de la dérivée seconde d'une gaussienne 2D (voir l'[équation 3.11](#)). Autrement dit,  $G_{\sigma_s}$  est remplacé par :

$$G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) = k_a(\theta(\mathbf{p}))G_{2a}(\|\mathbf{p} - \mathbf{q}\|) + k_b(\theta(\mathbf{p}))G_{2b}(\|\mathbf{p} - \mathbf{q}\|) + k_c(\theta(\mathbf{p}))G_{2c}(\|\mathbf{p} - \mathbf{q}\|) \quad (3.14)$$

la pondération  $W_{\mathbf{p}}$  est mise à jour selon la modification apportée et  $\theta(\mathbf{p})$  est l'orientation au pixel  $\mathbf{p}$ .

Pour appliquer le filtre, il est nécessaire de connaître en chaque point l'orientation  $\theta$ . Nous pourrions calculer l'orientation à l'aide d'outil spécialement conçu pour cela, cependant le bruit intracerne risque de fausser le calcul (voir le [chapitre 4](#)). Nous avons décidé de choisir une orientation théorique. Notre hypothèse est la suivante : on suppose



que les cernes décrivent des cercles concentriques dont la moelle est le centre (pour le calcul de la moelle voir le [chapitre 4](#)). Le calcul de l'orientation au point  $\mathbf{p} = (x, y)$  vaut :

$$\theta(x, y) = \arctan\left(\frac{y}{x}\right) \quad (3.15)$$

où les coordonnées  $x$  et  $y$  sont exprimées dans le repère dont la moelle est l'origine.

Une fois l'orientation  $\theta$  calculée en tout point de l'image, nous pouvons appliquer le filtre bilatéral orienté. L'image résultante est le résultat de la convolution par le filtre appliqué sur l'image convertie en niveaux de gris.

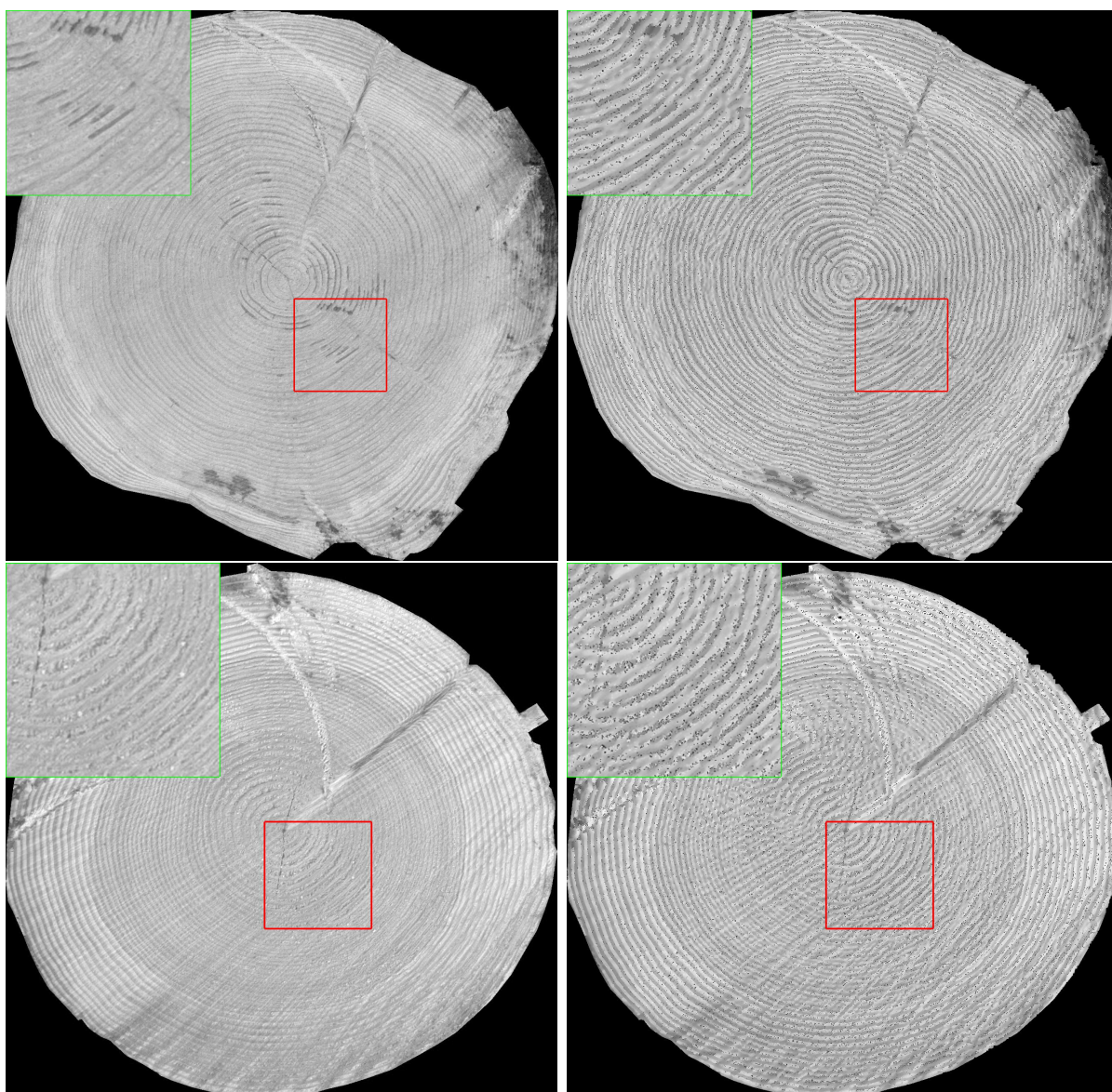


FIGURE 3.24 – Deux exemples du filtrage bilatéral directionnel. À gauche, l'image en niveaux de gris. À droite, l'image filtrée. En haut à gauche de chaque image, un zoom du rectangle rouge.

La [figure 3.24](#) illustre le résultat du filtre bilatéral directionnel proposé avec  $\sigma_s = 4.0$ ,  $\sigma_r = 15.0$  et la taille du masque de convolution fixé à 21. Un zoom du rectangle rouge a été fait et est mis en haut à gauche de l'image. Nous pouvons voir que le lissage a bien été fait, que les cernes sont rehaussés. Cependant, un effet de bord est visible. Des artefacts, semblable à un bruit impulsif, apparaissent après l'application du filtre. Pour le moment, nous ne savons pas encore la raison de leur apparition. Nous avons remarqué que cela dépendait principalement de la valeur de  $\sigma_s$  (le filtrage spatial). Notre explication actuelle est que le bruit est dû à l'utilisation de la dérivée seconde de la gaussienne. Malgré tout, ce bruit est moins gênant, car il a été étudié plus largement, et des solutions ont été proposées pour le supprimer [[Chan et al., 2005](#), [Gonzalez-Hidalgo et al., 2018](#)]. Ceci étant dit, nous proposons, par la suite, d'effectuer un seuillage adapté afin de ne pas tenir compte du bruit (voir la [sous-section 3.5.3](#)). C'est-à-dire que le seuillage va être en adéquation avec le fait qu'il y ait potentiellement un bruit poivre et sel dans l'image.

### 3.5.3 Seuillage

Une fois l'image filtrée, nous seuillons l'image. Pour le seuillage, nous avons choisi un seuillage adaptatif. Le seuillage adaptatif est une méthode dans laquelle la valeur du seuil est calculée dans un voisinage local du pixel et, par conséquent, la valeur sera différente pour chaque pixel. L'utilisation d'un filtrage adaptatif et local est pertinent, car il permet de prendre en compte d'éventuels changements d'intensité présents dans l'image.

Le seuillage est celui utilisant la moyenne [[Glasbey, 1993](#)]. Ce seuillage a été choisi, car il va nous permettre de ne pas tenir compte du bruit poivre et sel présent à cause du filtre bilatéral directionnel. La moyenne est calculée en un voisinage carré  $\omega \times \omega$ , avec  $\omega \in \mathbb{N}$ . Si l'intensité du pixel est supérieure à la moyenne moins une constante  $C \in \mathbb{R}$ , alors le pixel est mis à 1 sinon il est mis à 0.

En appliquant le seuillage sur l'image filtrée, avec  $C = -4$  et  $\omega = 31$ , nous obtenons les images illustrées dans la [figure 3.25](#).

Nous pouvons remarquer que certaines composantes connexes sont de très petites taille. Il est impossible de savoir s'il s'agit de cernes ou du bruit. Nous avons choisi de supprimer celles ayant une taille inférieure à un seuil  $M$ . La suppression des petites composantes connexe est faite en fixant  $M$  à 100. À droite de la [figure 3.25](#), nous observons l'effet de la suppression des petites composantes connexes.

### 3.5.4 Affinement du seuillage

Notre objectif est de trouver la limite des cernes, nous devons amincir les zones qui ont été seuillées précédemment. Pour se faire, nous utilisons l'information suivante : lorsque l'on se déplace de la moelle vers l'écorce, le passage d'un cerne au suivant se traduit par une transition de faible intensité (pixel noir) vers une haute intensité (pixel blanc). Le but est de trouver ces transitions. Pour cela, des rayons partant de la moelle vers l'écorce, c'est-à-dire le bord de l'image, sont tracés. Puis, le long de chaque rayon, s'il y a un transition blanc vers noir entre un pixel et le suivant, alors le pixel reste à 1. Sinon le pixel est mis à 0. Afin d'avoir connecter les limites cernes, nous appliquons une fermeture

morphologique (dilatation suivie d'une érosion). L'élément structurant choisi a été un élément structurant simple :

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

La [figure 3.26](#) montre les résultats de cet affinement. La [figure 3.27](#) et la [figure 3.28](#) montrent deux exemples des limites de cernes segmentées. Nous pouvons observer que la segmentation des limites de cernes n'est pas homogène, c'est-à-dire qu'il y a des zones où la segmentation semble très bonne et d'autres zones où la segmentation est soit mauvaise,

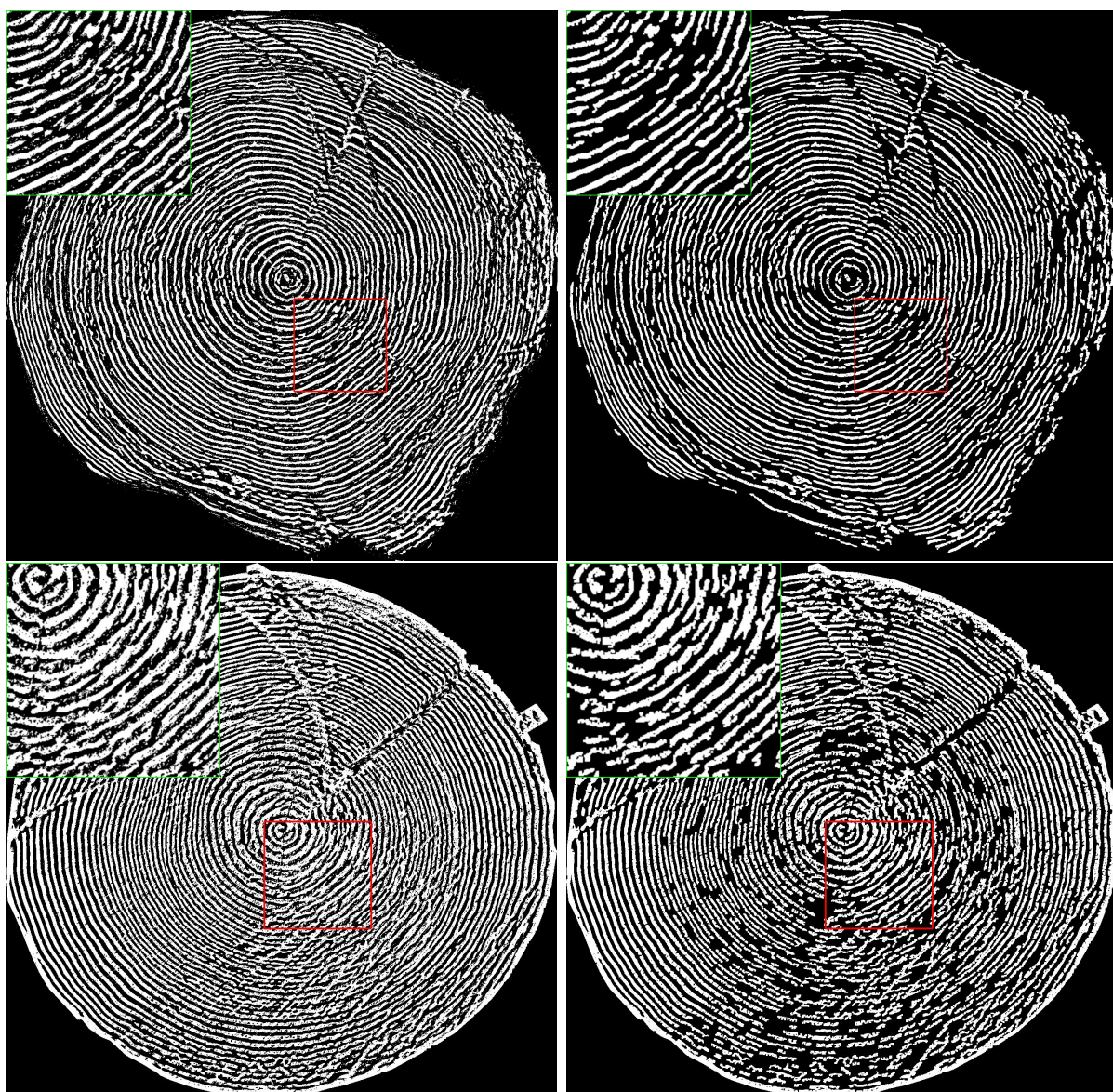


FIGURE 3.25 – Deux exemples du seuillage. À gauche, le seuillage adaptatif local avec la moyenne. À droite, le seuillage avec la suppression des petites composantes connexes. En haut à gauche de chaque image, un zoom du rectangle rouge.

soit le nombre de limites est très faible.

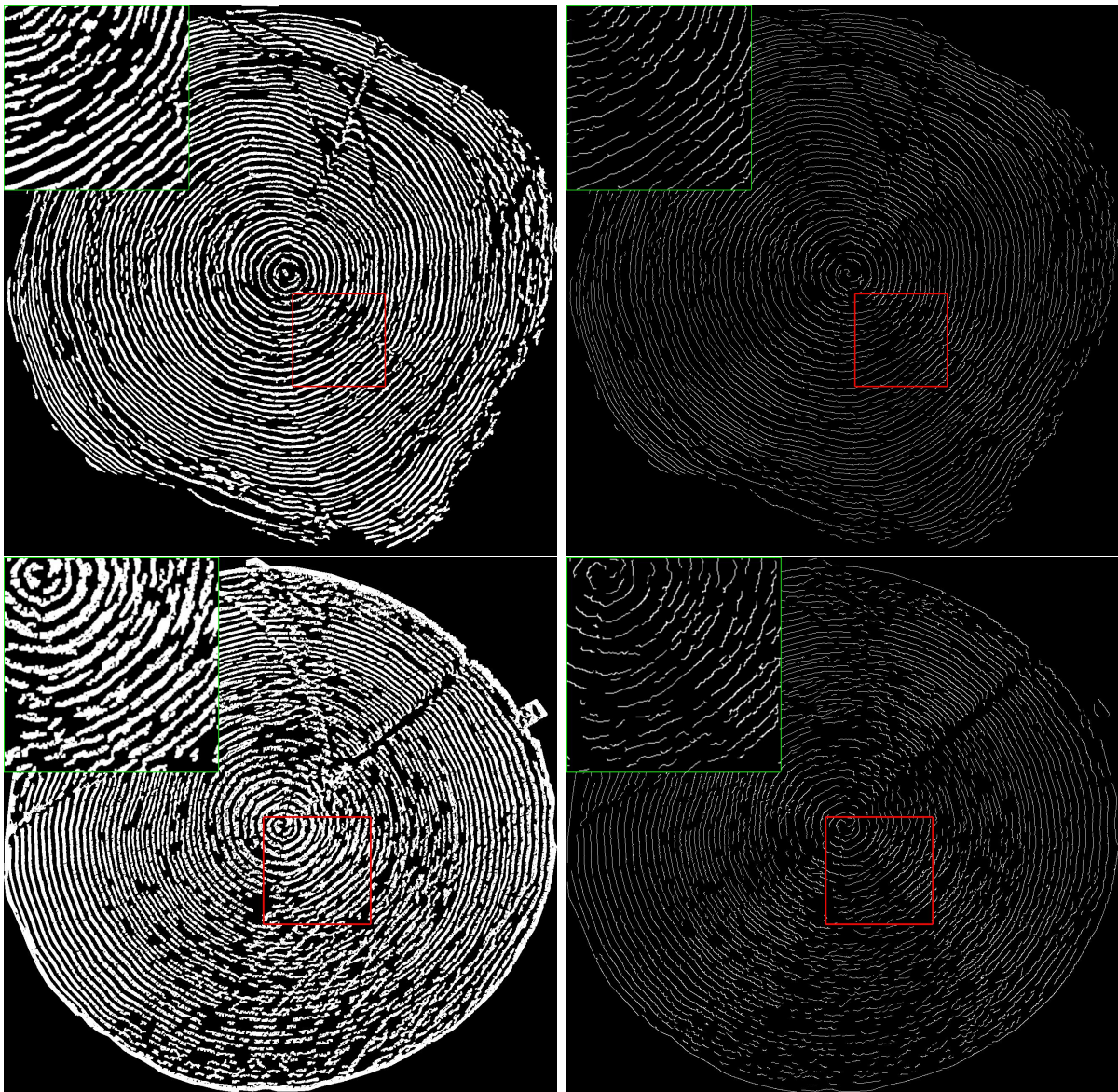


FIGURE 3.26 – Deux exemples de l’affinement du seuillage. À gauche, l’image seuillée. À droite, le seuillage affinée avec uniquement les transition noir/blanc le long d’un rayon partant de la moelle vers l’écorce. En haut à gauche de chaque image, un zoom du rectangle rouge.



FIGURE 3.27 – Résultat final de la segmentation des limites de cernes. En haut à gauche de chaque image, un zoom du rectangle rouge.

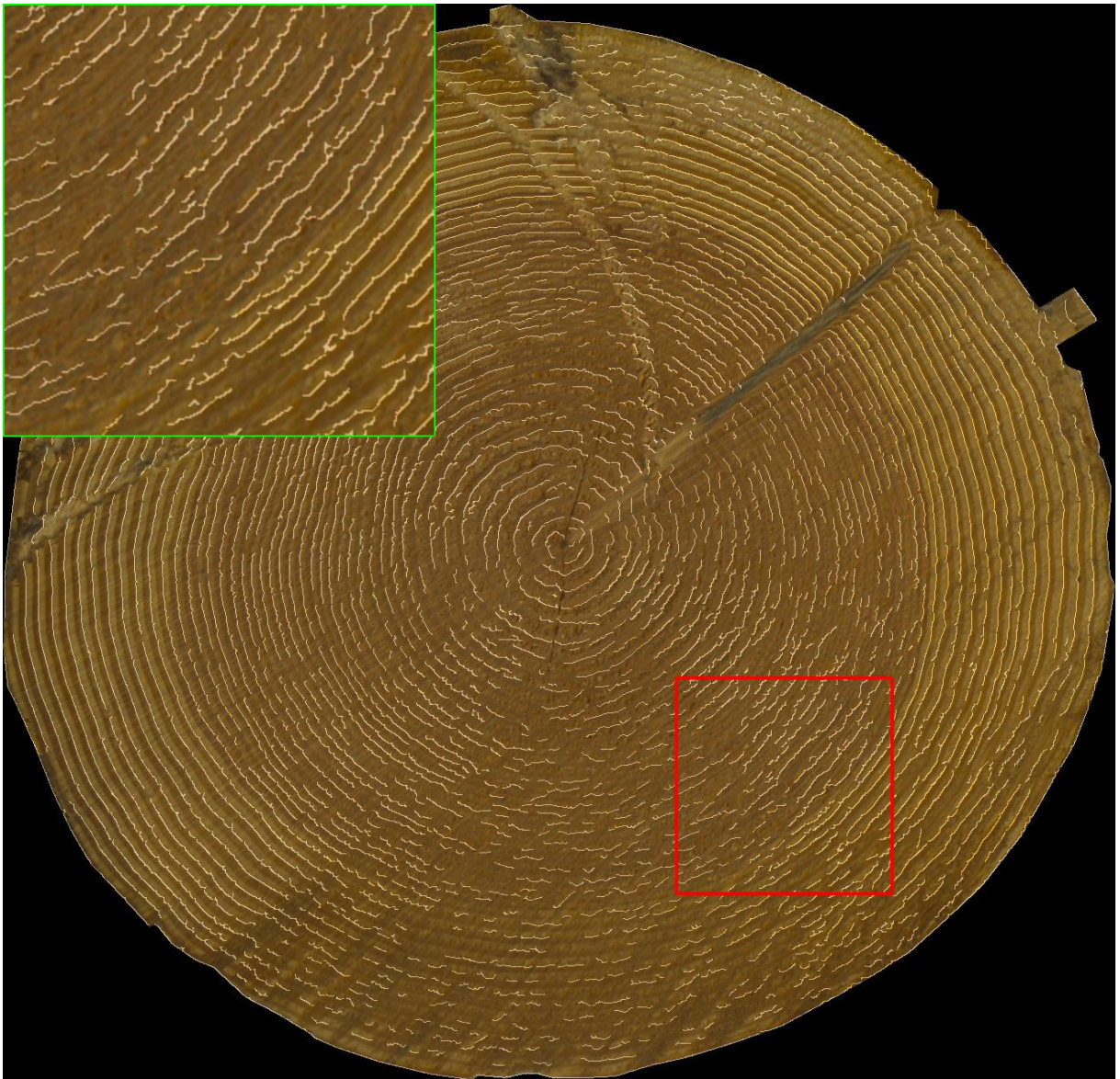


FIGURE 3.28 – Résultat final de la segmentation des limites de cernes. En haut à gauche de chaque image, un zoom du rectangle rouge.

### 3.5.5 Calcul de la largeur moyenne

Afin d'éviter de prendre en compte, dans le calcul de la largeur moyenne, les zones où la segmentation n'est pas bonne, nous proposons une méthode basée sur la décomposition de l'image combinée à un calcul de qualité. La décomposition va définir des zones où la qualité sera estimée. Cette qualité doit répondre à la question : pouvons-nous nous fier à la largeur calculée dans cette zone ? La qualité est évaluée selon un calcul d'orientation avec un filtre directionnel discret, basé sur des outils de géométries discrètes (voir le [chapitre 5](#)). En effet, si la segmentation est bonne, alors l'orientation dans cette zone doit être la même en tout point.

#### 3.5.5.1 Décomposition en quadtree

L'idée est de décomposer l'image en sous-images. Chaque sous-image doit être suffisamment petite de façon à ce que le calcul de qualité soit précis, mais pas trop petite pour éviter trop de calcul. Nous avons choisi de faire une décomposition en quadtree [[Samet, 1984](#)].

Le découpage d'un nœud se fait de la manière suivante. Un nœud est représenté par un rectangle dans l'image. Tant que nous n'avons pas atteint une profondeur minimum  $d_{min}$  dans l'arbre, nous découpons le nœud de l'arbre en 4. Si nous avons dépassé une profondeur maximale  $d_{max}$ , nous arrêtons de découper le nœud. Dans le cas où la profondeur est entre  $d_{min}$  et  $d_{max}$ , nous appliquons le découpage suivant :

1. Si le nombre de points (pixels blancs) dans le nœud est inférieur à un certain nombre (25% de la taille), alors nous ne découpons pas davantage le nœud. En effet, cela indique qu'il y a trop peu de cerne bien segmentés, il n'y a donc pas raison de se fier au calcul de la largeur dans ce nœud ;
2. Dans l'autre cas, le nœud est suffisamment petit pour faire un calcul de qualité. Si la qualité est inférieure à  $\epsilon$ , alors le nœud n'est pas de bonne qualité, nous le découpons à nouveau (en 4). Sinon nous arrêtons la découpe du nœud.

Chaque nouveau nœud est évalué et découpé de manière similaire. L'algorithme s'arrête quand plus aucun nœud n'a été découpé. Tout est résumé dans l'[algorithme 2](#).

#### 3.5.5.2 Calcul de la qualité

Le calcul de la qualité d'un nœud est fait à partir de l'orientation en utilisant un filtre directionnel discret (voir le [chapitre 5](#)). En effet, si les limites des cerne ont bien été segmentées, alors l'orientation dans une petite fenêtre doit être la même en tout point. Cependant, la méthode présentée au [chapitre 5](#) est très coûteuse en temps de calcul. C'est pourquoi nous faisons le choix d'appliquer la méthode sur un nombre  $m$  restreint de point. Les  $m$  points sont choisis de manière aléatoire. Pour chaque point, un ensemble de segments de droites discrètes est calculé. Au final, nous avons, pour un nœud, un ensemble de segments de droites discrètes d'angle  $\theta_i$ . Nous calculons l'orientation moyenne ainsi que

**Algorithme 2** : Pseudo-code de la création du quadtree

---

**Entrées** : Une image  $I$   
 $d_{min}$  la profondeur minimale de découpe  
 $d_{max}$  la profondeur maximale avant l'arrêt du découpage  
 $\epsilon$  critère de qualité  
 $F$  une file

**Sorties** : Un quadtree  $T$

- 1  $T \leftarrow$  ajouter l'image en tant que racine
- 2  $F \leftarrow$  filevide
- 3  $F \leftarrow$  ajouter la racine de  $T$
- 4 **tant que**  $F \neq \emptyset$  **faire**
- 5      $n \leftarrow$  retirer  $F$
- 6     Décomposer le noeud en quatre rectangles
- 7     **si**  $profondeur(n) < d_{min}$  **alors**
- 8          $F \leftarrow$  ajouter les quatre rectangles de  $n$
- 9          $T \leftarrow$  ajouter les quatre rectangles de  $n$  en tant que fils du noeud  $n$
- 10    **sinon si**  $d_{min} \leq profondeur(n) \leq d_{max}$  **alors**
- 11         **si** Nombre de pixels blancs dans  $n > 0,25 * taille(n)$  **alors**
- 12             **si**  $qualite(n) < \epsilon$  **alors**
- 13                  $F \leftarrow$  ajouter les quatre rectangles de  $n$
- 14                  $T \leftarrow$  ajouter les quatre rectangles de  $n$  en tant que fils du noeud  $n$
- 15         **sinon**
- 16             **si** Nombre de pixels blancs dans  $n > 0,25 * taille(n)$  **alors**
- 17                 Calculer la qualité du noeud  $n$
- 18 Retourner  $T$

---

la variance [Jammalamadaka and Sengupta, 2001]. Posons :

$$\bar{S} = \sum_i \sin(2\theta_i) \quad (3.16)$$

$$\bar{C} = \sum_i \cos(2\theta_i) \quad (3.17)$$

alors l'orientation moyenne est donnée par :

$$\bar{\theta} = \pi + \frac{1}{2} \text{atan2}\left(\frac{\bar{S}}{\bar{C}}\right) \quad (3.18)$$

et la variance par :

$$R = \sqrt{\bar{S}^2 + \bar{C}^2} \quad (3.19)$$

Ainsi, si un noeud a une variance  $R$  inférieure à  $\epsilon$ , ce noeud ne sera pas pris en compte pour le calcul de la largeur moyenne des cernes.



Les [figure 3.29](#) et [figure 3.30](#) montrent la décomposition en quadtree des deux images résultantes du seuillage non affiné (voir [figure 3.25](#)). Le choix de cette image est de pouvoir appliquer l'outil de géométrie discrète développé (voir le [chapitre 5](#)). Si possible, l'orientation moyenne et la qualité du nœud sont affichées. Plus un nœud est vert, plus sa qualité est proche de 1. Plus un nœud est rouge, plus sa qualité est proche de 0. Les nœuds en bleus sont les nœuds pour lesquels le calcul de la qualité n'a pu être faite, soit parce qu'il n'y pas assez de pixels blancs, soit parce que le nœud aurait dû être découpé mais la profondeur maximale  $d_{max}$  a été atteinte.

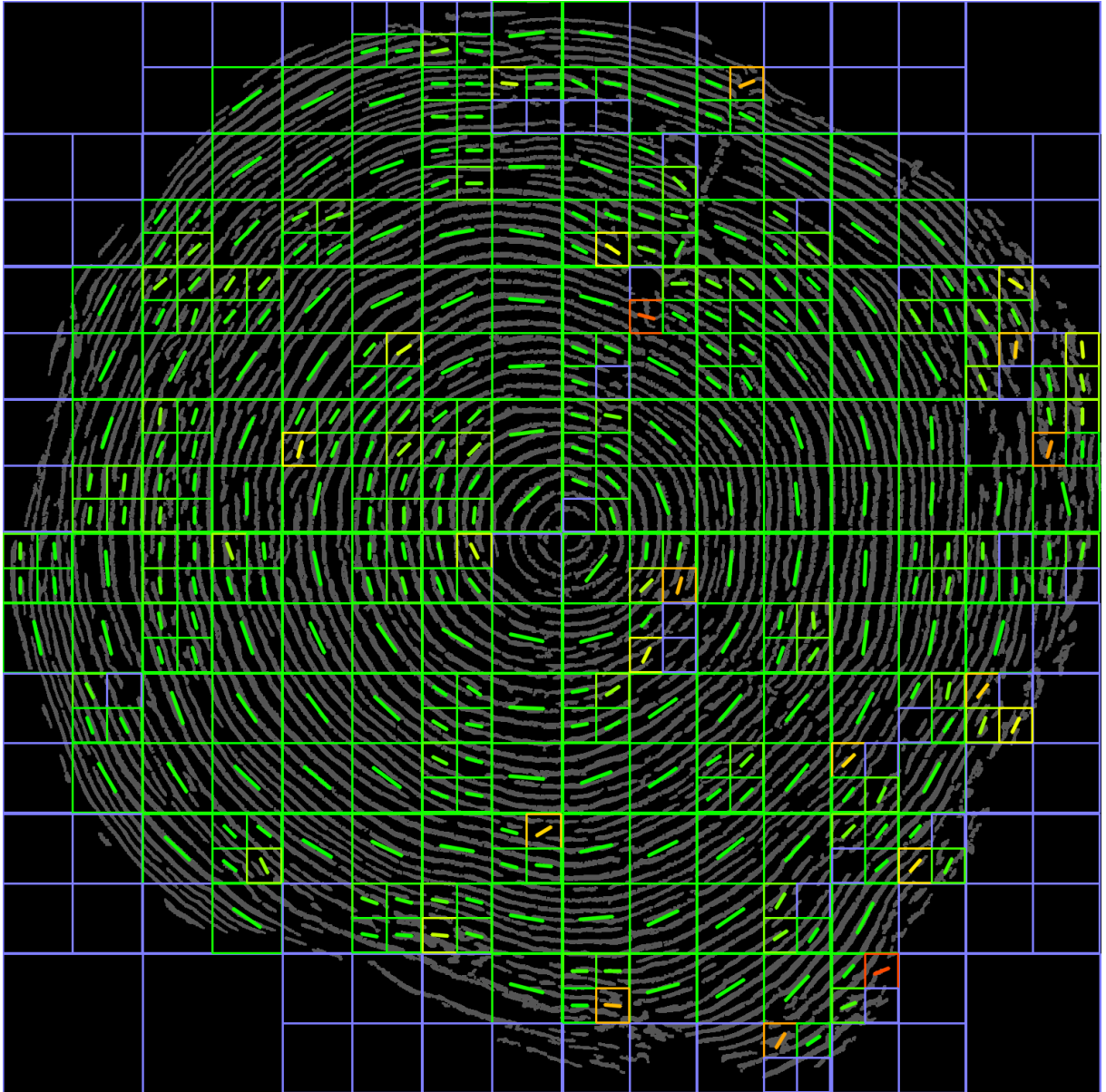


FIGURE 3.29 – Décomposition en quadtree avec  $\epsilon = 0,99$ ,  $d_{min} = 4, d_{max} = 5$ . La couleur du nœud indique la qualité du nœud. L'orientation moyenne est représentée par le trait au centre d'un nœud. Les nœuds bleus sont ceux pour lesquels il n'y a pas eu de qualité estimée.

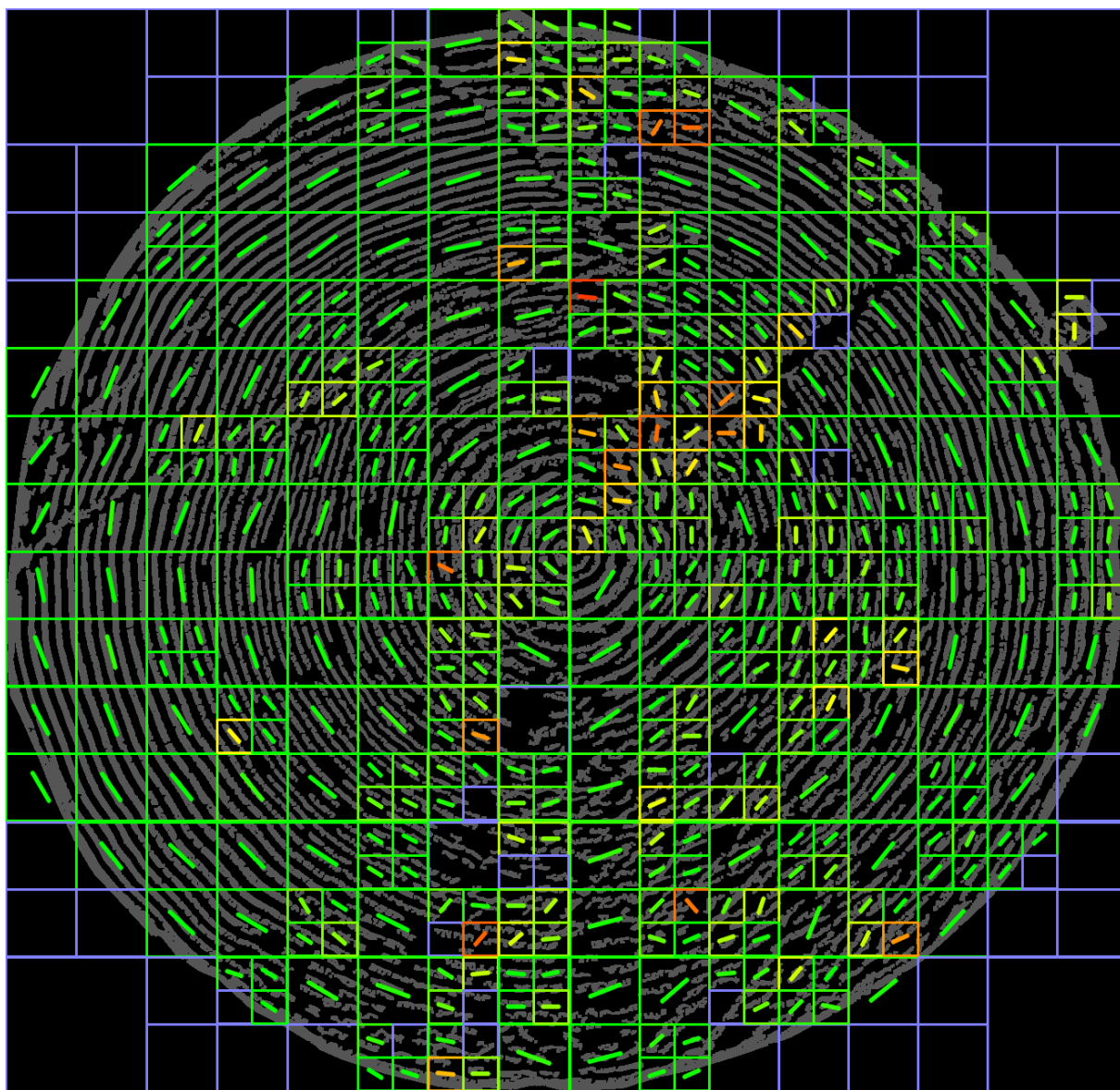


FIGURE 3.30 – Décomposition en quadtree avec  $\epsilon = 0,99$ ,  $d_{min} = 4, d_{max} = 5$ . La couleur du nœud indique la qualité du nœud. L'orientation moyenne est représentée par le trait au centre d'un nœud. Les nœuds bleus sont ceux pour lesquels il n'y a pas eu de qualité estimée.

### 3.5.5.3 Largeur moyenne des cernes

Une fois la décomposition quadtree de l'image, nous avons pu déterminer les zones de qualité, où nous pouvons nous fier à la segmentation et donc au calcul de la largeur.

Le calcul de la largeur se fait en traçant des rayons partant de la moelle vers le bord de l'image (comme pour l'affinement du seuillage). Nous calculons la distance moyenne entre deux pixels blancs le long de ce rayon. Si le rayon passe par un nœud de faible qualité (i.e. inférieure à  $\epsilon$ ), alors nous passons au rayon suivant.

### 3.5.6 Résultats

Malheureusement, nous n'avons pas de vérités terrain des cernes segmentés. Cela implique qu'il n'est pas possible, pour le moment, de vérifier la qualité des limites de cernes détectées de manière similaire aux autres segmentations (grume et duramen). Nous devons passer par un calcul intermédiaire pour vérifier notre approche.

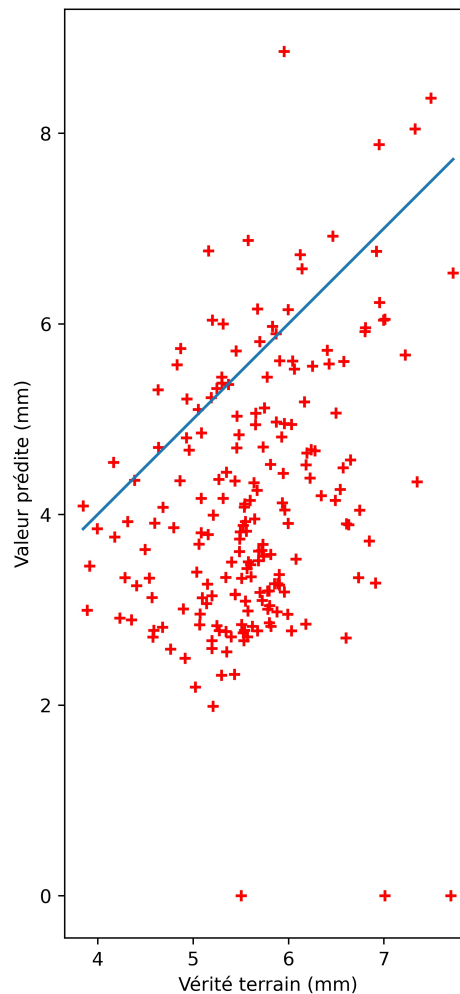


FIGURE 3.31 – Largeur moyenne estimée en fonction de la largeur moyenne de la vérité terrain. La droite représente  $y = x$ .

Les informations dont nous disposons sont : le nombre de cernes et la largeur de cerne pour 4 directions orthogonales. Il s'agit de la seule façon pour nous d'estimer la qualité de notre approche.

En appliquant la méthode proposée dans les paragraphes précédents, nous obtenons, pour le jeu *Sawmill*, les résultats présentés dans la [figure 3.31](#). Les croix ayant une estimation à 0 mm sont les images pour lesquels aucun rayon traversant uniquement des nœuds de qualité n'a pu être trouvé. Pour ces images, les 4 nœuds proches de la moelle ont une faible qualité (i.e. inférieure à  $\epsilon$ ), ce qui implique que l'ensemble des rayons ne sont pas pris en compte, puisqu'ils passent forcément par l'un de ces nœuds. Il n'y a pas de largeur moyenne calculée pour ces images.

Le biais est de  $-1,49$  mm, le RSME obtenu est de  $2,01$  mm et la valeur moyenne (des largeurs de cernes) des vérités terrains est de  $5,60$  mm. Nous en déduisons que la variance (RMSE) de nos prédictions correspond à environ 36% de la moyenne des vérités terrains. Cet écart est à mettre en perspective. D'une part, la vérité terrain a été obtenue à partir de 4 directions orthogonales, alors que dans notre méthode nous prenons potentiellement toutes les directions possibles. D'autre part, la résolution des images traitées n'est pas la même que celle utilisée pour les vérités terrain. Puisque nous ne pouvons être plus précis qu'une certaine distance, à savoir  $0,21$  mm, nous ne pouvons être plus précis que 3,75%. Cela étant, il reste à mieux paramétrer et ajuster notre modèle pour descendre jusqu'à ce seuil.

D'après les résultats, vu le biais négatif, nous en déduisons que nous sous-estimons la largeur moyenne des cernes. Cette sous-estimation peut s'expliquer par rapport à la détection des limites de cernes. En effet, dans les [figure 3.27](#) et [figure 3.28](#) nous pouvons remarquer que certains cernes sont parfois dédoublés, c'est-à-dire que nous avons pris le milieu d'un cerne pour une limite de cerne. Ce phénomène fait que la largeur moyenne est réduite.

## 3.6 Synthèse

Dans ce chapitre, nous avons présenté les images qui ont été acquises durant le projet. Différentes méthodes de segmentation d'images brutes de coupes transversales de grumes ont été présentées.

Ces méthodes ont permis de segmenter : la grume afin de faciliter l'estimation d'autres indicateurs de qualité du bois, le duramen et les cernes de croissances.

Pour la segmentation de la grume, deux méthodes n'impliquant pas de réseaux de neurones convolutifs ont été testées et quatre réseaux, fréquemment utilisés, ont été entraînés sur plusieurs jeux d'images acquis (6 jeux). Les deux méthodes sans apprentissage ont montré de bons résultats avec des scores de MCC de l'ordre de 0,7 pour les deux méthodes (K-Means et contours actifs). Mais, elles sont très limitées pour une généralisation. En effet, beaucoup d'hypothèses ont été faites et peuvent ne pas convenir à toutes les situations. Notamment, les images sur les parcs à grumes sont différentes des images acquises dans la scierie. Cependant, K-Means est la plus rapide et les contours actifs la plus lente. Les réseaux de neurones convolutifs ont montré de meilleurs résultats pour cette tâche. Notamment, RefineNet a des scores de MCC variant de 0,773 à 0,968, alors

que la méthode K-Mean varie entre 0,352 et 0,902 et le contour actif varie entre 0,246 et 0,826. Les autres réseaux offrent des scores légèrement inférieurs à 0,9.

Pour la segmentation du duramen, nous avons proposé une nouvelle couche d'attention utilisant de la morphologie mathématique. Cela a permis de mettre en place une architecture de réseau, basée sur U-Net, plus petite en nombre de paramètres et donc plus adaptée à nos jeux d'images réduits. Cette architecture légère (LU-Net) combinée à notre couche d'attention offre des scores de MCC de l'ordre de 0,93 avec peu de variabilité (un écart-type de 0,009). De plus, l'utilisation de cette couche n'augmente pas significativement le temps de calcul. Ce dernier est du même ordre de grandeur que l'architecture classique de U-Net et à peine supérieur de 20 ms à LU-Net sans couche d'attention.

Enfin, nous avons mis en place une méthode de segmentation des cernes qui repose sur un filtrage bilatéral directionnel. Ce filtrage permet de lisser l'image dans la direction des tangentes aux cernes, réduisant l'intensité des pixels entre les cernes tout en conservant au mieux les contours (et donc les limites des cernes). En effet, à cause du sciage de la grume, les cernes présentent un bruit (une texture) qui rend plus difficile leur analyse. Suite à ce filtrage, nous avons fait un seuillage adaptatif afin d'avoir une première segmentation. Celle-ci est affinée par un traitement afin de n'avoir que les limites des cernes. Nous avons par la suite proposé de décomposer l'image en bloc, en utilisant la décomposition en quadtree, afin d'identifier les zones dans lesquels nous pouvons effectuer le calcul de largeur des cernes. Ces zones sont identifiées selon un calcul de la variance de l'orientation, puisqu'à une échelle locale, les cernes ont la même direction. Cette décomposition nous permet d'omettre certaines largeurs moyennes qui sont calculées à l'aide de rayons partant de la moelle vers l'écorce. Une première estimation de la largeur moyenne des cernes montre que notre méthode sous-estime la largeur moyenne et montre que nous avons une variance par rapport à la moyenne d'environ 36%. Ces résultats peuvent faire l'objet d'amélioration, aussi bien dans la détection des limites de cernes que dans la décomposition et l'estimation de la qualité.



# Détection de la moelle dans l'image

## Sommaire

---

<b>4.1</b>	<b>État de l'art</b>	<b>132</b>
4.1.1	Norell et Borgefors (2008) [Norell and Borgefors, 2008]	132
4.1.2	Schraml et Uhl (2013) [Schraml and Uhl, 2013]	136
4.1.3	Kurdthongmee et al. (2018) [Kurdthongmee et al., 2018]	138
4.1.4	Kurdthongmee (2020) [Kurdthongmee, 2020]	141
<b>4.2</b>	<b>Approche proposée</b>	<b>142</b>
4.2.1	Pré-traitement	144
4.2.2	Application à la détection de la moelle	147
4.2.3	Dépôt des phéromones	149
4.2.4	Extraction de la position de la moelle	150
<b>4.3</b>	<b>Résultats</b>	<b>151</b>
<b>4.4</b>	<b>Analyse de la méthode proposée</b>	<b>153</b>
4.4.1	Analyse de la complexité	153
4.4.2	Effet des paramètres sur la convergence de l'algorithme	155
<b>4.5</b>	<b>Synthèse</b>	<b>158</b>

---



Nous venons de voir dans les chapitres précédents des méthodes de segmentation, puis leur application au bois à travers la segmentation de la grume, du duramen et des cernes de croissance. Nous avons également présenté de nouvelles méthodes pour ces différentes tâches. Au cours de ces chapitres, nous avons évoqué régulièrement la position de la moelle, aussi bien en tant qu'indicateur de qualité du bois, qu'en tant que point de l'image permettant d'effectuer des traitements plus adaptés. Pourtant, nous n'avons jamais abordé la question de sa localisation. Ce chapitre s'intéresse à cette question.

Nous allons aborder, dans la [section 4.1](#), les études qui se sont penchées sur la détection de la moelle sur des images brutes de coupes transversales de grumes de bois, c'est-à-dire une image d'une coupe fraîchement sciée sans traitement additionnel (tel que le ponçage ou polissage). Ces études sont présentées ici car elles n'utilisent pas de techniques de segmentation contrairement aux études présentées précédemment. Puis, nous proposerons une nouvelle approche, basée sur l'algorithme des colonies de fourmis, pour détecter la position de la moelle. Cette nouvelle approche est décrite dans la [section 4.2](#). Les résultats de la méthode sont présentés dans la [section 4.3](#). Puis, nous analyserons plus en détail le temps de calcul et la convergence de notre méthode dans la [section 4.4](#). Enfin, nous concluons le chapitre dans la [section 4.5](#).

## 4.1 État de l'art

Une hypothèse régulièrement faite, dans le traitement d'image de coupe transversale de grume de bois, est de considérer que proche de la moelle les cernes décrivent des cercles concentriques. La position de la moelle est le centre de ces cercles. Cette hypothèse a été faite régulièrement dans l'état de l'art traitant le même type d'images que celles présentées, mais également lorsqu'il s'agit d'images tomodensitométriques. Les méthodes pour la détection de la moelle peuvent être décrites de la manière suivante :

- calculer de l'orientation des cernes (c'est-à-dire l'orientation des tangentes aux limites de cernes) ;
- accumuler les normales aux cernes dans un espace d'accumulation similaire à la transformée de Hough [[Hough, 1962](#)] (voir [figure 4.1](#)) ;
- récupérer la position de la moelle selon l'espace d'accumulation.

La différence majeure entre les études est dans la manière de calculer l'orientation. Détaillons maintenant les approches qui ont été proposées.

### 4.1.1 Norell et Borgfors (2008) [[Norell and Borgfors, 2008](#)]

Dans cette étude deux méthodes pour calculer l'orientation sont présentées. Une fois l'orientation calculée, l'accumulation des normales des tangentes aux cernes est fait de manière différente selon la méthode. L'extraction de la position se fait à partir de l'espace d'accumulation. Ce processus est itéré une ou plusieurs fois sur des fenêtres centré sur l'estimation courante.

Les images utilisées dans cette étude ont été prises par une caméra industrielle. Elles ont une taille de  $2046 \times 1534$  pixels. Plusieurs jeux de données sont faits : un jeu exclusivement d'épicéa commun (*Picea abies*), un jeu exclusivement de pin sylvestre (*Pinus*

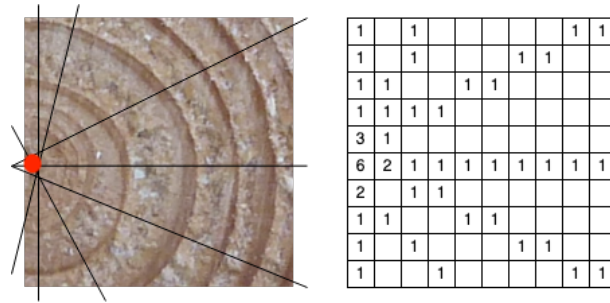


FIGURE 4.1 – Création d'un espace d'accumulation à l'aide des normales aux tangentes des cerne.

*sylvestris*) et le dernier jeu mélangeant les deux essences. Au total, 53 images sont traitées. Les vérités terrains ont été faites manuellement. L'une des images a dû être retirée, car la moelle n'était pas visible. Deux exemples sont montrés dans la [figure 4.2](#).

La méthode proposée est la suivante [[Norell and Borgefors, 2008](#)]. Les images sont converties en niveaux de gris avec le même poids pour les 3 composantes. Puis, l'orientation est calculée de deux manières différentes. Cela amène les autrices à développer de deux manières différentes l'estimation de la localisation de la moelle.

Le résumé de ces deux méthodes est illustré dans la [figure 4.3](#).

#### 4.1.1.1 Filtres en quadrature

##### Calcul de l'orientation

La première méthode pour calculer l'orientation est faite avec les filtres en quadrature [[Granlund and Knutsson, 1994](#)]. Le calcul est effectué dans le domaine de Fourier car ces filtres sont séparables en deux, une composante radiale et une composante angulaire :

$$F(\mathbf{u}) = R(\rho)D(\hat{\mathbf{u}}) \quad (4.1)$$

où  $\mathbf{u}$  est la coordonnée fréquentielle,  $\rho = \|\mathbf{u}\|$  et  $\hat{\mathbf{u}}$  le vecteur  $\mathbf{u}$  normalisé. La composante radiale est calculée par la formule :

$$R(\rho) = \exp\left(-\frac{4}{B \ln(2)} \ln^2\left(\frac{\rho}{\rho_i}\right)\right) \quad (4.2)$$

où  $\rho_i$  est la fréquence centrale et  $B$  la bande-passante relative (i.e. la plage des fréquences à détecter). La composante angulaire est calculée par la formule :

$$D(\hat{\mathbf{u}}) = \begin{cases} (\hat{\mathbf{u}}^T \hat{\mathbf{n}})^2 & \text{si } \hat{\mathbf{u}}^T \hat{\mathbf{n}} > 0 \\ 0 & \text{sinon} \end{cases} \quad (4.3)$$

où  $\hat{\mathbf{n}}$  est le vecteur unitaire de  $j$  directions. Pour une image 2D,  $j$  vaut 3 [[Knutsson, 1982](#)]. Le résultat des filtres en quadrature permet de calculer un tenseur d'orientation de taille  $2 \times 2$ . Ce tenseur possède deux valeurs propres  $\lambda_1 \geq \lambda_2$  associés à deux vecteurs propres

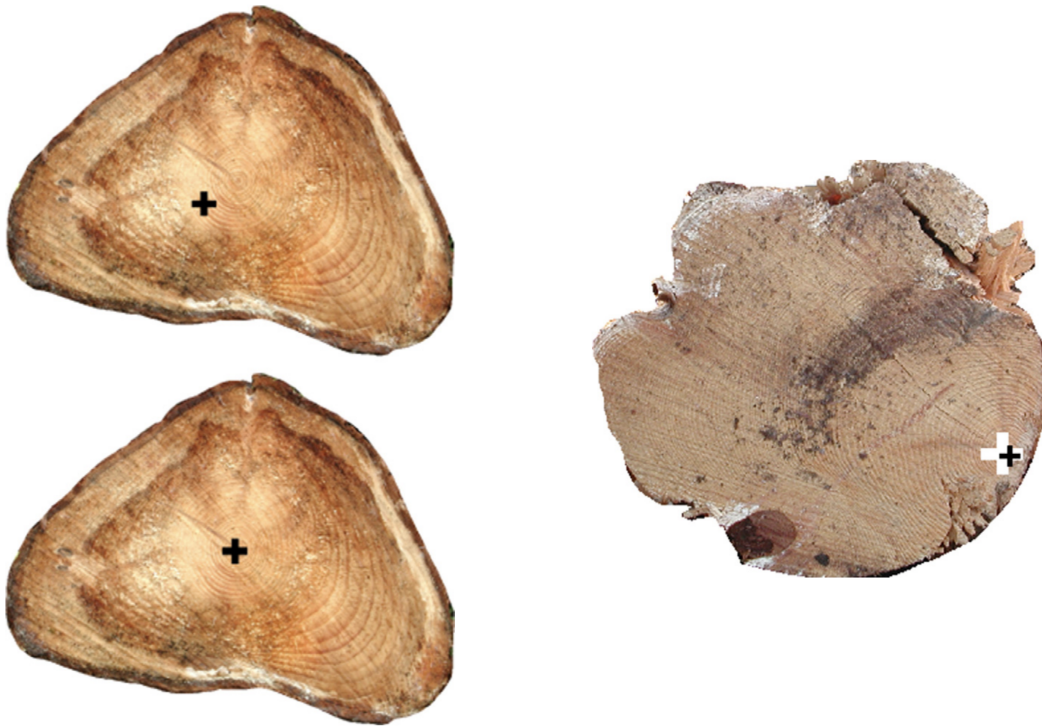


FIGURE 4.2 – Images utilisées dans [Norell and Borgfors, 2008]. La première montre l'estimation de la première étape puis le résultat de la seconde étape. Dans les deux, il s'agit de la méthode utilisant la pyramide Laplacienne.

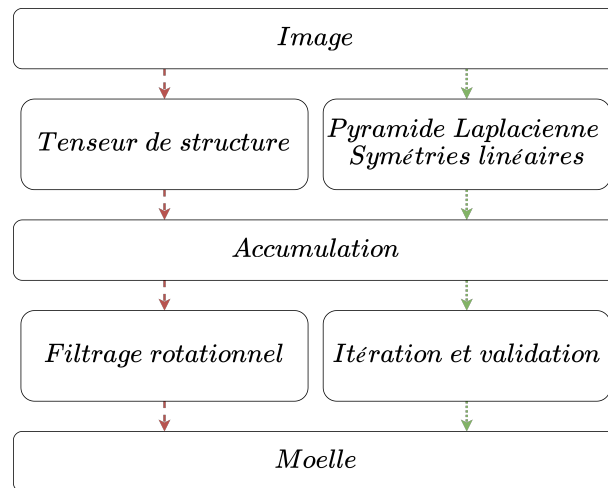


FIGURE 4.3 – Résumé des deux méthodes proposées par [Norell, 2011]

$\mathbf{e}_1$  et  $\mathbf{e}_2$ . Le premier vecteur propre représente la direction dominante dans le voisinage du pixel. Il est possible d'attribuer à ce pixel une valeur de confiance :

$$c = \frac{\lambda_1 - \lambda_2}{\lambda_1}$$

qui varie entre 0 et 1.

## Création de l'espace d'accumulation

Pour le calcul de l'espace d'accumulation, afin de ne pas accumuler l'intégralité des orientations possibles qui serait très onéreux en temps de calcul, les autrices proposent d'effectuer le calcul sur des blocs carrés d'une certaine taille (paramètre de leur méthode).

Dans cette première méthode, les autrices utilisent une valeur de *confiance*  $c$  permettant de savoir si l'orientation calculée est fiable. Ainsi, si la confiance en l'orientation est trop faible pour un bloc, alors aucune droite n'est accumulée (pour ce bloc-ci).

## Estimation de la position de la moelle

La première estimation de la position de la moelle est faite en calculant le barycentre des points ayant une valeur d'accumulation supérieure à  $T \times M$ , où  $T$  est un seuil et  $M$  le nombre maximum d'accumulations.

Pour cette première méthode (celle utilisant les filtres en quadrature), lors de la deuxième estimation, les autrices corrigent le calcul de l'orientation. Pour cela un produit scalaire est calculé entre l'orientation et un champ de vecteurs de taille  $2m + 1$  centrée sur le pixel  $(i, j)$ . Le résultat de ce filtrage est mis à la puissance 20, puis filtré par une gaussienne d'écart-type 0,67. La position de moelle est la moyenne des pixels ayant une valeur supérieure à 0,99 fois la valeur maximale.

Si la distance entre la première estimation et la deuxième est supérieure à un seuil, alors une nouvelle estimation est faite (à partir de la deuxième estimation avec la seconde étape). Ce processus est itéré un certain nombre de fois (10 au maximum).

### 4.1.1.2 Pyramide Laplacienne

Les autrices proposent une seconde méthode pour estimer la position de la moelle.

## Calcul de l'orientation

La deuxième méthode employée utilise une pyramide Laplacienne, c'est-à-dire le calcul de l'orientation est fait sur plusieurs échelles de l'image et la symétrie linéaire. Cette pyramide se construit en calculant la différence entre deux images filtrées par une gaussienne avec différents écart-types  $\sigma$  (fixé à 2, 5). L'orientation est calculée de la manière suivante, posons :

$$z = \frac{(\nabla f)^2 * m}{|\nabla f|^2 * m} \quad (4.4)$$

où  $\nabla f$  est la dérivée de l'image,  $m$  un filtre et  $*$  le produit de convolution.  $z$  est interprété comme la confiance dans cette deuxième méthode. L'orientation est égale à l'argument du nombre complexe  $z$ . La dérivée est calculée par convolution avec le filtre Gaussien suivant :

$$D = (x + iy)e^{-\beta(x^2+y^2)}$$

où le calcul est effectué dans le plan complexe ( $i$  désigne l'unité imaginaire ici).

## Création de l'espace d'accumulation

Pour la pyramide Laplacienne, l'écart-type de l'espace d'accumulation est calculé afin de savoir s'il l'on procède à l'accumulation dans le niveau suivant dans la pyramide ou non. Le niveau ayant le petit écart-type est choisi comme première estimation.

## Estimation de la position de la moelle

La première estimation de la position de la moelle est faite en calculant le barycentre des points ayant une valeur d'accumulation supérieure à  $T \times M$ , où  $T$  est un seuil et  $M$  le nombre maximum d'accumulations.

Pour cette seconde méthode, une fois la première estimation calculée, une deuxième estimation est faite sur une fenêtre centrée sur la première estimation. La deuxième estimation est faite avec de nouvelles valeurs pour chaque paramètre.

### 4.1.1.3 Résultats

Leurs méthodes ont de bons résultats. Sur les jeux d'images, composés de 12 images d'épicéa commun, de 21 images de pin sylvestre et de 20 images des deux essences (ayant été laissées environ 5 semaines à l'air libre), la position de la moelle est inférieure à une distance de 10 mm par rapport à leurs vérités terrains. Leurs méthodes permettent de prendre en compte des grumes non-circulaires, mais aussi le cas d'une moelle très excentrée. La présence de bruits (nœuds, coups de scie) semble n'avoir que très peu d'effet sur les résultats, à l'exception des cas où ces bruits sont très présents au point qu'il est difficile à l'œil de savoir où se situe la moelle.

Les défauts de leurs méthodes reposent sur le temps de calcul, notamment pour les filtres en quadrature. En effet, la correction de l'orientation est très coûteuse en temps. La pyramide Laplacienne est beaucoup plus rapide d'après les autrices. Il serait de l'ordre de 2 à 3 secondes, mais aucun benchmark n'est fourni.

## 4.1.2 Schraml et Uhl (2013) [Schraml and Uhl, 2013]

La méthode proposée repose sur le même principe que celui de l'entête de la section. Les auteurs calculent l'orientation des cernes, puis les normales sont accumulées pour finalement estimer la position de la moelle à partir de cet espace d'accumulation. Les auteurs proposent d'analyser la transformée de Fourier pour déduire l'orientation locale des cernes. Plusieurs méthodes d'accumulations sont également proposées.

Les images de l'article de Schraml et Uhl [Schraml and Uhl, 2013] ont été présentées dans le [chapitre 2](#). Il s'agit d'images d'épicéa commun. La [figure 4.4](#) montrent deux images issues de leur jeu de données.

### 4.1.2.1 Calcul de l'orientation

Le calcul de l'orientation se base sur l'analyse de la transformée de Fourier dans une fenêtre de taille carré. À une échelle locale, les cernes peuvent être vus comme des droites parallèles. Ainsi, dans le domaine fréquentiel, une droite perpendiculaire aux cernes devrait



FIGURE 4.4 – Deux exemples du type d'images traitées par [Schraml and Uhl, 2013].

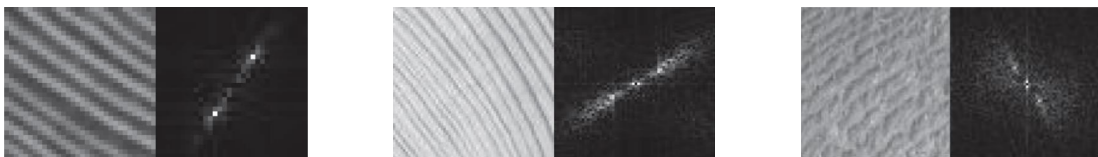


FIGURE 4.5 – Exemples de couples image d'origine/FFT pour des blocs montrant des cernes de croissances.

être visible. La [figure 4.5](#) montre quelques exemples de la transformée de Fourier appliquée sur les cernes. Après avoir effectué la transformée de Fourier, les auteurs appliquent un filtre passe-bande pour éliminer les fréquences ne contenant pas d'informations relatives à l'orientation des cernes. Puis, un seuillage est fait dépendant des maxima. À partir de cette transformée de Fourier seuillée, les auteurs proposent plusieurs façons d'effectuer une analyse de l'orientation :

- une analyse des pics ;
- une régression (méthode des moindres carrés) ;
- une régression pondérée (méthode des moindres carrés) ;
- une analyse par composantes principales.

#### 4.1.2.2 Création de l'espace d'accumulation

Pour le calcul de l'espace d'accumulation, l'analyse de la transformée de Fourier est faite sur des blocs. Les auteurs ne font l'accumulation que d'un certain nombre de blocs.

##### Par blocs

Ces blocs sont disposés autour du centre géométrique de l'image de façon soit rectangulaire soit circulaire. La superposition des blocs est aussi testée. Pour chaque bloc, l'orientation est estimée, puis une droite est tracée selon la direction calculée. L'espace d'accumulation est ainsi obtenu.

## Point Block Selection (PBS)

Une autre méthode est proposée [Österberg et al., 2004]. Il s'agit d'une approche similaire à la méthode RANSAC [Fischler and Bolles, 1981]. Plusieurs points de l'image sont choisis de manière aléatoires, sur un cercle de rayon prédéfini autour d'un point de référence (initialement le centre de l'image). Chaque point est le centre d'un ou plusieurs blocs disposés de manière rectangulaire. L'orientation est calculée pour chacun de ces blocs. Le barycentre des intersections des droites (issues de l'analyse de la transformée de Fourier) devient le nouveau point de référence. Les droites ne sont prises en compte que si l'orientation estimée est fiable. Il doit y avoir au minimum deux blocs valides afin d'avoir au moins une intersection. L'algorithme s'arrête lorsque le nouveau point de référence est proche de l'ancien ou au bout d'un certain nombre d'itérations.

### 4.1.2.3 Estimation de la position de la moelle

Dans le cas où l'espace d'accumulation serait obtenu par l'ensemble des blocs disposés dans l'image, les auteurs effectuent un filtrage de ce dernier. Le filtrage est une convolution par un filtre gaussien. La moelle est localisée au niveau du pixel ayant le plus d'accumulations.

Dans la méthode PBS, la nouvelle estimation est le barycentre des intersections.

### 4.1.2.4 Résultats

Quelle que soit la méthode pour estimer l'orientation, les auteurs obtiennent une distance moyenne à la moelle d'environ 10 mm. Dans ce cas, le temps de calcul est de l'ordre d'une seconde.

## Par blocs

Si les blocs sont autorisés à se superposer, alors la distance moyenne se voit réduite à environ 6,8 mm. Mais le temps de calcul est doublé.

## PBS

Dans le cas de la méthode PBS, les résultats sont plus disparates. En moyenne, la distance à la vérité terrain est de 11,9 mm. De plus, la variation autour de cette moyenne est plus élevée (13,6 mm d'écart-type). Autrement dit, la méthode, en plus d'être moins précise, est moins robuste. Cependant, cette approche permet d'effectuer l'estimation de la moelle en moins de 200 ms.

### 4.1.3 Kurdthongmee et al. (2018) [Kurdthongmee et al., 2018]

Cet article propose d'utiliser une autre approche pour calculer l'orientation des cernes. L'orientation est estimée à l'aide d'histogrammes des gradients orientés (HOGs).

L'essence de bois étudiée est de l'hévéa. La [figure 4.6](#) montrent deux images issues de leur jeu de données. Nous pouvons y voir des cernes très peu marqués par rapport aux bois européens des études précédentes.



FIGURE 4.6 – Deux images issues du jeu d'images utilisé par [Kurdthongmee et al., 2018] et [Kurdthongmee and Suwannarat, 2019].

#### 4.1.3.1 Calcul de l'orientation

Le calcul de l'orientation se fait à l'aide l'histogramme des gradients orientés [Dalal and Triggs, 2005] (HOGs). L'idée est de faire un histogramme des orientations, obtenu selon le gradient, pour un ensemble de blocs disposés dans l'image . Les blocs ne se superposent pas entre eux. La première étape consiste à calculer le gradient (masque de Sobel-Feldmann de taille  $3 \times 3$  par exemple).

La seconde étape est la création des HOGs. Ceci est fait dans des blocs carrés de petite taille (typiquement  $4 \times 4$  ou  $12 \times 12$  pixels). Chaque pixel du bloc *vote* pour une classe de l'histogramme, en fonction de l'orientation de son gradient. La valeur ajoutée dans l'histogramme est pondérée par l'intensité du gradient.

La troisième étape est la normalisation afin d'éviter les disparités dues aux variations. Cette étape introduit également de la redondance. Pour cela, les blocs sont agrégés entre eux (en général  $2 \times 2$  blocs ou plus si besoin). C'est sur cette agrégation de blocs que la normalisation est faite (norme-1, norme euclidienne, etc.).

Dalal et Triggs font remarquer qu'une quantification fine de l'histogramme est nécessaire, et ils obtiennent leurs meilleurs résultats avec un histogramme à 9 classes. Prendre en compte le signe du gradient n'améliore pas les performances pour la détection de personnes, mais peut être significatif pour d'autres types d'objets.

#### 4.1.3.2 Création de l'espace d'accumulation

Pour chaque bloc résultant du calcul des HOGs, une droite est tracée dans la direction perpendiculaire. Mais les directions sont sélectionnées selon la position du bloc dans l'image. En effet, les auteurs considèrent que certaines directions ne sont pas pertinentes à accumuler selon la position du bloc. La figure 4.7 montre ces possibilités selon la position



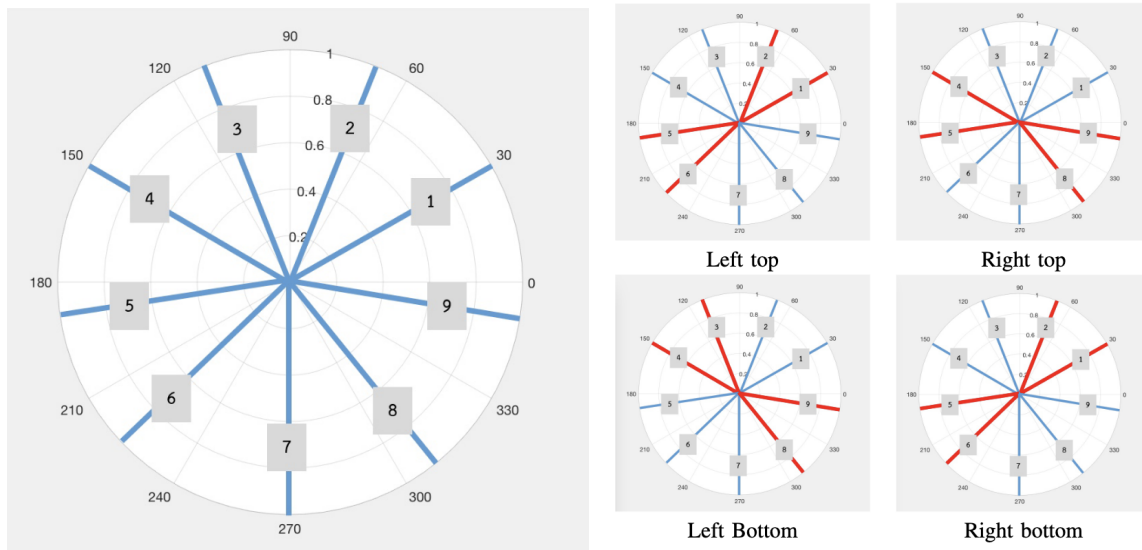


FIGURE 4.7 – Illustration des directions possibles en utilisant les HOGs avec un paramètre de 9 possibilités. À droite, en rouge les directions uniquement possibles selon la position dans l'image.

dans l'image. Cette restriction sous-entend que la moelle est proche du centre de l'image.

L'espace d'accumulation n'est pas de la taille de l'image. Les auteurs, pour réduire le temps de calcul, proposent d'avoir un espace d'accumulation plus petit. C'est-à-dire qu'un pixel de l'espace représente  $k \times k$  pixels de l'image.

#### 4.1.3.3 Estimation de la position de la moelle

Une fois l'espace d'accumulation créé, la position de la moelle est considérée comme étant la position du point ayant le plus d'accumulations.

Dans une version plus récente de leur travaux [Kurdthongmee and Suwannarat, 2019], les auteurs proposent une autre façon d'estimer la position de la moelle à partir de l'espace d'accumulation créé. En effet, les auteurs constatent qu'estimer la position de la moelle avec le point ayant le maximum d'accumulations donne des résultats médiocres dès que les cernes ne décrivent pas des cercles concentriques. La position de la moelle est le barycentre, pondéré par le nombre d'accumulations, des points ayant au moins un certain nombre d'accumulations. Ce seuil est fixé comme  $T$  fois la valeur maximale ( $T$  variant entre 0 et 1).

#### 4.1.3.4 Résultats

Les auteurs comparent leur méthode [Kurdthongmee and Suwannarat, 2019] à leur implémentation de la méthode proposée par Norell et Borgefors [Norell and Borgefors, 2008]. Néanmoins, rien n'est dit sur le choix de la méthode implémentée, à savoir les filtres en quadrature ou la pyramide Laplacienne.

La première chose à noter est le temps de calcul donné pour la méthode [Norell and Borgefors, 2008] qui dépasse les 5000 secondes. Cette valeur est considérablement élevée, alors que les autrices [Norell and Borgefors, 2008] évoquaient un temps de calcul de l'ordre de quelques secondes pour la pyramide Laplacienne. Le temps de calcul est d'autant plus surprenant, que la puissance des ordinateurs a considérablement augmenté entre 2008 et 2018.

Ceci étant, leur méthode est quant à elle beaucoup plus rapide et prend environ 32 secondes pour des HOGs sur de fenêtres de taille  $8 \times 8$  pixels. Ce temps reste important dans le cadre d'une application en temps réel. Le temps de calcul descend à 2 secondes pour des fenêtres de taille  $16 \times 16$  pixels et une seconde pour des fenêtres de taille  $32 \times 32$  pixels.

Cependant, le gain en temps de calcul a un effet sur la performance de leur méthode. La distance moyenne entre la vérité terrain et l'estimation est d'environ 7 mm lorsque les blocs sont de taille  $8 \times 8$  pixels (avec un temps de calcul 32 secondes). La distance moyenne passe à 10 mm pour des blocs de taille  $16 \times 16$  pixels. Enfin, elle passe à plus 17 mm pour des blocs de taille  $32 \times 32$  pixels.

L'avantage de leur méthode est de pouvoir estimer la moelle même si les cernes sont très peu présents. Néanmoins, leur méthode a le défaut de considérer que la moelle est forcément proche du centre de l'image ; ce qui n'est pas forcément le cas (voir la [figure 4.2](#)).

#### 4.1.4 Kurdthongmee (2020) [Kurdthongmee, 2020]

Kurdthongmee propose en 2020 [Kurdthongmee, 2020] d'utiliser des réseaux de neurones afin d'estimer la position de la moelle. Les images sont les mêmes que celles de l'article [Kurdthongmee and Suwannarat, 2019]. Les deux réseaux utilisés sont YOLO v3 [Redmon and Farhadi, 2018] et SSD MobileNet [Howard et al., 2017]. La sortie de ces deux réseaux est une boîte englobante. La moelle est le centre de cette boîte. Les auteurs ne proposent pas de modifications des réseaux.

Les auteurs utilisent une augmentation de données, ce qui amène leur jeu d'images à avoir 1380 images. L'augmentation est faite en effectuant des rotations, en changeant la saturation, la teinte et l'exposition.

La fonction de coût est l'erreur quadratique :

$$\mathcal{L} = \sum_i (y_i - \hat{y}_i)^2$$

Après l'apprentissage des deux réseaux, les auteurs obtiennent les résultats présentés dans la [tableau 4.1](#). Les auteurs calculent, dans cet article, la distance en pixels (en non en millimètres) entre la vérité terrain et l'estimation.

Les résultats présentés sont meilleurs que ceux de la méthode basée sur les HOGs. Nous pouvons remarquer que les réseaux ne garantissent pas une estimation précise pour chaque image. D'une part, l'écart-type est élevé. En effet, en considérant qu'un pixel vaut 0,3 mm (l'ordre de grandeur pour nos jeux d'images), alors un écart-type de 15,32 pixels donne un écart-type de 4,596 mm. D'autre part, des valeurs maximales de 96,71 (pour YOLO v3) et 193,19 (pour SSD MobileNet) illustre la variabilité des résultats. De tels écarts ne peuvent pas être envisagés, puisqu'il s'agit d'un élément important dans l'estimation de la

	Moyenne	Écart-type	Min	Max	Temps (en ms)
<b>YOLO v3</b>	7,61	13,8	0,05	96,71	1050
<b>SSD MobileNet</b>	7,04	15,32	0,11	193,19	780

TABLE 4.1 – Moyenne, écart-type, minimum et maximum entre les vérités de terrain et l'estimation de la position de la moelle en pixels d'après [Kurdthongmee, 2020].



FIGURE 4.8 – Craquelures présentes proches de la moelle. Ces craquelures apparaissent lors du séchage du bois.

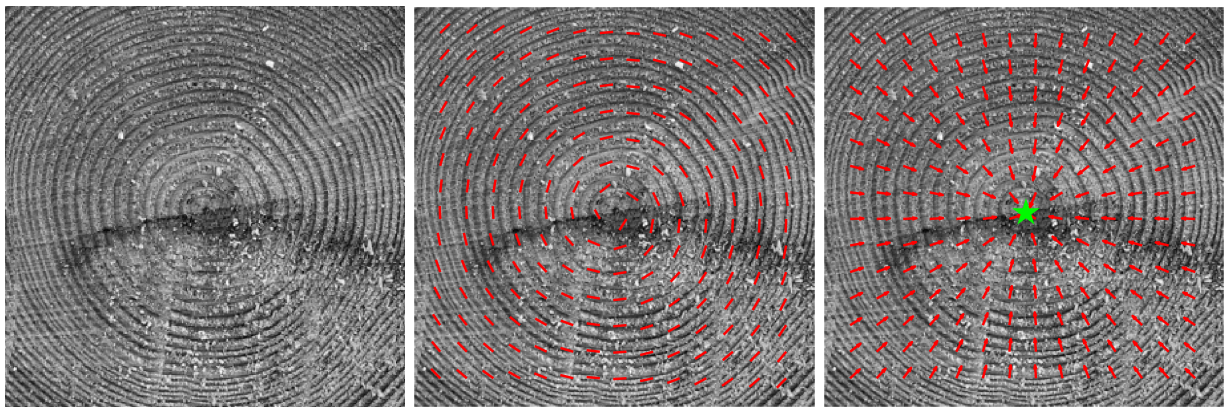
qualité du bois (nécessaire pour estimer l'écart avec le centroïde de la grume). La moelle est aussi importante dans la suite, puisque, comme nous l'avons vu dans le [chapitre 3](#), elle peut être utilisée pour faciliter l'estimation d'autres caractéristiques (largeur de cernes). Enfin, soulignons que la tache rouge au niveau de la moelle a potentiellement un effet sur l'apprentissage des réseaux. Ce biais potentiel n'est pas mentionné par les auteurs.

Malgré tout, les réseaux de neurones arrivent à estimer la position de la moelle sur des cas très difficiles, là où des méthodes se basant sur le calcul de l'orientation ne pourraient être performantes. Notamment, les auteurs soulignent que les réseaux peuvent apprendre à détecter des caractéristiques souvent omises, comme la présence de craquelures partant de la moelle (voir la [figure 4.8](#)).

## 4.2 Approche proposée

Contrairement aux méthodes de l'état de l'art qui se sont focalisées principalement sur le calcul de l'orientation, nous proposons de nous concentrer sur la manière de construire l'espace d'accumulations. En effet, les méthodes de l'état de l'art possèdent des temps d'inférence (hors réseaux de neurones) qui peuvent être un frein dans mise en application en temps-réel. Ce temps de calcul peut-être lié soit à la méthode d'estimation de l'orientation, soit à la manière d'effectuer l'accumulation. C'est sur ce dernier point que nous nous focalisons.

Si nous essayons de formaliser le problème de recherche de la moelle, nous pouvons définir l'orientation comme une fonction  $\theta$  qui en tout point de l'image associe l'angle des tangentes aux cernes (voir la [figure 4.9b](#)). Il est possible de définir un champ de vecteurs



(a) Zone proche de la moelle      (b) Champ d'orientation      (c) Champ de vecteurs

FIGURE 4.9 – Image illustrant la notion de champ d'orientation et de champs de vecteurs dans le cas de la détection de la moelle. L'étoile verte représente le point fixe recherché.

$\mathcal{V}$  qui en tout point de l'image associe un vecteur,  $\mathcal{V} : (x, y) \mapsto (f(x), g(x))$ . Ce champ de vecteurs est illustré dans la [figure 4.9c](#). On définit point fixe  $s^*$ , un point tel que  $\mathcal{V}(s^*) = 0$ . L'objectif est alors de trouver le point fixe correspondant à la moelle (voir l'étoile verte sur la [figure 4.9c](#)).

Pour trouver ce point fixe, nous avons choisi d'utiliser l'algorithme des colonies de fourmis [[Dorigo et al., 1996](#)] que nous avons adapté à la détection de la moelle. Ce choix d'algorithme tiens d'abord du fait que nous avons décidé d'améliorer de la méthode [[Österberg et al., 2004](#), [Schraml and Uhl, 2013](#)]. Lors de nos recherches, nous nous sommes rendus compte que notre proposition était très similaire à l'algorithme des colonies de fourmis. Nous avons formalisé notre approche en suivant cet algorithme d'optimisation et nous l'avons mis en œuvre.

Dans cette section, nous présenterons les principales étapes de notre méthode pour estimer la position de la moelle. La [figure 4.10](#) montre les trois étapes de notre méthode : un pré-traitement qui vise à la fois à supprimer les marques de sciage et à rehausser les cernes, le calcul de l'orientation des cernes et enfin notre implémentation, pour la détection de la moelle, de l'algorithme des colonies de fourmis. À noter que la segmentation de l'image n'est pas obligatoire. En effet, l'approche proposée a comme objectif d'être efficace même si l'image n'est pas segmentée ou que la segmentation n'est pas précise.

Toute cette partie a fait l'objet de deux articles, dont l'un a été publié [[Decelle et al., 2021b](#)] et l'autre a été accepté (voir [section 6.2](#)), mais n'est pas encore publié<sup>11</sup> (en *pre-print*) (dont une démonstration de la méthode est disponible<sup>12</sup>). Le code source est disponible<sup>13</sup>.

11. <https://www.ipol.im/pub/pre/338/>

12. <https://ipolcore.ipol.im/demo/clientApp/demo.html?id=338>

13. <https://gitlab.com/Ryukhaan/treetrace>

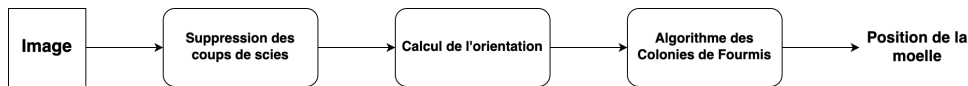


FIGURE 4.10 – Les trois étapes de la méthode proposée.

## 4.2.1 Pré-traitement

La première étape consiste à appliquer un pré-traitement. Il vise à réduire à la fois le temps de calcul et le bruit pour mieux calculer leur orientation. Le pré-traitement consiste à :

- convertir l'image en niveaux de gris ;
- réduire la taille de l'image ;
- supprimer les marques du sciage, principal facteur rendant difficile le calcul de l'orientation des cernes.

### 4.2.1.1 Conversion en niveaux de gris

L'image RVB est convertie en niveaux de gris (la formule usuelle utilisée). La conversion en niveaux de gris est effectuée en utilisant la formule suivante :

$$0,299R + 0,114V + 0,587B$$

où  $R$ ,  $V$  et  $B$  sont respectivement les composantes rouge, vert et bleu. Ensuite, un sous-échantillonnage est effectué en utilisant un facteur  $\delta$  et l'interpolation bilinéaire. Utiliser un facteur permet de conserver le rapport entre hauteur et largeur de l'image. Cela évite de déformer les cernes. Ce sous-échantillonnage vise à réduire à la fois le temps de calcul et le bruit entre les cernes causé à la fois par les outils de sciage (non aiguisés) mais également par notre dispositif d'acquisition qui prend des images de haute résolution permettant d'avoir beaucoup plus d'informations que nécessaire.

### 4.2.1.2 Suppression de coups de scies

Insistons encore une fois, l'une des principales perturbations pour la détection de la moelle sur des images brutes de coupes transversales de grumes est la présence des coups de scies (voir la [figure 4.11a](#)). Comme la détection de la moelle repose sur le calcul de l'orientation des cernes, ces coups de scies interfèrent avec les cernes dans le calcul de l'orientation. Nous proposons une méthode pour réduire leur influence. Celle-ci est similaire à celle proposée par Norell [[Norell, 2011](#)].

Voici notre solution pour supprimer les coups des scies. Tout d'abord, nous pouvons observer que les coups de scies sont généralement alignés dans un secteur circulaire, c'est-à-dire une plage d'angles possibles, mais sont toujours uniformément espacés (voir la [figure 4.11a](#)). Ce motif répétitif suggère qu'un filtrage dans le domaine de Fourier est approprié. En effet, dans le domaine de Fourier, ce motif correspond à une ligne passant par le centre avec une direction perpendiculaire à sa direction. Le long de cette ligne,

l'énergie sera élevée. En réduisant cette énergie et en reconvertissant le spectre filtré dans le domaine spatial, les coups de scies peuvent être supprimés ou du moins réduits (voir la [figure 4.11f](#)).

Nous calculons dans un premier temps la transformée de Fourier (FFT). Si la grume n'est pas segmentée, il est possible d'appliquer un filtre pour réduire le bruit dû à l'arrière-plan ou d'appliquer la méthode dans des sous-images. Puis, un filtre passe-bande est appliqué. Nous supprimons les lignes horizontales et verticales qui sont des régions à haute énergie (voir les [figure 4.11b](#) et [figure 4.11c](#)).

La transformée de Fourier est seuillée. Seules les valeurs supérieures à  $\lambda$  fois la valeur maximale sont conservées (voir la [figure 4.11d](#)). Un nuage de points est obtenu sur lequel nous effectuons une analyse par composantes principales (ACP) pour estimer deux directions orthogonales. Une fois le vecteur propre principal estimé, un filtre défini par une ligne dans la direction obtenue et passant par le centre de la transformée est créé. Ce filtre est convolué par une gaussienne d'écart-type  $\sigma$  (voir [figure 4.11e](#)). Les coups de scies ne sont pas des lignes strictement parallèles, c'est pourquoi le paramètre  $\sigma$  est introduit. Il agit comme marge d'erreur. Cela permet d'épaissir la ligne à supprimer.

Enfin, le filtre est multiplié (pixel par pixel) avec la FFT initiale. L'image finale est obtenue dans le domaine spatial avec la FFT inverse (voir [figure 4.11f](#)).

Parfois, les directions principales du nuage de points ne sont pas orthogonales. Cela conduit l'ACP à mal estimer les directions souhaitées (la méthode recherche des directions orthogonales entre elles). La [figure 4.12a](#) et la [figure 4.12b](#) illustrent ce problème. Les flèches rouges indiquent les directions souhaitées, et les flèches bleues celles estimées par l'ACP. Pour limiter l'impact de ce problème, nous appliquons la suppression si :

$$\frac{\nu_1 - \nu_2}{\nu_1} > 0,66$$

où  $\nu_1$  est la valeur propre de la composante principale et  $\nu_2$  la valeur propre de la deuxième composante. Si la valeur est inférieure à ce seuil, il peut y avoir une autre direction à supprimer. Comme nous ne pouvons pas savoir laquelle supprimer, nous n'appliquons pas la suppression. Cela peut aussi indiquer que les marques de sciage sont peu prononcées, elles ne devraient pas perturber la suite des calculs. La valeur de 0,66 a été choisie empiriquement.

### 4.2.1.3 Calcul de l'orientation

Rappelons l'hypothèse qui est souvent faite : les cernes peuvent être décrits par des cercles concentriques dont la moelle en est le centre. Le point clef de notre méthode est d'accumuler les normales aux tangentes des cernes. C'est la raison pour laquelle, après avoir supprimé les coups de scies, nous devons déterminer l'orientation des tangentes aux cernes.

Nous utilisons une méthode basée sur le gradient  $\nabla I = (\partial_x(I), \partial_y(I))$  de l'image. Pour rappel, l'orientation est calculée avec la formule :

$$\theta = \arctan\left(\frac{\partial_y(I)}{\partial_x(I)}\right)$$

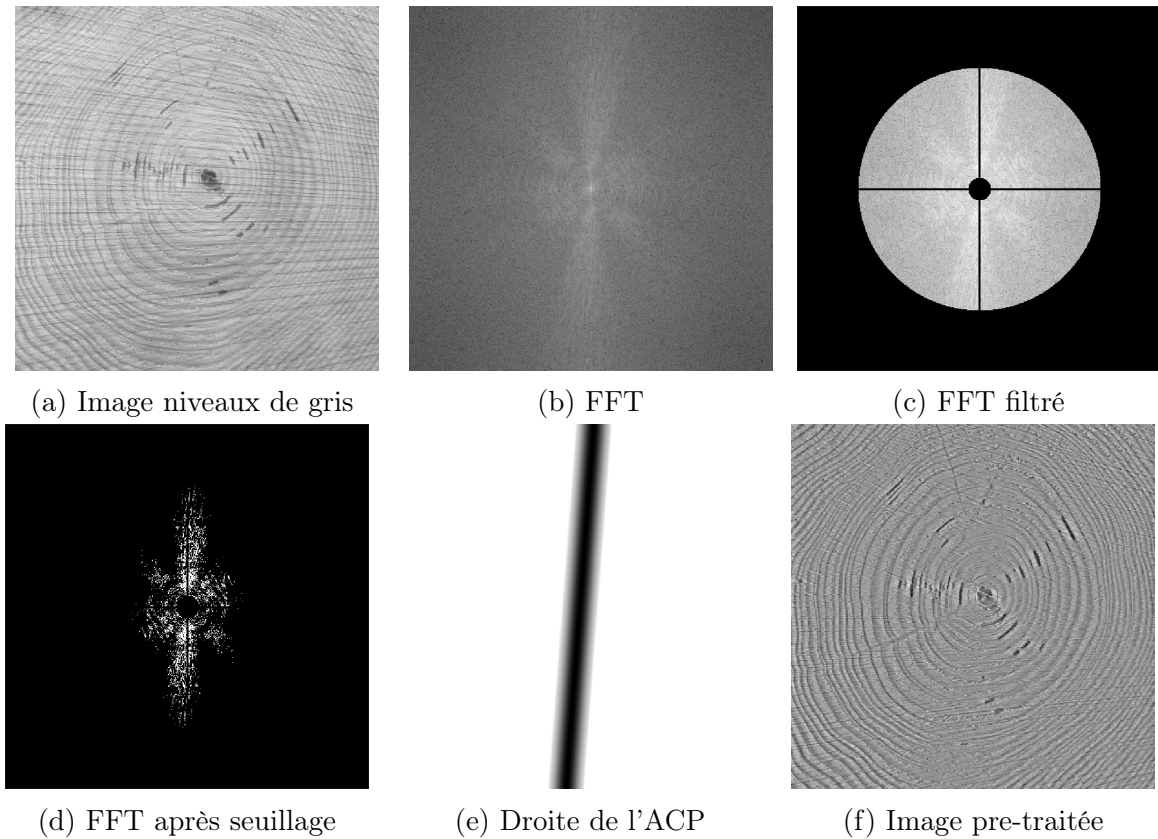


FIGURE 4.11 – Étapes du pré-traitement : (a) Niveaux de gris, (b) FFT, (c) FFT filtré, (d) FFT après seuillage, (e) droite détectée par l'ACP, (f) l'image de sortie.

Nous avons vu que le gradient de ce type n'est pas nécessairement probant (voir le [chapitre 1](#)). Mais il existe des manières plus robuste d'estimer l'orientation en utilisant le gradient.

L'orientation est calculée à l'aide d'un tenseur [[Hong et al., 1998](#)]. Ce type de structure donne une estimation de l'orientation et permet de faire des mesures locales d'anisotropie dans un voisinage local du pixel. Le gradient est calculé par convolution avec la dérivée (masque de Sobel-Feldmann) d'un filtre gaussien  $F$  d'écart-type  $\sigma_g$  et de taille  $6\sigma_g + 1$ . L'utilisation d'une gaussienne réduit l'influence du bruit sur le calcul du gradient. Le tenseur 2-D  $J$  est une version lissée de la matrice du second moment  $(\nabla I)(\nabla I)^T$ . Le tenseur est :

$$J_{\sigma_g} = \begin{bmatrix} \partial_x(I)^2 * F & \partial_x(I)\partial_y(I) * F \\ \partial_y(I)\partial_x(I) * F & \partial_y(I)^2 * F \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \quad (4.5)$$

Par construction  $J$  est une matrice symétrique, positive et semi-définie. Les informations d'anisotropie et d'orientation sont données par la décomposition en valeurs propres de cette matrice. L'orientation est calculée par la formule :

$$\theta = \frac{\pi}{2} + \frac{1}{2} \arctan \left( \frac{2J_{12}}{J_{22} - J_{11}} \right) \quad (4.6)$$

La covariance du tenseur est aussi lissée par un filtre gaussien  $G$ , d'écart-type  $\sigma_b$  et de

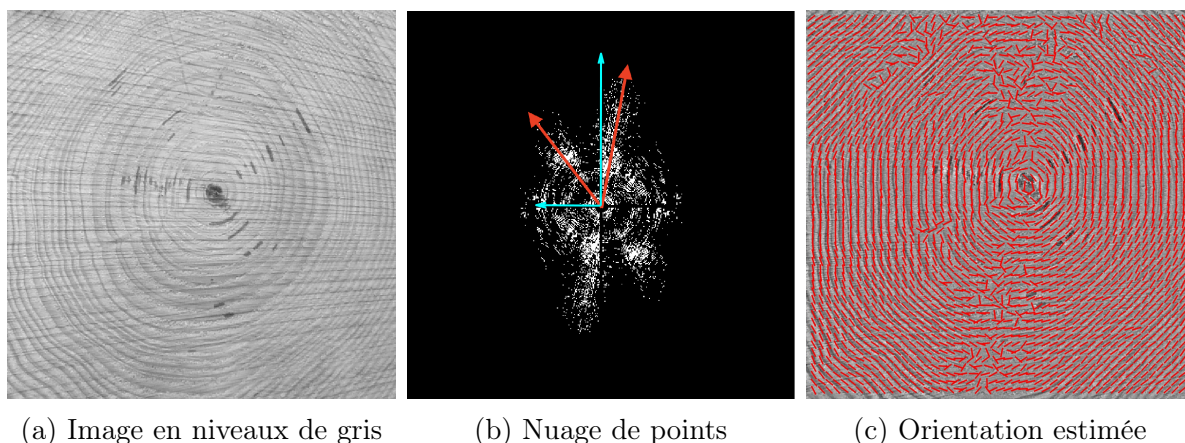


FIGURE 4.12 – Image d’entrée (à gauche) et le nuage de points correspondant lors du pré-traitement (au milieu) où les deux directions principales (flèches rouges) dans le domaine de Fourier sont non-orthogonales. L’ACP ne peut estimer ces deux directions. À droite, le résultat du calcul de l’orientation des cernes (segments rouges). Quelques orientations ne sont pas correctes, cela est dû à la suppression des coups de scies.

taille  $6\sigma_b + 1$ , afin de réduire les erreurs de calcul.

Enfin, il est possible d’effectuer un filtrage post-calcul afin d’homogénéiser l’orientation et d’éviter des sauts de valeurs trop importants. Pour cela, un dernier filtre gaussien  $H$ , d’écart-type  $\sigma_s$  et de taille  $6\sigma_s + 1$ , est appliqué.

La [figure 4.12c](#) montre l’image pré-traitée et l’orientation des cernes. En raison de l’élimination des coups de sciage, quelques orientations de cernes sont mal calculées. Malgré tout, cela ne sera pas gênant pour la suite de la méthode proposée.

À noter que le calcul de l’orientation utilisé est le même que celui fait par [\[Norell and Borgfors, 2008\]](#) mais effectué dans le domaine spatial.

## 4.2.2 Application à la détection de la moelle

Une fois que les orientations des cernes ont été calculées, nous devons accumuler les normales. Près de la moelle, les cernes peuvent être décrits par des cercles concentriques dont la moelle est le centre. Ainsi, les normales devraient s’accumuler de manière à ce que la moelle soit à la position avec le plus d’accumulations. Au lieu de concevoir une heuristique pour accumuler les normales de manière robuste (c’est-à-dire en rejetant les normales mal calculées ou ne prenant que l’orientation défini par un bloc), nous nous appuyons sur les algorithmes des colonies de fourmis (ACO) pour le faire [\[Dorigo et al., 1996, Tian et al., 2008\]](#).

L’idée de la méthode proposée est la suivante. Une fourmi a une position dans l’image à partir de laquelle des normales seront accumulées. Chaque fourmi effectue deux actions : se déplacer selon une carte de phéromones  $\tau$  et déposer des phéromones qui augmentent les valeurs de  $\tau$ . Cette carte sera notre espace d’accumulation. Ensuite, les fourmis se déplaceront plus souvent vers les points avec une grande quantité de phéromones, qui devraient être les zones proches de la moelle. En effet, les normales près de la moelle



vont se couper plus souvent entre elles, augmentant la quantité de phéromones dans cette zone. Les fourmis se déplaceront moins souvent vers les zones éloignées de la moelle (c'est-à-dire les zones avec une faible quantité de phéromones) et accumuleront moins souvent des normales loin de la moelle. Autrement dit, il devrait y avoir convergence vers un point d'accumulation qui devrait être la moelle. Décrivons maintenant en détail le fonctionnement de l'algorithme.

Soit  $K^2$  le nombre de fourmis et  $\tau^{(t)}$  la carte de phéromones à un instant  $t$ . Les deux questions principales de l'algorithme sont : comment établir la matrice de transition probabiliste, dénotée par  $\mathbf{p}^{(t)}$ , servant aux déplacements des fourmis et comment mettre à jour la carte de phéromones  $\tau^{(t)}$ ? Nos choix pour chacune des questions sont présentés ci-dessous.

#### 4.2.2.1 Initialisation

Nous avons  $K^2$  fourmis placées dans une grille uniforme sur l'image pré-traitée. Chaque fourmi peut se déplacer librement. Initialement, la valeur de chaque élément de la carte de phéromones  $\tau^{(0)}$  est fixée à un nombre aléatoire entre 0 et 1 par une distribution normale. Une distribution normale semble accélérer la convergence de l'algorithme.

Un élément de la matrice  $\tau^{(t)}$  représente  $m \times m$  pixels dans l'image pré-traitée. En d'autres termes, si l'image pré-traitée  $\mathbf{I}$  est de taille  $H \times W$  alors  $\tau^{(t)}$  est de taille  $\frac{H}{m} \times \frac{W}{m}$ . Dans la suite, nous dénoterons les coordonnées de la matrice  $\tau^{(t)}$  par  $(u, v)$  et ceux de l'image par  $(x, y)$ . La relation entre un pixel  $(x, y)$  et un élément  $(u, v)$  est :

$$(u, v) = \left( \left\lfloor \frac{x}{m} \right\rfloor, \left\lfloor \frac{y}{m} \right\rfloor \right)$$

Une valeur élevée pour  $m$  réduit le temps de calcul mais cela implique de perdre en précision. A contrario, une valeur faible de  $m$  augmente le temps de calcul, mais améliore la précision.

#### 4.2.2.2 Construction de la matrice de transition probabiliste

Une fourmi se déplace de manière aléatoire avec une certaine probabilité. La probabilité  $\mathbf{p}_{u,v}$  pour une fourmi de se déplacer vers l'élément  $(u, v)$  est définie par :

$$\mathbf{p}_{u,v} = \frac{(\tau_{u,v})^\alpha (\eta_{u,v})^\beta}{\sum_{i,j} (\tau_{i,j})^\alpha (\eta_{i,j})^\beta} \quad (4.7)$$

où  $\tau_{u,v}$  est la quantité de phéromones à la position  $(u, v)$ ,  $\eta_{u,v}$  est la désirabilité de l'élément  $(u, v)$ ,  $\alpha$  et  $\beta$  sont respectivement des paramètres pour contrôler l'influence de  $\tau_{u,v}$  et  $\eta_{u,v}$ .

La désirabilité de la position  $(u, v)$  est égale à l'inverse de la distance entre cette position et la position  $(a, b)$  où se trouve la fourmi. Ainsi, la désirabilité est égale à

$$\eta_{u,v}(a, b) = \frac{1}{\sqrt{(u-a)^2 + (v-b)^2 + 1}}.$$

La désirabilité peut être vue comme une pondération de la carte des phéromones. Elle vise à s'assurer que les fourmis ont une probabilité plus élevée de se déplacer vers des

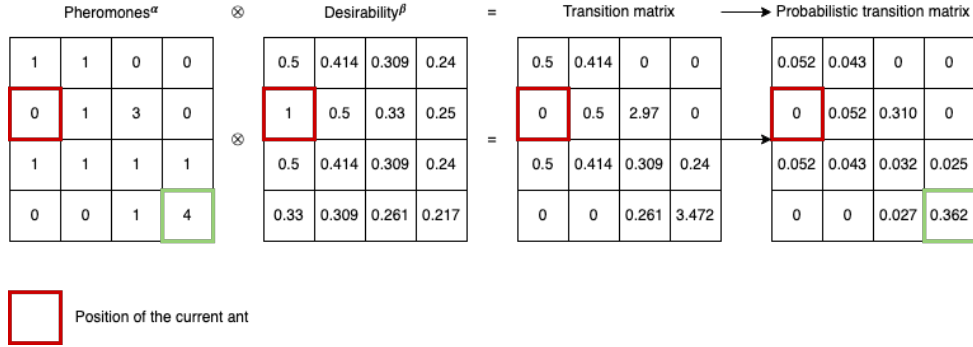


FIGURE 4.13 – D’après [Decelle et al., 2021b], exemple du calcul de la matrice de transition probabiliste  $\mathbf{p}$ . Le carré rouge représente la position actuelle de la fourmi. Après le calcul, la fourmi a une probabilité de 36,2% de se déplacer vers la position contenant le plus de phéromones (carrés verts). Le résultat est obtenu avec  $\alpha = 2$  et  $\beta = 1$ .

maxima locaux et non vers un maximum global qui pourrait être une mauvaise estimation de la moelle. Autrement dit, les fourmis doivent privilégier des petits déplacements.

Le ratio  $\frac{\alpha}{\beta}$  permet de modifier le comportement des fourmis. Une valeur élevée conduit les fourmis à se déplacer plus rapidement vers les pics de phéromones. Une valeur faible conduit les fourmis à continuer à explorer les zones en image.

#### 4.2.2.3 Mise à jour de la carte de phéromones

La carte de phéromones est mise à jour de la manière suivante :

$$\tau^{(t+1)} = (1 - \gamma)\tau^{(t)} + \sum_{k \in \{0, \dots, K^2 - 1\}} \rho_k^{(t)} \quad (4.8)$$

où  $\rho_k^{(t)}$  est la matrice des phéromones déposées par la  $k^{i\text{eme}}$  fourmi à l’instant  $t$  et  $\gamma$  est le taux d’évaporation des phéromones. Le taux d’évaporation permet d’éliminer les accumulations erronées au fur et à mesure du processus. Il vise à supprimer les pics locaux de phéromones. Plus la valeur  $\gamma$  est élevée, plus les phéromones sont éliminées rapidement. La mise à jour n’intervient qu’une fois que toutes les fourmis se sont déplacées.

La figure 4.13 montre comment calculer  $\mathbf{p}$  pour une fourmi en fonction de  $\tau$  (carte de phéromones) et de  $\eta$  (matrice de désirabilité).

### 4.2.3 Dépôt des phéromones

Détaillons maintenant comment une fourmi dépose des phéromones. Pour chaque fourmi  $k$ , la matrice  $\rho_k^{(t)}$  est calculée avant de faire le déplacement. Mais, elle n’est pas ajoutée tout de suite à la matrice des phéromones  $\tau^{(t)}$ . L’ajout ne se fait qu’après avoir déplacé toutes les fourmis. Le calcul de  $\rho_k^{(t)}$  est le suivant.

Chaque fourmi est le centre d’un ensemble de blocs d’images. Cet ensemble est constitué de  $n \times n$  blocs et chaque bloc a une taille de  $\omega \times \omega$  pixels. En effet, pour chaque bloc, la valeur médiane de l’orientation est prise comme orientation pour le bloc. Ainsi,

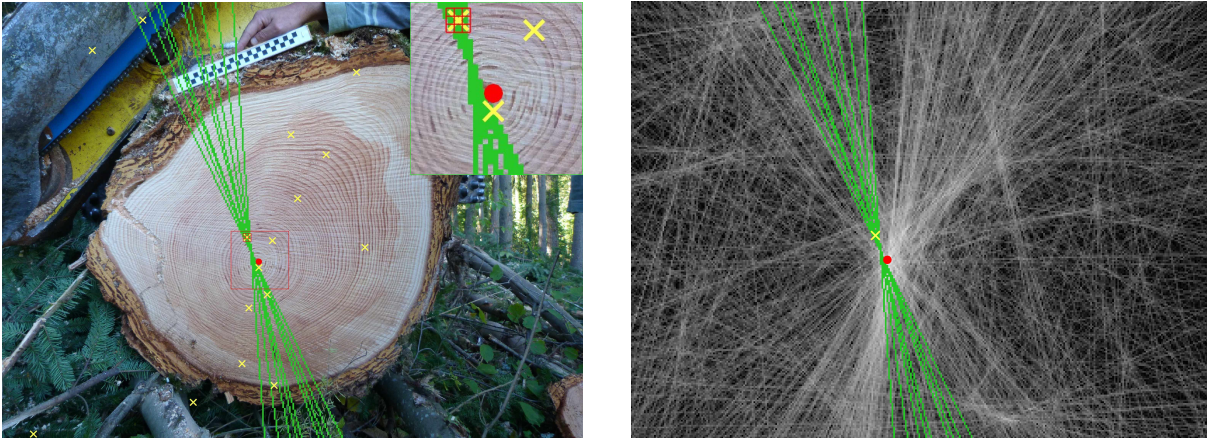


FIGURE 4.14 – À gauche, image d’origine avec les fourmis en croix jaunes, l’estimation courante de la moelle en rouge et un ensemble de  $n = 3$  blocs de taille  $\omega = 8$  (grille rouge). Les normales servant d’accumulation sont les lignes vertes. En haut à droite, un zoom sur le rectangle rouge. À droite, la matrice de phéromones. Une position  $(u, v)$  est égal à  $m = 5$  pixels en coordonnées image.

le paramètre  $\omega$  agit comme un lissage car l’orientation locale a pu être mal calculée. Une petite valeur revient à considérer que l’orientation a été correctement calculée. Toutes ces étapes sont effectuées en coordonnées  $(x, y)$  de l’image. La position  $(x, y)$  du centre du bloc est convertie en coordonnées  $(u, v)$ . Pour rappel, la relation entre  $(x, y)$  et  $(u, v)$  est :

$$(u, v) = \left( \left\lfloor \frac{x}{m} \right\rfloor, \left\lfloor \frac{y}{m} \right\rfloor \right)$$

où  $m$  est un paramètre. Ensuite, une ligne est tracée selon l’orientation et passant par  $(u, v)$ . Tous les indices de  $\rho_k^{(t)}$  le long de la droite sont incrémentés de 1 (en utilisant l’algorithme de Bresenham pour tracer la droite).

La figure 4.14 montre, à gauche, une fourmi (croix jaune) avec son ensemble de blocs (la grille rouge). Pour chaque bloc, une ligne est tracée en vert en fonction de son orientation. L’estimation actuelle de la position de la moelle est le point rouge. À droite, les mêmes objets (avec la même couleur) sont représentés en coordonnées  $(u, v)$ . L’estimation de la position de la moelle est le point rouge. L’image de droite représente la matrice des phéromones  $\tau$ , avec la fourmi en jaune, les droites selon l’orientation des blocs en vert (comme sur l’image de gauche). Plus la quantité de phéromones en un pixel est grande, plus l’intensité est élevée (pixel blanc).

#### 4.2.4 Extraction de la position de la moelle

Il ne reste plus qu’à détailler comment la position de la moelle est estimée. Pour extraire de la matrice  $\tau$  la position de la moelle, nous prenons le barycentre de tous les pixels situés au-dessus de  $\kappa * \max(\tau)$ . Prendre le maximum est trop risqué et est moins robuste que le barycentre des valeurs les plus élevées. Plus  $\kappa$  est élevé, plus l’estimation de la moelle est sensible aux petites variations.

<b>Besle</b>	Moyenne	Écart-type	Min	Max	Temps (en ms)
[Schraml and Uhl, 2013]	<b>2,29</b>	<b>0,98</b>	<b>0,39</b>	<b>4,96</b>	8344
[Kurdthongmee et al., 2018]	25,06	21,23	2,15	92,44	667
YOLO v3 [Kurdthongmee, 2020]	2,88	1,67	0,87	7,68	<b>138</b>
Notre méthode	2,34	1,02	0,46	5,04	910
<b>BBF</b>					
[Schraml and Uhl, 2013]	2,39	1,48	0,49	7,59	8660
[Kurdthongmee et al., 2018]	38,38	38,91	3,14	232,74	721
YOLO v3 [Kurdthongmee, 2020]	12,69	53,55	0,50	341,92	<b>186</b>
Notre méthode	<b>2,26</b>	<b>1,32</b>	<b>0,44</b>	<b>4,63</b>	1055

TABLE 4.2 – Moyenne, écart-type, minimum et maximum entre les vérités de terrain et l’estimation de la position de la moelle pour notre méthode et les méthodes de [Schraml and Uhl, 2013, Kurdthongmee et al., 2018, Kurdthongmee, 2020] (en mm) et temps moyen pour traiter une image (en ms).

Nous introduisons un critère d’arrêt. À chaque itération, nous calculons la distance entre l’estimation actuelle et la dernière estimation de la moelle. L’algorithme s’arrête dès que la moyenne des cinq dernières distances tombe sous une valeur  $\varepsilon$ . Quoiqu’il arrive, l’algorithme s’arrête après un maximum de  $N$  itérations.

### 4.3 Résultats

Les résultats ont été fait en 2019 sur les jeux de données acquis à ce moment là. Les deux jeux d’images utilisés sont **Besle** et **BBF** (voir le chapitre 2). La méthode de [Kurdthongmee, 2020] utilise des réseaux de neurones, nous avons utilisé YOLO v3 car ce dernier avait les meilleurs résultats. Nous avons repris le réseaux entraîné sur leur jeux d’images et nous avons sélectionnées 10% des images de chaque jeu afin d’effectuer un apprentissage sur nos images (soit 6 images pour le jeu **Besle** et 4 images du jeu **BBF**). Ces 10 images ne sont pas prises en compte dans les résultats présentés.

Le tableau 4.2 présente une comparaison sur nos jeux de données entre notre méthode et trois algorithmes présentés : [Schraml and Uhl, 2013] (la méthode avec les meilleures résultats), [Kurdthongmee et al., 2018] et [Kurdthongmee, 2020] (YOLO v3). La méthode d’apprentissage profond est la plus rapide. Les inconvénients sont le temps d’apprentissage et la nécessité de créer d’énormes jeux de données avec des vérités terrain de base. Notre méthode est en moyenne 5 fois plus rapide que [Schraml and Uhl, 2013] et peut être facilement parallélisée. Nous pouvons observer que notre méthode et [Schraml and Uhl, 2013] sont plus précises que [Kurdthongmee, 2020, Kurdthongmee et al., 2018]. Les résultats de [Kurdthongmee, 2020] sont moins bons sur le jeu d’images **BBF**, cela peut être dû au petit nombre d’images qu’il contient.

La figure 4.15 montre les boîtes à moustaches pour comparer notre méthode, celle de [Schraml and Uhl, 2013] et celle de [Kurdthongmee, 2020]. Cela permet de mieux illustrer les différences entre ces méthodes. Nous avons exclu [Kurdthongmee et al., 2018] car leurs

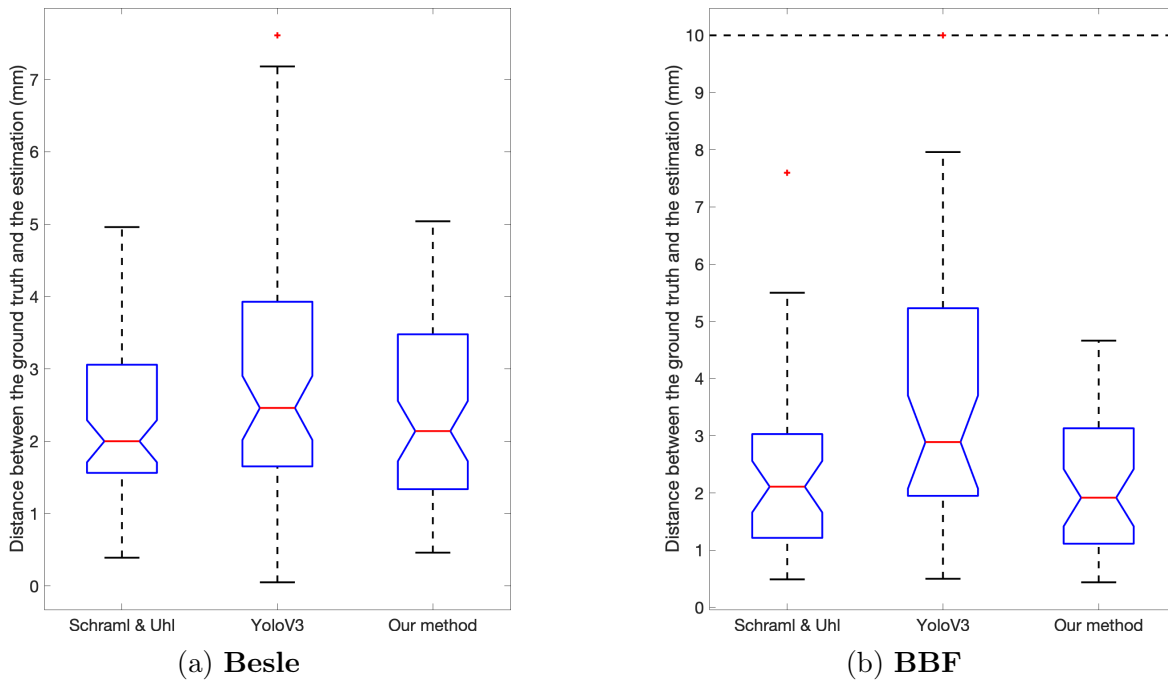


FIGURE 4.15 – Boîtes à moustaches représentant la distance entre l’estimation et la vérité terrain (en mm) pour les méthodes [Schraml and Uhl, 2013, Kurdthongmee et al., 2018, Kurdthongmee, 2020] et notre méthode.

résultats sont beaucoup moins précis que les trois autres.

Pour le jeu d’images **Besle**, notre méthode est un peu moins précise que [Schraml and Uhl, 2013]. La méthode de [Kurdthongmee, 2020] est un peu moins précise de manière générale. La méthode donne une estimation très imprécise pour une image (7,61 mm). Le premier quartile et le troisième quartile sont plus élevés pour cette méthode par rapport à la notre et celle de [Schraml and Uhl, 2013].

Pour le jeu d’images **BBF**, [Schraml and Uhl, 2013] présente une valeur peu précise (7,60 mm) et [Kurdthongmee, 2020] quatre valeurs très imprécises (supérieures à 10 mm). Hormis ces cas, les réseaux de neurones sont proches de nos résultats et de ceux de [Schraml and Uhl, 2013].

La figure 4.16 montre un exemple de chaque jeux de données de l’estimation de la moelle pour chacune méthode comparée. Ces deux images sont les pires cas pour notre méthode. La croix blanche est la vérité terrain. La croix verte est l’estimation par la méthode [Kurdthongmee, 2020] (YOLO v3), la croix bleu est l’estimation par la méthode [Schraml and Uhl, 2013] et la croix rouge est l’estimation de notre méthode. Nous constatons que, même dans le pire pour notre méthode, l’estimation reste proche de la vérité terrain. Nous pouvons observer que la méthode avec les réseaux de neurones a détecté la moelle de la deuxième grume (à droite de l’image de gauche). Malheureusement ce n’est pas la moelle souhaitée (celle-ci n’est pas cadrée).



FIGURE 4.16 – Deux exemples où notre méthode n’arrive pas à moins de 5 mm de la vérité terrain. La vérité terrain est la croix blanche, la croix verte est le résultat du réseau YOLO v3 [Kurdthongmee, 2020], celle en bleue montre la moelle détectée par [Schraml and Uhl, 2013], et notre méthode est représentée par la croix rouge.

## 4.4 Analyse de la méthode proposée

Dans cette section, nous proposons de montrer l’influence de certains paramètres de notre méthode sur le temps de calcul. Nous en profitons pour présenter un calcul de la complexité.

### 4.4.1 Analyse de la complexité

Bien que notre méthode paraisse complexe et donc potentiellement coûteuse en temps de calcul, sa complexité dépend fortement de deux facteurs : le nombre de fourmis et la taille de l’image. Pour rappel, le nombre de fourmis est  $K^2$  et la taille de l’image  $H \times W$  pixels. D’abord, la matrice de désirabilité  $\eta$  et la matrice de transition probabiliste  $\mathbf{p}$  associé à la fourmi  $k$  se calcul en  $\mathcal{O}(\frac{HW}{m^2})$ .

Puis, pour le dépôt de phéromones nous devons nous rappeler que chaque fourmi est le centre d’un ensemble de blocs d’image qui consiste en  $n \times n$  blocs et chaque bloc est de taille  $\omega \times \omega$  pixels. Par conséquent, pour estimer l’orientation du bloc, nous calculons  $n^2$  fois la médiane d’un bloc de  $\omega^2$  pixels. Cette opération est effectuée en triant le tableau en  $\mathcal{O}(n^2\omega^2 \log \omega)$ . Nous mettons à jours  $n^2$  fois la matrice de phéromones  $\rho_k^t$  pour chaque bloc. Le nombre de pixels à mettre à jour est  $\mathcal{O}(\max(\frac{H}{m}, \frac{W}{m}))$ . En d’autres termes, la mise à jour de la matrice de phéromones est effectuée en  $\mathcal{O}(n^2 \max(\frac{H}{m}, \frac{W}{m}))$ .

Enfin, la position de la fourmi est mise à jour en  $\mathcal{O}(1)$ . Une fois que chaque fourmi s’est déplacée, la matrice des phéromones  $\rho$  est mise à jour en  $\mathcal{O}(\frac{HW}{m^2})$ . Par conséquent, la complexité temporelle totale pour une itération est :

$$\mathcal{O}\left(K^2 \left[ n^2\omega^2 \log \omega + n^2 \max\left(\frac{H}{m}, \frac{W}{m}\right) + \frac{1}{m^2}HW + 1 \right] + \frac{1}{m^2}HW \right) \quad (4.9)$$

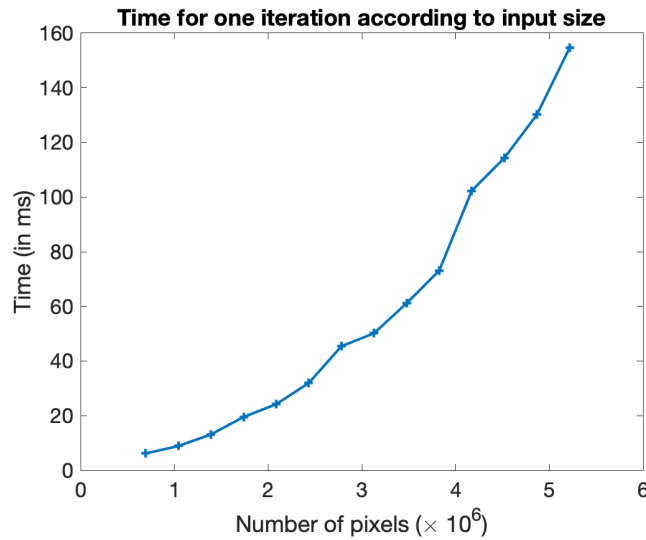


FIGURE 4.17 – Temps de calcul pour une itération selon la taille (nombre de pixels) de l'image.

En supposant que  $H > W$  et en ne gardant que les paramètres principaux à savoir  $K$ ,  $H$  et  $W$ , nous pouvons simplifier la formule par :

$$\mathcal{O}(K^2 [H + H^2 + 1] + H^2) = \mathcal{O}(K^2 H^2) \quad (4.10)$$

Validons maintenant cette complexité théorique par les expériences. Tout d'abord, nous pouvons voir dans l'équation 4.10 que la méthode est linéaire par rapport à la taille de l'image d'entrée  $I$ . La figure 4.17 ne montre cependant pas cette linéarité. Nous avons le temps d'exécution pour une itération en fonction de la taille de  $I = H \times W$ . Cette non-linéarité est potentiellement due à la simplification de la complexité, notamment vis-à-vis de la taille de l'image. L'équation 4.9 montre la complexité plus détaillée.

D'après l'équation 4.9, la méthode a une complexité temporelle linéaire par rapport au nombre de fourmis  $K^2$ . En effet, la figure 4.18 montre le temps de calcul en fonction de  $K$ . Le nombre de fourmis  $K \times K$  augmente quadratiquement. C'est la raison pour laquelle le temps de calcul augmente également de façon quadratique.

Toujours dans l'équation 4.9, le temps de calcul diminue lorsque  $m$  augmente. Et ceci est proportionnel à l'inverse du carré de  $m$ . La figure 4.18 montre le temps de calcul en fonction de  $m$ . Nous constatons qu'avoir une valeur de  $m$  supérieure à 1 est très utile pour réduire le temps de calcul. En effet, la matrice  $\tau$  est largement utilisée au cours du processus, lors du dépôt de phéromones, de la mise à jour des phéromones. Ainsi diminuer  $m$  réduit la taille de  $\eta$ ,  $\tau$  et  $\rho$ . Examinons maintenant les deux paramètres suivants : la taille du bloc  $\omega \times \omega$  et le nombre de blocs  $n \times n$ .

Le temps de calcul en fonction de la taille des blocs est montré dans la figure 4.18. Comme le nombre de blocs est au moins égal à un, le temps de calcul ne commence pas à 0. Pour une taille de bloc de  $3 \times 3$  il faut en moyenne 243 ms, alors que pour une taille de bloc de  $11 \times 11$  il est de 259 ms. Notons que le temps de calcul dépend de la taille de l'image. Pour  $n$ , il devrait être quadratique. Comme  $n$  doit être au moins égal à un, le

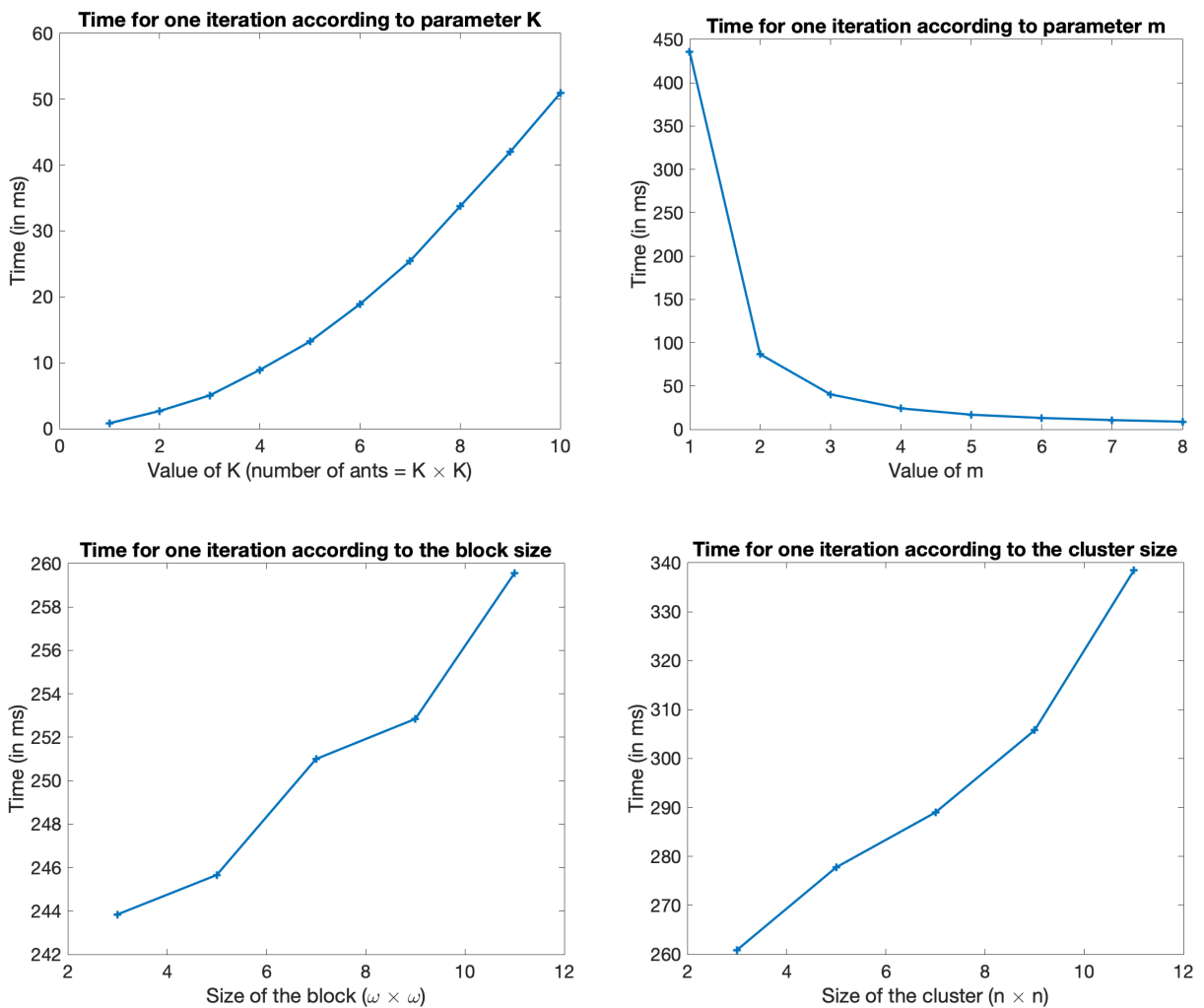


FIGURE 4.18 – Temps de calcul pour une itération selon (a) le nombre de fourmis  $K$ , (b) la taille de  $\tau$ , (c) la taille  $\omega$  d'un bloc et (d) le taille  $n$  du nombre de blocs.

temps de calcul ne commence pas à 0. La [figure 4.18](#) montre le temps de calcul en fonction de  $n$ , et il est proche d'être quadratique.

#### 4.4.2 Effet des paramètres sur la convergence de l'algorithme

Les algorithmes des colonies de fourmis sont probabilistes, il est donc nécessaire de montrer qu'il y a bien convergence vers une solution (qu'elle soit celle attendue ou non) [Dorigo and Blum, 2005]. C'est pourquoi, dans cette sous-section, nous analysons l'influence des paramètres sur la convergence de l'algorithme. Nous n'avons pas de preuve formelle de la convergence de notre algorithme. Ainsi, pour savoir s'il y a convergence, nous avons fait des tests empiriques.

Plus précisément, l'algorithme converge si la distance euclidienne entre deux estimations successives ne varie plus. Nous calculons les variations moyennes à chaque itération



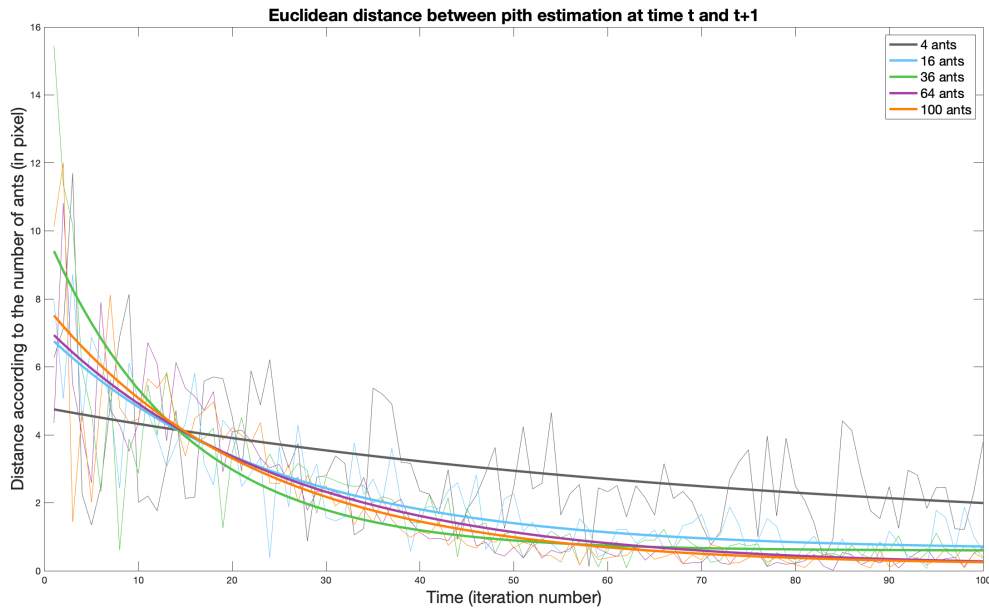


FIGURE 4.19 – Variations selon le paramètre  $K$  entre la position de la moelle à l'instant  $t$  et celle à l'instant  $t + 1$ .

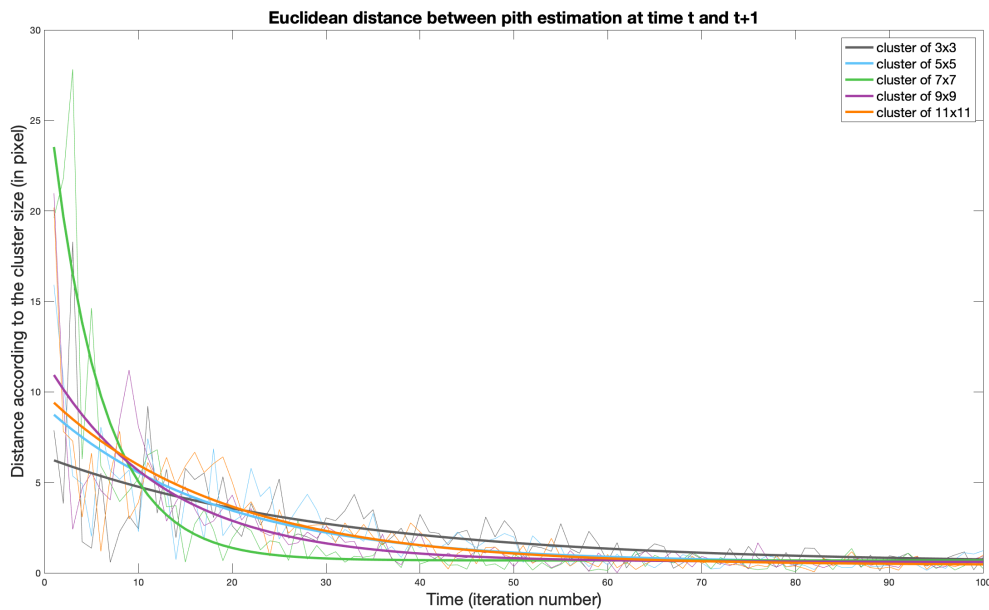


FIGURE 4.20 – Variations selon le paramètre  $n$  entre la position de la moelle à l'instant  $t$  et celle à l'instant  $t + 1$ .

sur l'ensemble des images. Afin de mieux visualiser la convergence, nous ajustons une courbe de la forme  $ae^{bx} + c$ , où  $x$  est la distance calculer,  $a, b$  et  $c$  sont les paramètres

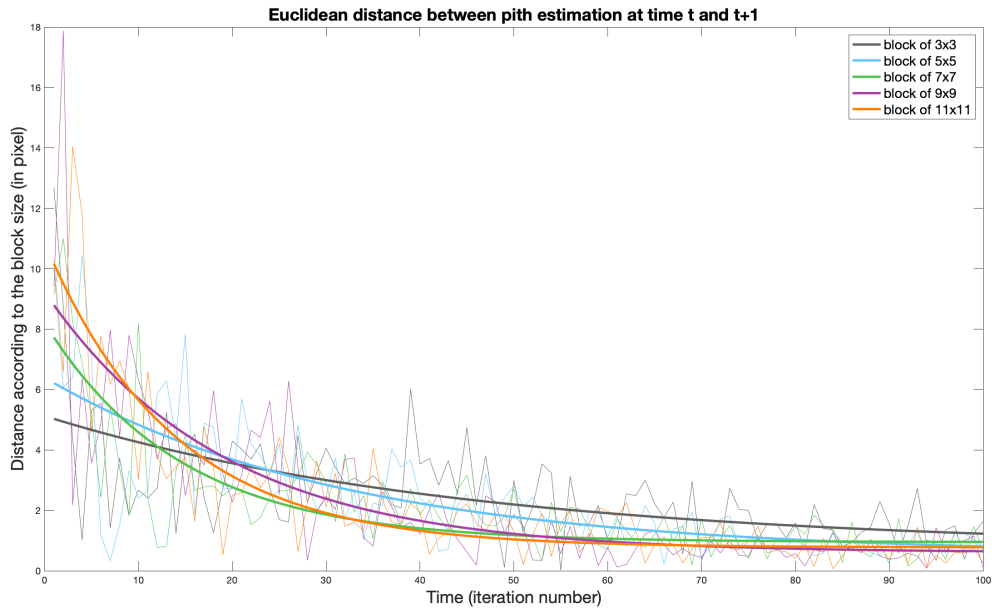


FIGURE 4.21 – Variations selon le paramètre  $\omega$  entre la position de la moelle à l'instant  $t$  et celle à l'instant  $t + 1$ .

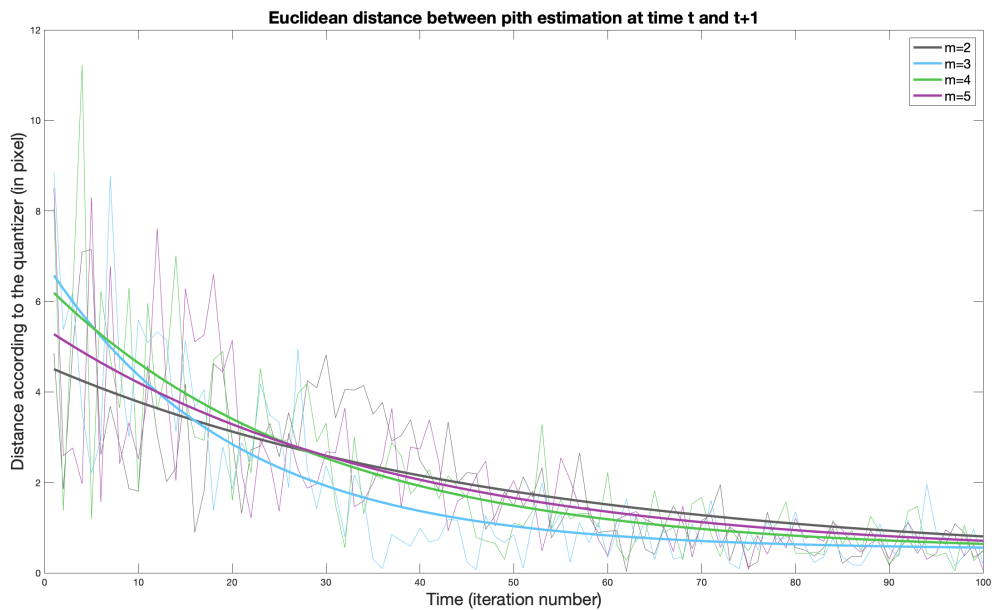


FIGURE 4.22 – Variations selon le paramètre  $m$  entre la position de la moelle à l'instant  $t$  et celle à l'instant  $t + 1$ .

à ajuster et  $e$  désigne la fonction exponentielle. L'ajustement est fait en utilisant les moindres carrés non-linéaires avec l'algorithme des régions de confiance (*Trust Region*

*Methods* en anglais) dont les valeurs initiales sont  $a = \frac{\max - \min}{2}$ ,  $b = 0$  et  $c = 0$ , où  $\max$  (resp.  $\min$ ) est la distance la plus grande (resp. la plus petite). Cet ajustement permet de mettre en évidence le comportement de l'algorithme.

Tout d'abord, nous regardons la convergence en fonction du nombre de fourmis  $K \times K$ . La [figure 4.19](#) montre la vitesse de convergence en fonction de  $K$ . On peut voir que plus le nombre de fourmis est élevé, plus l'algorithme converge rapidement. Avec seulement quatre fourmis, l'algorithme fluctue entre certaines positions. Cependant, un nombre élevé de fourmis n'accélère pas la convergence, mais rend le point de convergence plus stable (le paramètre  $c$  est plus faible avec une valeur élevée de  $K$ ).

Regardons maintenant la convergence selon la taille des bloc  $\omega$ . La [figure 4.21](#) montre la vitesse de convergence en fonction de  $\omega$ . Il semble qu'un grand bloc accélère la convergence, mais pas aussi fortement que  $K$ . On observe qu'une forte augmentation de  $\omega$  semble ralentir la convergence. Ceci est dû à une augmentation des phéromones déposées. En effet, plus  $\omega$  est élevé, plus il y a de *mauvaises* phéromones (n'indiquant pas la position de la moelle). Par conséquent, il faut plus d'itérations pour éliminer ces *mauvaises* phéromones.

Observons maintenant la convergence en fonction du nombre de blocs  $n$ . La [figure 4.20](#) montre la vitesse de convergence en fonction de  $n$ . Contrairement à ce que nous pourrions penser, un grand nombre de blocs ne conduit pas à une convergence plus rapide, mais il ralentit l'algorithme. Cela pourrait être dû à une orientation locale moins précise avec des blocs plus grands.

Enfin, nous regardons l'influence de  $m$ . Pour rappel, plus  $m$  est élevé, plus  $\tau$  est petit. La [figure 4.22](#) montre la vitesse de convergence en fonction de  $m$ . Il apparaît que  $m$  accélère légèrement la convergence. Augmenter la valeur de  $m$  provoque un léger ralentissement de la convergence. Cela pourrait s'expliquer par le fait qu'avec un petit  $\tau$  les petites variations de l'orientation locale ne sont pas prises en compte lorsque les fourmis déposent des phéromones.

## 4.5 Synthèse

Tout d'abord, nous avons résumé les méthodes existantes, traitant le même type d'image que celles du projet TreeTrace, qui estiment la position de la moelle. Nous avons vu que les méthodes peuvent se synthétiser par trois grandes étapes : calcul de l'orientation des cernes, création d'un espace d'accumulations et extraction de la position selon cet espace. Une seule étude [[Kurdthongmee, 2020](#)] propose d'utiliser des réseaux de neurones convolutifs pour estimer la position de la moelle. Cette méthode a une précision de l'ordre de quelques millimètres, environ 4 mm (les résultats sont exprimés en pixels). Le temps d'exécution (hors apprentissage) est de 1 seconde.

Les autres méthodes se sont principalement concentrées sur différentes méthodes existantes pour calculer l'orientation : pyramide laplacienne, filtre en quadrature, analyse de la transformée de Fourier ou histogramme des gradients orientés. Le [tableau 4.3](#) récapitule les informations concernant ces méthodes.

Nous avons proposé une nouvelle approche. Notre calcul de l'orientation est proche de celui de [[Norell and Borgfors, 2008](#)] (filtre en quadrature). Mais là où se distingue notre

	[Norell and Borgefors, 2008]	[Schraml and Uhl, 2013]	[Kurdthongmee and Suwammarat, 2019]	Notre méthode
Orientation	<ul style="list-style-type: none"> <li>• Filtre en quadrature</li> <li>• Pyramide Laplacienne</li> </ul>	Analyse FFT	HOGs	Tenseur
Accumulation	Blocs de <i>confiance</i>	<ul style="list-style-type: none"> <li>• Tous les blocs</li> <li>• RANSAC [Österberg et al., 2004]</li> </ul>	Tous les blocs	Algorithme des colonies de fourmis
Extraction	Barycentre des points de forte accumulations	Barycentre des points de forte accumulations	<ul style="list-style-type: none"> <li>• Barycentre pondéré des points de forte accumulations</li> <li>• Point avec le maximum d'accumulations ([Kurdthongmee et al., 2018])</li> </ul>	Barycentre des points de forte accumulations
Précision (moyenne)	de 1 à 10 mm	de 3 à 8 mm	de 6 à 17 mm	de 1 à 6 mm
Temps	Estimé à 3 secondes (sans benchmark)	de 1 à 2 secondes	de 1 à 32 secondes	de 1 à 2 secondes

TABLE 4.3 – Résumé des différentes méthodes pour estimer la moelle

méthode, c'est qu'elle se focalise sur la manière d'accumuler les droites. Cette accumulation s'inspire des algorithmes des colonies de fourmis. Notre méthode a montré de bons résultats. Nous obtenons la même précision que la méthode de [Schraml and Uhl, 2013] et elle plus précise qu'un des réseaux de [Kurdthongmee, 2020]. De plus, nous avons un temps de calcul réduit. En effet, la méthode [Schraml and Uhl, 2013] se fait en 8 secondes sur nos images, alors que notre approche effectue le calcul en 1 seconde, c'est-à-dire le même temps que le réseau de neurones.



# Filtre directionnel discret

## Sommaire

---

<b>5.1</b>	<b>Introduction</b>	<b>162</b>
<b>5.2</b>	<b>Notions de base</b>	<b>165</b>
5.2.1	Droites et segments discrets	165
5.2.2	Arbre de Stern-Brocot et suite de Farey [G. Hardy, 1989, Stern, 1858]	166
<b>5.3</b>	<b>Filtre directionnel : le cas binaire</b>	<b>167</b>
5.3.1	Calcul des segments naïfs symétriques	167
5.3.2	Complexité en temps	168
5.3.3	Application à l'estimation de caractéristiques	169
<b>5.4</b>	<b>Filtre directionnel : image en niveaux de gris</b>	<b>172</b>
5.4.1	Première approche	173
5.4.2	Inclusion de l'épaisseur	173
5.4.3	Simplification du calcul pour l'épaisseur	176
5.4.4	Estimation de l'orientation	178
5.4.5	Application pour le filtrage d'images	180
<b>5.5</b>	<b>Conclusion</b>	<b>181</b>

---

## 5.1 Introduction

Nous avons proposé, dans les chapitres précédents, un ensemble d'outils pour l'estimation de caractéristiques nécessaire pour évaluer la qualité d'une grume de bois. Nous avons vu que les cernes de croissance sont importants puisqu'il existe un lien de corrélation entre leur largeur et la densité. La densité influence les propriétés physiques de la grume qui a pour conséquences de modifier la qualité de la grume. La [figure 5.1](#) montre une façon de procéder pour estimer les caractéristiques liées aux cernes. La première étape est de délimiter leurs limites (changement entre bois d'été et bois de printemps). Nous avons abordé cette question dans le [chapitre 3](#). À partir de cette détection, il est possible de récupérer les cernes un à un. Si l'on prends un cerne en particulier, nous pouvons le voir comme un objet discret composé de pixels. Sur cet objet, notre objectif est d'estimer un certain nombre de caractéristiques : la largeur, la longueur, l'orientation, etc. Ces caractéristiques peuvent être estimées en chaque point de l'objet, ici le cerne. C'est dans ce cadre-ci que nous abordons la problématique de l'estimation de longueur, largeur et d'orientation des cernes. Pour cela, nous aborderons la question à travers l'utilisation d'outils issus de la géométrie discrète et de la morphologie mathématique.

Ainsi, ce chapitre s'intéresse aux notions de droites discrètes. Nous avons évoqué, dans le [chapitre 3](#) sur la détection des cernes, une méthode avec des outils de géométries discrètes. Cette méthode permet d'estimer l'orientation locale des cernes. Nous avons utilisé une information relative à cela, à savoir la variance, pour en déduire une qualité de la détection des limites de cernes. Cette qualité nous a permis d'omettre certaine estimation dans le calcul de la largeur moyenne des cernes.

Nous avons montré, à travers cet exemple, l'intérêt des outils de géométrie discrète dans l'analyse de coupe transversale de grume de bois. Au-delà de l'orientation, ces outils peuvent également estimer d'autres caractéristiques, comme l'épaisseur, la longueur, etc. Formulé autrement, ces outils vont nous permettre d'analyser minutieusement les cernes, et ainsi nous fournir davantage d'informations concernant les cernes. In fine, nous allons pouvoir extraire plus d'informations sur la qualité de la grume.

Il s'agit maintenant d'entrer dans les détails des outils de géométrie discrète proposés.

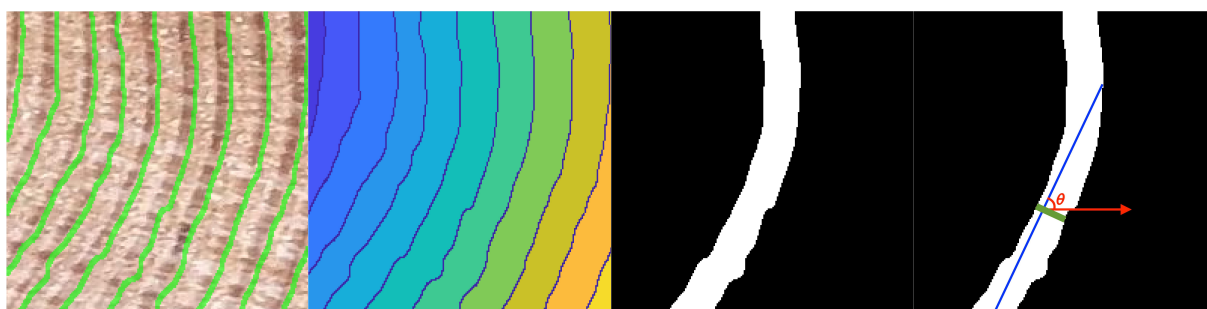


FIGURE 5.1 – Processus d'estimation des caractéristiques d'un cerne de croissance : détection des limites de cernes, labellisation, analyse d'un cerne en tant qu'objet discret, estimation de la longueur, de l'épaisseur et de l'orientation  $\theta$ .

Dans le domaine de la géométrie discrète [Klette and Rosenfeld, 2004], de nouvelles définitions mathématiques des objets géométriques de base ont été introduites pour mieux s'adapter à ces données discrètes, en particulier la notion de segments de droites discrètes (DSS) [Reveillès, 1991, Klette and Rosenfeld, 2004]. Les DSS ont été utilisés dans de nombreux contextes différents pour étudier et analyser les caractéristiques géométriques de courbes extraites à partir du contour d'objets d'une image [Dorksens-Reiter and Debled-Rennesson, 2006, Lachaud et al., 2007, Provençal and Lachaud, 2009, Lachaud, 2010, Nasser et al., 2018], comme la détection des parties convexes et concaves de l'objet, les estimateurs géométriques discrets : longueur, tangente, courbure, ...

Dans le cadre des filtres directionnels utilisant des outils de géométrie discrète et de morphologie mathématique, plusieurs travaux ont été proposés : [van Herk, 1992, Soille et al., 1996, Soille and Talbot, 2001, Merveille et al., 2017, Merveille et al., 2018]. Van Herk a présenté un algorithme efficace pour les filtres locaux min / max utilisant un élément structurant linéaire.

Soille et al. [Soille et al., 1996] ont proposé par la suite une implémentation généralisée de cette méthode pour l'érosion et la dilatation le long d'une ligne discrète à des angles arbitraires.

De plus, selon la position de l'élément structurant le long de la ligne, les résultats obtenus peuvent varier légèrement d'une position à l'autre. La figure 5.2 montre l'effet du choix de l'élément structurant, ici un DSS, sur le résultat après une opération de morphologie mathématique (l'ouverture). Nous pouvons voir que pour un même segment de droite le choix du paramètre  $\mu$  (voir la Définition 5.2.0.1) change radicalement le résultat. Les auteurs ont proposé une solution face à ce problème : utiliser les opérateurs *rank-max* et *rank-min*. Ils ont aussi proposé une méthode permettant de calculer efficacement les différentes opérations morphologiques des images en niveaux de gris. Cette optimisation se fonde sur la périodicité de la structure des droites discrètes. Toujours dans l'article, une application pour calculer l'orientation est présentée. Le calcul de l'orientation est fait en considérant qu'il existe, dans les images en niveaux de gris, deux types de structures : les structures claires et les structures sombres. Selon le type de structures, l'opération d'ouverture ou de fermeture morphologique est employée. L'élément structurant utilisé est un DSS. Ce segment est de longueur fixe, c'est-à-dire que la longueur est un paramètre d'entrée de la méthode proposée. L'outil développé permet, en plus du calcul de l'orientation locale, d'effectuer un lissage directionnel et de calculer de la granulométrie [Matheron, 1967, Serra, 1982]. Cependant, les résultats obtenus dépendent fortement de ce paramètre de longueur.

En utilisant l'opérateur de chemin issu de la morphologie mathématique, Merveille et ses collègues [Merveille et al., 2017, Merveille et al., 2018] ont introduit un opérateur non linéaire, appelé *Ranking the Orientation Responses of Path Operators (RORPO)*, pour l'analyse de structures curvilignes en 2D et 3D. RORPO est bien adapté pour extraire différentes caractéristiques (orientation, etc.) des objets qui ont une anisotropie intrinsèque. Il a été appliqué comme un filtre curviligne dans le cadre de segmentation des vaisseaux sanguins en imagerie médicale. L'outil a deux grandes applications : l'estimation de l'orientation et le filtrage. Bien que l'outil offre des résultats convaincants, nous



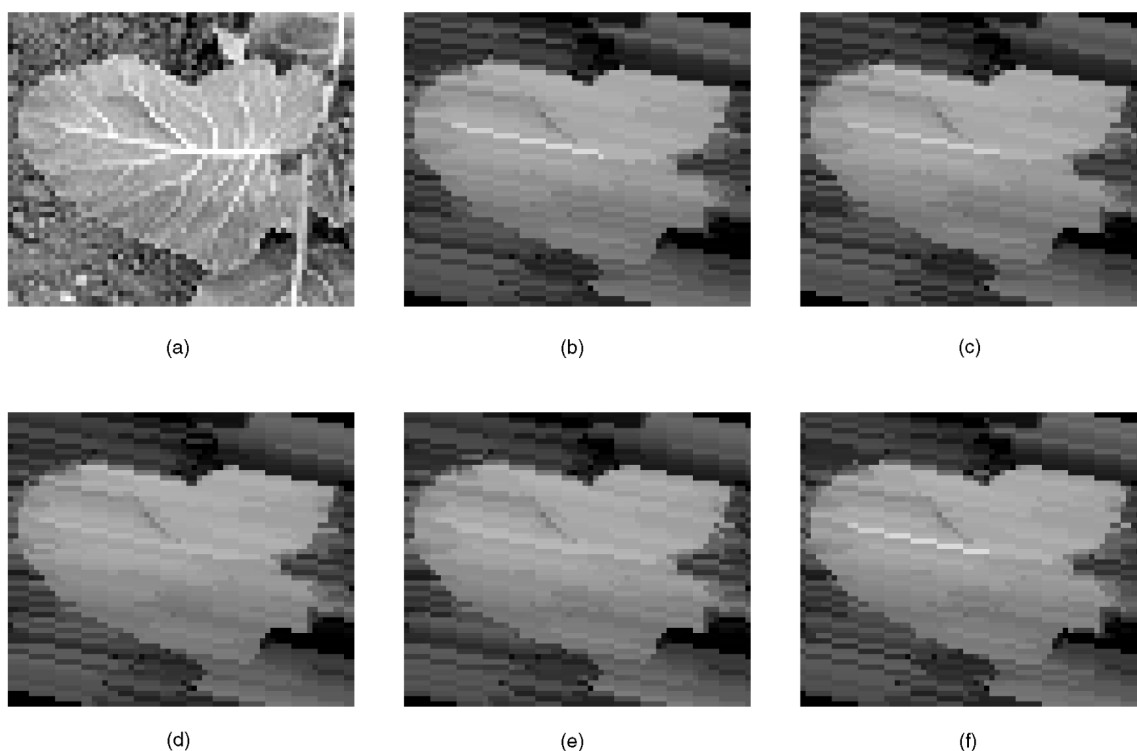


FIGURE 5.2 – Images issues de l'article [Soille et al., 1996]. Ouverture morphologique par des segments de droite discrète de 15 pixels. (a) Image d'origine (de taille  $66 \times 56$  pixels). Résultat de l'ouverture par l'élément structurant : (b)  $DSS(5, -1, 0)$ , (c)  $DSS(5, -1, -1)$ , (d)  $DSS(5, -1, -2)$ , (e)  $DSS(5, -1, -3)$  et (f)  $DSS(5, -1, -4)$ .

nous plaçons d'une part dans un cadre où l'outil principal est le segment de droite discrète et non d'objet curviligne. D'autre part, les temps de calcul présentés ne sont pas compatibles avec le cadre de temps réel qui l'un des objectifs du projet (application aussi bien en scierie qu'en forêt). C'est pourquoi, nous ne ferons pas de comparaisons avec cet outil dans la suite du chapitre.

Nous verrons dans un premier temps, dans la [section 5.2](#) les définitions de bases nécessaires pour la suite. Puis, nous présenterons, dans la [section 5.3](#), un nouveau filtre directionnel discret afin d'extraire des informations locales d'un pixel dans une image binaire. Enfin, la [section 5.4](#) nous permettra de présenter des pistes d'adaptation pour étendre l'outil dans le cas d'images en niveaux de gris. Cette piste d'adaptation s'inspire des travaux de [Soille and Talbot, 2001] mais dans laquelle nous allons étendre leur méthode afin de prendre en considération des variations de longueur des DSS mais aussi des variations d'épaisseur.

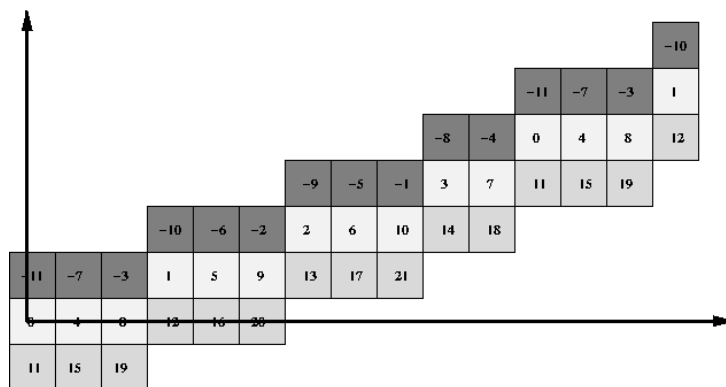


FIGURE 5.3 – Segment naïf de taille 15 (pixels) pour  $x \in [0, 14]$  et les segments de droites  $DSS(4, 11, 0)$  (blanc),  $DSS(4, 11, 11)$  (gris clair),  $DSS(4, 11, -11)$  (gris foncé). Le segment épais  $TDSS_{1,1}(4, 11)$  contient tous les pixels pour  $x \in [0, 14]$ . La valeur  $4x - 11y$  est indiquée pour chaque pixel  $(x, y)$ .

## 5.2 Notions de base

Nous nous intéressons, dans ce chapitre, au calcul de caractéristique géométriques à l'échelle locale d'un pixel d'une image, et pas uniquement du contour des objets. Nous rappelons ci-dessous des notions de bases de la géométrie discrète.

### 5.2.1 Droites et segments discrets

La première notion qui sera utile dans la suite est la notion de droite discrète et de segment discret.

**Définition 5.2.0.1 (Droite discrète [Reveillès, 1991])** Une droite discrète, ou DSL pour Digital Straight Line, dénotée par  $\mathcal{D}(a, b, \mu, \omega)$ , avec  $a, b, \mu$  et  $\omega$  des nombres entiers et  $\text{pgcd}(a, b) = 1$ , est l'ensemble de points  $(x, y) \in \mathbb{Z}^2$  vérifiant les inégalités :

$$\mu \leq ax - by < \mu + \omega$$

L'entier  $\mu$  est appelée le décalage et l'entier  $\omega$  est appelé épaisseur de la droite. Si  $\omega = \max(|a|, |b|)$ , la droite est 8-connexes est appelée naïve et notée  $\mathcal{D}(a, b, \mu)$ . Si  $\omega > |a| + |b|$ , alors la droite est dite épaisse. Un segment de droite naïve (DSS) est un ensemble fini connexe d'une droite naïve. Nous dénoterons le segment de droite naïve de caractéristiques  $a, b$  et  $\mu$  par  $DSS(a, b, \mu)$ .

Un segment naïf est dit *symétrique* pour un point  $(x, y)$  si et seulement si  $(x, y)$  appartient au segment et le segment contient autant de point des deux côtés de  $(x, y)$ . Nous pouvons souligner que pour des valeurs de  $a, b$  et  $\mu$  fixées, l'ensemble de droites naïves :

$$\{i \in \mathbb{Z}, \quad \mathcal{D}(a, b, \mu + i \max(|a|, |b|))\}$$

pave le plan  $\mathbb{Z}^2$  (voir la figure 5.3).

Dans la suite du manuscrit, nous nous intéressons aux segments naïfs dans un voisinage d'un point  $(x, y)$  donné. Nous étendons les segments naïfs en empilant les segments naïfs

les plus proches (d'après la notion de pavage du plan). Ainsi, nous définissons la notion de segment naïf épais.

**Définition 5.2.0.2 (Segment naïf épais)** Soit  $[0, l] \times [0, L] \subset \mathbb{Z}^2$ , un segment naïf épais d'indices  $k, j$  et de caractéristiques  $(a, b)$ , dénoté  $TDS_{k,j}(a, b)$ , est l'ensemble des points  $(x, y) \in [0, l] \times [0, L]$  appartenant à la droite :

$$\mathcal{D}(a, b, -k \max(|a|, |b|), (1 + k + j) \max(|a|, |b|))$$

avec  $k$  et  $j$  dans  $\mathbb{N}$ .

Nous pouvons voir  $k$  et  $j$  comme le nombre d'empilements (dans un sens et dans l'autre) de segments naïfs de caractéristiques  $(a, b)$ . La droite  $\mathcal{D}(a, b, 0)$  est appelée la *graine* du segment épais.

## 5.2.2 Arbre de Stern-Brocot et suite de Farey [G. Hardy, 1989, Stern, 1858]

La construction de l'ensemble des fractions irréductibles  $\frac{a}{b}$  comprises entre 0 et 1 peut se faire de manière itérative. En partant de deux fractions successives  $\frac{a}{b}$  et  $\frac{a'}{b'}$ , nous pouvons ajouter la nouvelle fraction  $\frac{a+a'}{b+b'}$ . L'arbre de *Stern-Brocot* est créé en commençant par les fractions  $0 = \frac{0}{1}$  et  $1 = \frac{1}{1}$ .

**Définition 5.2.0.3 (Suite de Farey)** La suite de Farey d'ordre  $n \in \mathbb{N}^*$ , dénotée par  $\mathcal{F}_n$ , est l'ensemble des fractions irréductibles comprises entre 0 et 1 et telle que le dénominateur est inférieur ou égal à  $n$ .

Les suites suivantes sont des suites de Farey :

$$\mathcal{F}_2 = \left\{ \frac{0}{1}, \frac{1}{2}, \frac{1}{1} \right\}$$

$$\mathcal{F}_3 = \left\{ \frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1} \right\}$$

Le nombre de segments naïfs dans le premier octant dans une grille carré de taille  $N \times N$  pixels est exactement le nombre d'éléments dans la suite de Farey d'ordre  $N$  [Berenstein and Lavine, 1988, Koplowitz et al., 1990]. La suite de Farey d'ordre  $N$  est un sous-arbre de l'arbre de Stern-Brocot (voir la figure 5.5).

La figure 5.4 montre les angles obtenus en utilisant la suite de Farey. Nous observons que la répartition des angles n'est pas uniforme, peu d'angles sont représentés autour des angles séparant deux octants successifs.

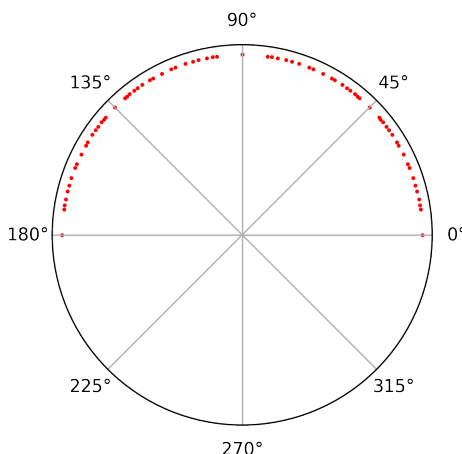


FIGURE 5.4 – Répartition des directions obtenues par les suite de Farey ( $\arctan \frac{a}{b}$ ).

### 5.3 Filtre directionnel : le cas binaire

Nous avons vu que le filtre proposé par [Soille and Talbot, 2001] se limite à l'orientation : la longueur est fixée. Inspirés par leur travail, nous proposons de surmonter les limitations de leur méthode. Ce travail a fait l'objet d'un article publié [Decelle et al., 2021a] (voir section 6.2).

Nous introduisons un nouveau filtre directionnel utilisant les DSS afin d'étendre le filtre avec des variations de longueurs, mais également des variations d'épaisseurs. En plus d'offrir un filtrage directionnel, les DSS sont utilisés pour estimer des caractéristiques de géométrie de l'image. Parmi ces caractéristiques, nous pouvons citer la longueur, l'épaisseur ou bien encore l'orientation. Plus précisément, nous récupérons les segments les plus longs qui passent par ce pixel. L'épaisseur des DSS récupérés est également prise en compte dans notre proposition d'extension. La longueur et l'épaisseur des segments sont utilisées pour extraire les caractéristiques géométriques locales des pixels, car elles contiennent des informations pertinentes pour la description du pixel.

#### 5.3.1 Calcul des segments naïfs symétriques

Soit l'ensemble  $\mathcal{S}$  décrivant un objet 8-connexe. Nous pouvons, sans perte de généralité, considérer le premier octant. Dans ce cas, l'arbre de Stern-Brocot va de  $0/1$  jusqu'à  $1/1$ . La méthode consiste à étendre d'abord à droite puis à gauche les segments jusqu'aux bords de  $\mathcal{S}$ .

Soit  $i \in \mathbb{N}$ , nous notons  $\mathcal{L}_i$  l'ensemble des segments naïfs de longueur  $i + 1$  (nombre de pixels) passant par  $p$  et inclus dans  $\mathcal{S}$ . Nous initialisons l'ensemble  $\mathcal{L}_{i+1}$  à l'ensemble vide. Pour chaque segment dans  $\mathcal{L}_i$ , nous ajoutons les points suivants  $q = (x_q, y_q)$  du segment. Nous ajoutons à droite puis à gauche du segment si possible, c'est-à-dire que nous ajoutons (si possible) le point en  $x_q = x_p + (i + 1)$  et le point en  $x_q = x_p - (i + 1)$ . Si ce segment étendu est dans  $\mathcal{S}$  nous l'ajoutons à  $\mathcal{L}_{i+1}$ , sinon il ne l'est pas. De plus, pour chacun des points ajoutés, s'il s'agit d'un point d'appui, il est possible de déplacer soit en  $(x_q, y_q - 1)$  (point d'appui supérieur) soit en  $(x_q, y_q + 1)$  (point d'appui inférieur). Chacune

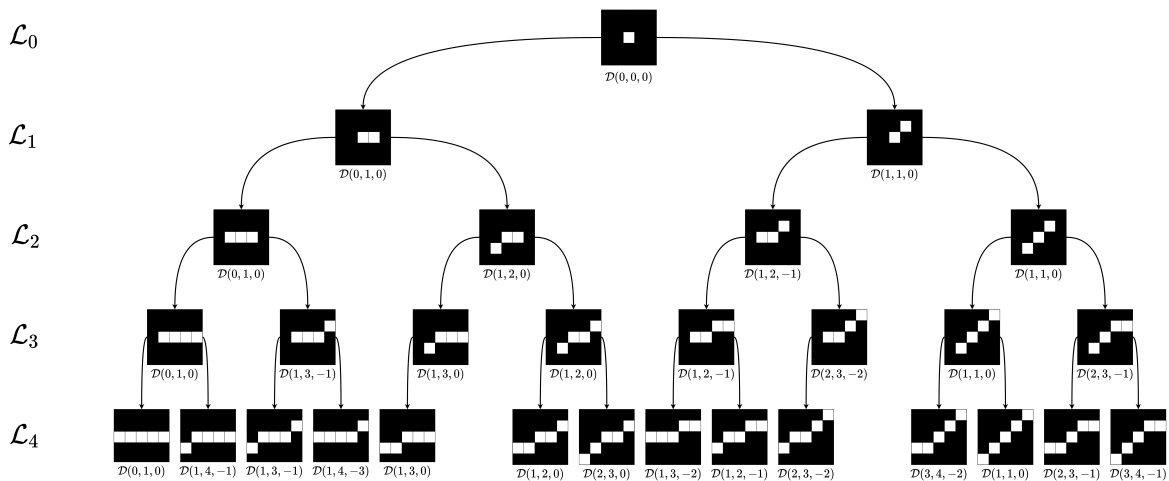


FIGURE 5.5 – Exemple des segments naïfs symétriques construits pendant l’algorithme.

des combinaisons possibles est dans  $\mathcal{S}$  alors nous l’ajoutons dans l’ensemble des segments potentiels. Nous continuons ce processus tant qu’il est possible d’étendre au moins un segment.

La méthode est résumée dans [algorithme 3](#). La [figure 5.5](#) montre les segments de droites naïfs construits durant l’algorithme. Nous pouvons remarquer qu’à chaque itération, pour un  $n$  donné, l’ensemble des caractéristiques  $(a, b)$  issues des segments naïfs de  $\mathcal{L}_n$  est égale à la suite de Farey d’ordre  $n$ .

La [figure 5.6](#) illustre certaines étapes de la méthode pour les segments symétriques (pour un pixel donné en orange). Sur le côté gauche des quatre figures, la couleur rouge montre la zone couverte par les DSS à différentes étapes. À chaque étape, les DSS extensibles sont conservés. Sur le côté droit, nous montrons les fractions et l’orientation associée des DSS. En vert, les DSS extensibles avec les fractions correspondantes. En rouge, Les DSS non-extensibles. En bleu, les DSS non-extensibles mais nécessaires pour construire de nouveaux DSS. Nous commençons par le pixel  $i = 0$ . Puis, nous passons au niveau  $i = 2$ . À ce stade, certains DSS ne sont pas extensibles, nous les supprimons et nous passons au niveau  $i = 4$ , et ainsi de suite. L’algorithme s’arrête au niveau  $i = 12$  (c’est-à-dire aux DSS de longueur 13 pixels) où seul  $DSS(5, 4, -4)$  est encore extensible. Une animation de l’algorithme est disponible <sup>14</sup>

### 5.3.2 Complexité en temps

Les opérations de base, en  $\mathcal{O}(1)$  sont : la comparaison (le point suivant appartient-il à l’ensemble ?), la mise à jour des paramètres  $(a, b, \mu)$  du segment, vérifier si le point d’appui est inférieur ou supérieur. Considérons le pire cas de notre méthode : un carré blanc de taille  $N \times N$  avec  $N \in \mathbb{N}$ . Regardons le cas du pixel au centre du carré (pour  $N$  impair). Notons que tous les segments naïfs passant par le centre et de taille  $N$  sont des segments maximaux. Le nombre de segments passant par l’origine [[Berenstein and Lavine, 1988](#)]

14. Vers la vidéo : <https://www.youtube.com/embed/tnDHB10QjU?start=316>. Ou bien vers la présentation : [https://gdmm2020.sciencesconf.org/data/MM1\\_DECILLE.pdf](https://gdmm2020.sciencesconf.org/data/MM1_DECILLE.pdf)

---

**Algorithme 3** : Calcul des segments naïfs symétriques les plus longs (cas du premier octant)

---

**Entrées** : Un ensemble  $\mathcal{S}$  connexe et un pixel  $p(x, y) \in \mathcal{S}$   
**Sorties** : L'ensemble  $\mathcal{L}_i$  des segments maximaux de longueurs  $i$  passant par  $p$  et inclus dans  $\mathcal{S}$

```

1  $\mathcal{L}_0 \leftarrow \{DSS(0, 1, 0)\}$ 
2  $i \leftarrow 0$ 
3 tant que  $\mathcal{L}_i \neq \emptyset$  faire
4    $\mathcal{L}_{i+1} \leftarrow \emptyset$ 
5   pour chaque  $DSS(a, b, \mu) \in \mathcal{L}_i$  faire
6     si  $DSS(a, b, \mu)$  peut s'étendre à droite alors
7        $DSS(a', b', \mu') \leftarrow \text{incrementer}(DSS(a, b, \mu), i + 1)$ 
8       si  $DSS(a', b', \mu') \in \mathcal{S}$  alors
9          $\mathcal{L}_{i+1} \leftarrow \mathcal{L}_{i+1} \cup DSS(a', b', \mu')$ 
10   $i \leftarrow i + 1$ 
11   $\mathcal{L}_{i+1} \leftarrow \emptyset$ 
12  pour chaque  $DSS(a, b, \mu) \in \mathcal{L}_i$  faire
13    si  $DSS(a, b, \mu)$  peut s'étendre à gauche alors
14       $DSS(a', b', \mu') \leftarrow \text{incrementer}(DSS(a, b, \mu), i + 1)$ 
15      si  $DSS(a', b', \mu') \in \mathcal{S}$  alors
16         $\mathcal{L}_{i+1} \leftarrow \mathcal{L}_{i+1} \cup DSS(a', b', \mu')$ 
17   $i \leftarrow i + 1$ 
18 retourner  $\mathcal{L}_{i-2}$ 

```

---

est de l'ordre de :

$$\frac{N^3}{\pi^2} + \mathcal{O}(N^2 \log(N))$$

Autrement dit, le nombre de segments est  $\mathcal{O}(N^3)$ . Il faut  $N$  calculs pour trouver ces segments : faire le calcul pour les segments de taille 1, puis ceux de taille 2 et ainsi de suite qu'aux segments de taille  $N$ . Ce qui nous amène à avoir  $\mathcal{O}(N^4)$  opérations. Or, nous devons effectuer ces opérations pour  $N^2$  pixels, cela revient, dans le pire cas, à une complexité de l'ordre de

$$\mathcal{O}(N^6)$$

par rapport au nombre de pixels  $M = N^2$ , la complexité est de :

$$\mathcal{O}(M^3)$$

### 5.3.3 Application à l'estimation de caractéristiques

Nous avons, malgré une complexité en temps non-négligeable, appliqué notre outil sur des images simples. Notre outil permet d'obtenir un ensemble d'informations à l'échelle local d'un pixel. Nous avons pour chaque pixel de l'image calculé :

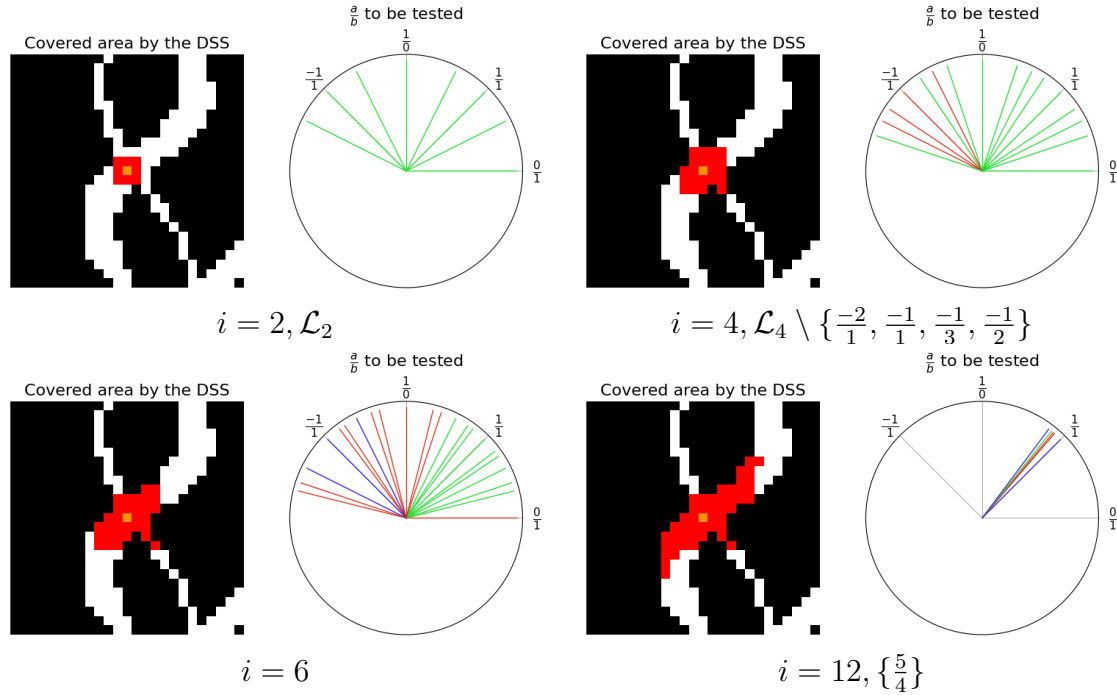


FIGURE 5.6 – Exemple, pas à pas, du cas particulier de segment naïf symétrique. En vert, les fractions  $(a, b)$  des segments pouvant être étendus. En rouge, celles ne pouvant plus être étendus.

- la longueur  $L$  (des segments naïfs symétriques les plus longs) ;
- l'orientation  $\theta$  moyenne (des segments naïfs symétriques les plus longs) ;
- l'épaisseur directionnel  $ED$  (des segments naïfs symétriques les plus longs) que nous allons détailler juste après.

La longueur  $L$  au pixel  $p$  est la longueur du plus long segment naïf symétrique. Pour calculer cette longueur, nous utilisons un estimateur simple :  $N_e + \sqrt{2}N_o$ , avec  $N_e$  et  $N_o$  le nombre de chiffres pairs et impairs dans le code Freeman associé au DSS. Cette caractéristique indique si localement l'objet est allongé. Une valeur élevée indique une forme droite et longue.

L'orientation  $\theta$  au pixel  $p$  est la moyenne des orientations des DSS, qui est calculée de la manière suivante [Jammalamadaka and Sengupta, 2001]. Posons :

$$\bar{S} = \sum_{DSS(a,b,\mu) \in \mathcal{L}_i} \sin(2 * \text{atan2}(\frac{a}{b})) = \sum_{DSS(a,b,\mu) \in \mathcal{L}_i} \frac{2ab}{a^2 + b^2} \quad (5.1)$$

$$\bar{C} = \sum_{DSS(a,b,\mu) \in \mathcal{L}_i} \cos(2 * \text{atan2}(\frac{a}{b})) = \sum_{DSS(a,b,\mu) \in \mathcal{L}_i} \frac{b^2 - a^2}{a^2 + b^2} \quad (5.2)$$

alors l'orientation est donnée par :

$$\theta(I)(x, y) = \pi + \frac{1}{2} \text{atan2}(\frac{\bar{S}}{\bar{C}}) \quad (5.3)$$

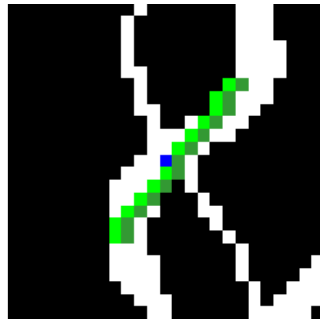


FIGURE 5.7 – En vert clair le segment symétrique  $DSS(5, 4, -4)$  le plus long passant par le pixel en bleu. Le segment en vert foncé est le segment  $TDSS_{5,4}(1, 0)$

Pour obtenir la caractéristique d'épaisseur directionnelle, nous épaississons chaque DSS final obtenu selon notre définition de  $TDSS_{k,j}(a, b)$  (voir la [section 5.2](#)). L'épaisseur directionnelle  $ED$  est alors définie comme la somme du maximum de  $j$  et  $k$  pour lesquels le  $TDSS$  est inclus dans  $I$  ; c'est-à-dire de combien de fois nous pouvons *paver* au maximum le segment dans les deux sens. Nous avons fait le choix d'effectuer l'élongation du segment puis de faire son épaississement. En faisant ce choix, nous pouvons plus facilement estimer la longueur et l'orientation. De plus, nous pensons que ce choix est moins onéreux en mémoire, dans le sens où l'on ne manipule pas la longueur et l'épaisseur en même temps ce qui augmenterait le nombre de segments à garder en mémoire. Néanmoins, cela implique que l'épaisseur obtenue est contrainte par la largeur. En effet, plus le segment sera long, plus il sera difficile de l'épaissir. D'autres choix pourront faire l'objet de recherches plus approfondies. Par exemple, nous pourrions décider d'alterner l'élongation du segment et son épaississement.

La [figure 5.7](#) montre le segment symétrique  $DSS(5, 4, -4)$  le plus long obtenu suite à l'algorithme (en vert clair). Ce segment possède 13 pixels. La longueur euclidienne  $L$  est de 15,7 pixels, l'orientation est  $\theta = \tan^{-1}(\frac{5}{4}) \approx 51^\circ$ , enfin l'épaisseur directionnelle est  $ED = j + k = 0 + 1 = 1$ . C'est-à-dire que nous pouvons le paver au plus une fois. Ce pavage possible est montré par le segment en vert foncé.

Le [tableau 5.1](#) montre les images des caractéristiques de longueur, d'orientation et d'épaisseur directionnelle obtenues après application de l'algorithme sur l'ensemble des pixels d'une image en entrée.

Nous pouvons voir que la notion de longueur est différente que l'objet soit localement convexe ou non. Dans le cas d'une convexité locale, alors la longueur sera plus petite. En effet, le bord de l'objet va limiter la possibilité de l'étendre. Dans le cas d'une non-convexité locale, le segment de droite a plus de liberté pour s'étendre, ce qui implique des longueurs plus grandes. Autrement dit, le calcul de longueur renseigne aussi sur la convexité locale de l'objet.

Dans le cas du calcul de l'orientation, nous pouvons voir que l'orientation fait un dégradé radial du noir (angle de  $0^\circ$ ) vers le blanc (angle de  $180^\circ$ ). Cela correspond à l'orientation d'un cercle.

Nous pouvons voir une corrélation entre la longueur et l'épaisseur directionnelle. Un segment court est potentiellement un segment que nous pouvons plus facilement épaissir.








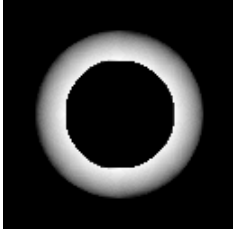




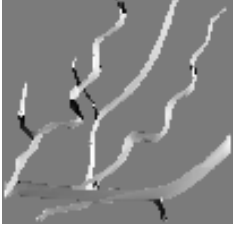

Image	Longueur	Orientation	ED
			
			
			

TABLE 5.1 – Extraction des caractéristiques géométriques à partir des segments symétriques

Ceci n'est pas incohérent : un segment court est moins contraint par l'espace autour de lui et ainsi l'épaississement (selon notre définition) se fait plus aisément. Autre point, les segments  $DSS(1, 1, *)$  ou  $DSS(-1, 1, *)$  sont plus épais. Cela s'explique simplement par le fait que ces segments appartiennent à deux octants, il est possible de les paver non pas dans deux directions mais quatre directions. Cela implique d'augmenter potentiellement la valeur de l'épaisseur directionnelle.

## 5.4 Filtre directionnel : image en niveaux de gris

Dans cette section, nous nous focalisons sur une approche afin d'étendre l'outil qui vient d'être présenté pour les images en niveaux de gris. La caractéristique sur laquelle nous nous focalisons est l'orientation. Nous nous intéressons à l'orientation locale des structures de l'image. Pour cela, nous avons décidé d'effectuer l'analyse en utilisant des outils de la morphologie mathématique. Nous allons nous inspirer de la méthode de [Soille and Talbot, 2001], mais nous allons l'étendre, comme pour le cas binaire, en faisant varier la longueur des segments et leur épaisseur.

### 5.4.1 Première approche

Dans les images en niveaux de gris, nous pouvons différencier les structures claires formées de pixels de forte intensité et les structures sombres formées de pixels de faible intensité. Afin de pouvoir différencier ces deux types de structures, nous proposons d'utiliser deux outils de la morphologie mathématique : l'érosion et la dilatation. Ainsi, les deux outils seront utilisés de la manière suivante :

- l'érosion morphologique  $\varepsilon$  pour analyser les structures claires ;
- la dilatation morphologique  $\delta$  pour analyser les structures sombres.

Soit  $k \in \mathbb{N}$ , nous posons  $\mathcal{C}(a, b, c)$  le segment naïf symétrique de paramètres  $a, b, \mu = 0$  et  $\omega = \max(|a|, |b|)$  et de longueur  $c$ . Pour un couple  $(i, j) \in \mathbb{Z}^2$ , nous posons également  $g = \text{pgcd}(|i|, |j|)$ , ce qui nous permet de définir la taille du segment (i.e. le nombre de pixels) qui est égale à :

$$c(i, j) = \begin{cases} \max(\lfloor \frac{i}{g} \rfloor, \lfloor \frac{j}{g} \rfloor) + 1 + 2(g - 1) & \text{si } \omega \text{ est pair} \\ \max(\lfloor \frac{i}{g} \rfloor, \lfloor \frac{j}{g} \rfloor) + 2 + 2(g - 1) & \text{sinon} \end{cases}$$

Cela revient à calculer la longueur selon la profondeur dans l'arbre de Stern-Brocot (voir la [figure 5.5](#)). Nous commençons par le premier niveau de l'arbre de Stern-Brocot, c'est-à-dire  $\mathcal{L}_2$ , et des segments de 3 pixels. Les segments suivants seront composés de 5 pixels ( $\mathcal{L}_4$ ), et ainsi de suite. Chaque segment définit un élément structurant qui permet de calculer l'érosion de l'image par celui-ci. La taille maximale des segments est fixée à  $k$ .

Nous introduisons deux images  $\mathcal{E}$  (pour l'érosion) et  $\mathcal{D}$  (pour la dilatation) de taille  $k \times k$  obtenues pour un pixel  $(x, y)$  de l'image d'entrée. Ainsi, pour le pixel  $(x, y)$  de l'image, chaque point  $(a, b)$  d'une de ces deux images est le résultat de l'opérateur morphologique associé à la transformée appliquée à l'image  $I$  (dont nous prenons la valeur au pixel  $(x, y)$ ). Le pixel central  $(\frac{k}{2}, \frac{k}{2})$  est la valeur en  $I(x, y)$ . La [figure 5.8](#) montre les éléments structurant pour les coordonnées  $(a, b)$  de l'image. Le pixel central est celui de coordonnées  $(0, 0)$ . D'un point de vue mathématique, nous pouvons écrire les images de la manière suivante :

$$\mathcal{E}_{a,b}(x, y) = \varepsilon_{\mathcal{C}(b,a,c(a,b))}(I)(x, y) \quad (5.4)$$

$$\mathcal{D}_{a,b}(x, y) = \delta_{\mathcal{C}(b,a,c(a,b))}(I)(x, y) \quad (5.5)$$

La [figure 5.9](#) montre des exemples concrets des images obtenues. Le pixel sur lequel sont appliqués ces images est le pixel rouge. Nous pouvons voir que selon si le pixel est clair alors l'image  $\mathcal{E}$  donne le plus d'information. Et à l'inverse, l'image  $\mathcal{D}$  donne plus d'information si le pixel est sombre. Nous pouvons également remarquer que les directions qui appartiennent à deux octants (par exemple  $(1, 1)$  ou  $(0, 1)$ ) sont très marquées.

Une analyse approfondie des transformées pourrait permettre d'estimer des caractéristiques locales comme la longueur, la largeur ou bien l'orientation.

### 5.4.2 Inclusion de l'épaisseur

Nous venons de voir que ces transformées ont quelques inconvénients. Par exemple, nous pouvons voir que certaines directions, comme par exemple  $0^\circ$  et  $45^\circ$  (de manière générale les segments appartenant à deux octants), perturbent l'analyse des transformées.

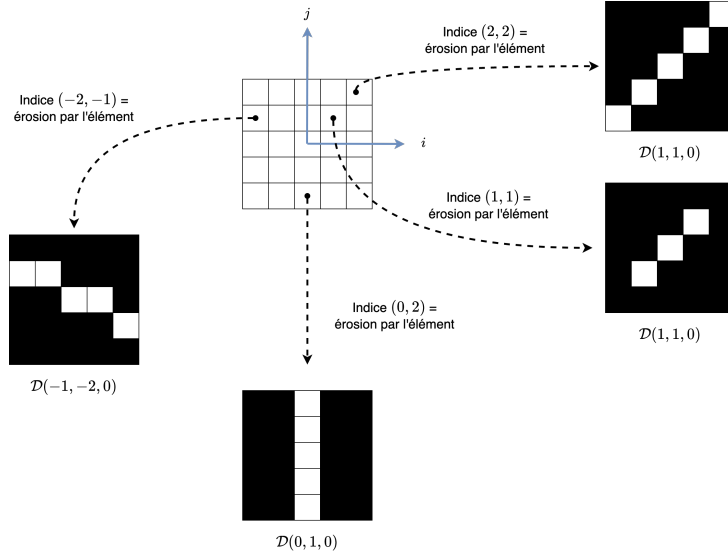


FIGURE 5.8 – Éléments structurants pour obtenir la transformée  $\mathcal{E}$  selon les coordonnées  $(i, j)$  (de la transformée).

Dans cette section, nous proposons d’inclure des segments naïfs épais afin de potentiellement estimer une épaisseur, mais également de limiter les perturbations des segments qui appartient à deux octants.

Nous épaississons le segment si possible, et calculons à nouveau l’érosion par celui-ci. Contrairement aux transformées précédentes, nous allons additionner les résultats de chaque érosion si les caractéristiques du segment sont les mêmes. C’est-à-dire que le résultat du point  $(0, 2)$  sera additionné au résultat du point  $(0, 1)$  (malgré la longueur plus longue de l’élément structurant en  $(0, 2)$ ). L’[algorithme 4](#) résume la méthode.

Pour chaque fraction  $a/b$ , nous notons  $B_{i,j,k}$  le  $TDSS_{k,j}(b, a)$  pour lequel le DSS de départ contient  $2i + 1$  pixels, nous calculons la transformée  $Y_{a,b}(x, y)$  égale à :

$$Y_{a,b}(x, y) = \frac{1}{n - \alpha + 1} \sum_{i=0}^n \left[ \sum_{k=0}^{i-1} \varepsilon_{B_{i,k,0}}(I)(x, y) + \sum_{j=1}^{i-1} \varepsilon_{B_{i,0,j}}(I)(x, y) \right] \quad (5.6)$$

et  $\alpha = \max(\max(|a|, |b|) - 1, 1)$ . Nous introduisons un coefficient de normalisation  $\alpha$  afin de tenir compte du fait que le nombre de DSS dépend de  $a/b$ . Nous additionnons sur le même indice  $(a, b)$  des segments de différentes longueurs, ce qui peut biaiser la valeur en ce point. La normalisation est là pour réduire cet effet. De fait, si le segment  $DSS(a, b, 0)$  est défini à un instant donné, il le sera pour les itérations suivantes. Par exemple, la fraction  $1/2$  apparaîtra  $n$  fois pour un élément structurant (segment de longueur 3, 5, 7, ...,  $2n + 1$  pixels), la fraction  $1/3$  apparaîtra  $n - 1$  fois (segment de longueur 5, 7, ...,  $2n + 1$  pixels), et ainsi de suite.

La [figure 5.10](#) présente, pour un pixel de l’image d’entrée (image en bas à droite dans la figure), les différentes étapes du processus proposé si on ne considère que les variations de longueur. Le pixel considéré est celui situé au centre de l’image (croix verte).

Les dix premières images de la [figure 5.10](#) mettent en valeur les résultats obtenus pour

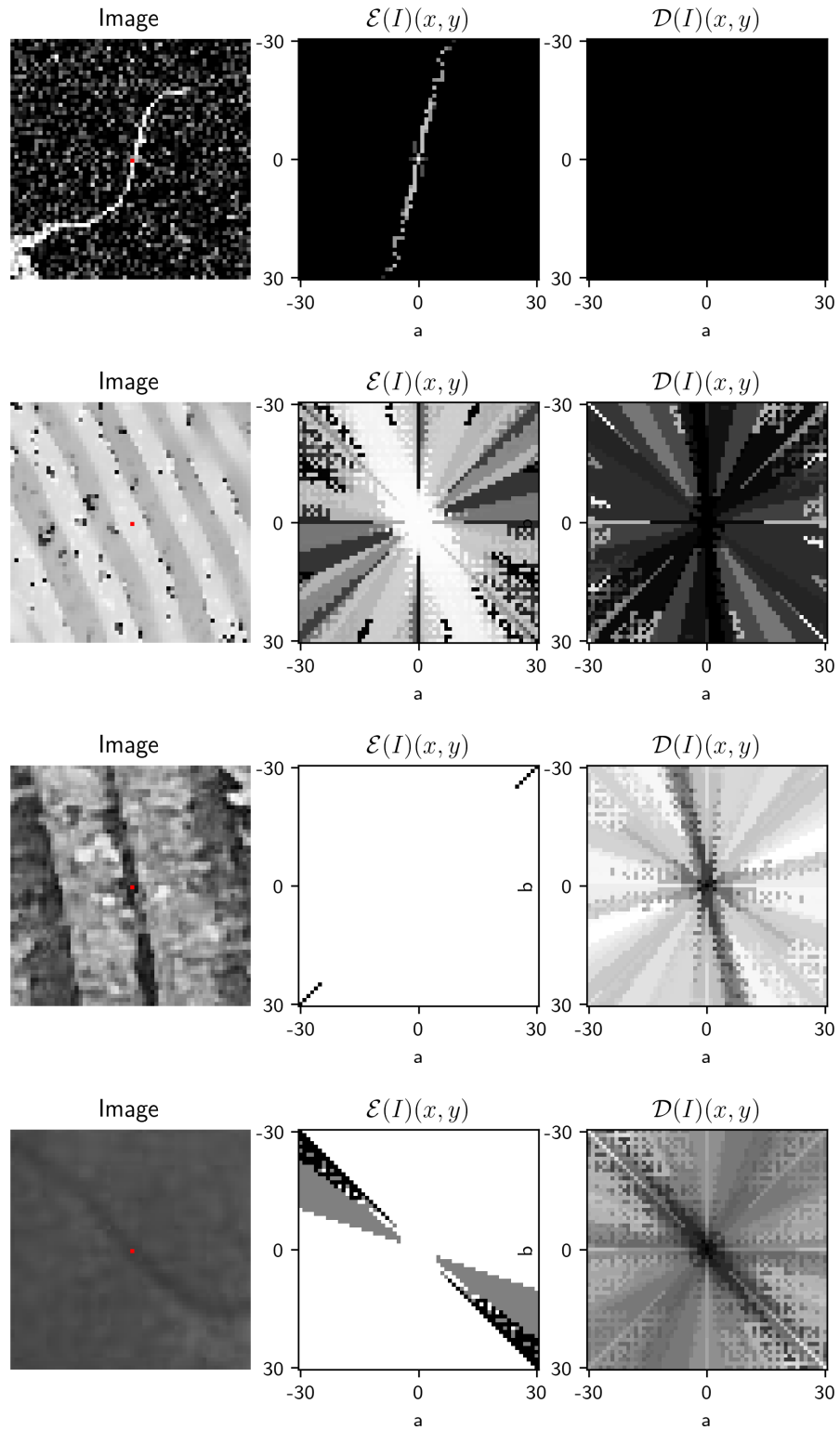


FIGURE 5.9 – Exemples des transformées en niveaux de gris. Un point rouge sur les images indique où la transformée a été appliquée.

---

**Algorithme 4 :** Méthode de filtrage en niveaux de gris (premier octant).

---

**Entrées :** Une image  $I$ ,  $n$  le nombre maximum de pixels dans un segment

**Sorties :** Une matrice d'images  $Y$

```

1  $Y \leftarrow$  matrice de taille  $(2n+1, 2n+1)$  // L'élément à l'indice  $(a, b)$  est
   l'image  $Y_{a,b}$ 
2  $A \leftarrow (0, 1, 1)^T$ 
3 pour  $i = 0$  à  $n$  faire // Itération sur la longueur jusque  $n$ 
4   pour chaque  $a/b \in \mathcal{F}_{i+2}$  faire // Itération sur la suite de Farey
   d'ordre  $i + 2$ 
5      $B \leftarrow TDSS_{0,0}(a, b)$  with  $2i+1$  points
6      $E \leftarrow \varepsilon_B(I)$  // Érosion de  $I$  par le DSS  $B_{i,0}$ 
7      $E^+ \leftarrow E$ 
8      $E^- \leftarrow E$ 
9      $Y_{a,b} \leftarrow Y_{a,b} + E$ 
10    pour  $k \leftarrow 1$  to  $i$  faire // Itération sur l'épaisseur
11       $E^+ \leftarrow \varepsilon_A(E^+)$  // Érosion par  $TDSS_{0,k}(a, b)$ 
12       $E^- \leftarrow \varepsilon_{\tilde{A}}(E^-)$  // Érosion par  $TDSS_{k,0}(a, b)$ 
13       $Y_{a,b} \leftarrow Y_{a,b} + E^+$ 
14       $Y_{a,b} \leftarrow Y_{a,b} + E^-$ 
15 Return  $Y$ 

```

---

chaque érosion, c'est-à-dire l'érosion de  $I$  par  $B_{i,0,0}$  avec  $i$  variant de 1 à 10. Les axes sont les paramètres  $a$  et  $b$  du segment naïf symétrique. Pour chaque couple  $(a, b)$ , nous mettons la valeur de gris résultant de l'érosion de  $I$  par  $B_{i,0,0}$ . Dans chaque image, le pixel ayant la valeur la plus élevée est encadré de la couleur rouge.

L'image précédant la dernière montre pour chaque couple  $(a, b)$  la somme de chaque érosion pour les différentes longueurs. On suppose que l'orientation est celle qui a la valeur de gris la plus élevée (pixel rouge).

Nous constatons que la fraction  $a/b$  pour laquelle le maximum est atteint change plusieurs fois en fonction de la valeur de  $i$ , c'est-à-dire du nombre de points du segment. En choisissant  $i = 9$  (soit 19 pixels), l'orientation semble correcte, mais en choisissant  $i = 10$  (soit 21 pixels), la direction n'est pas celle souhaitée (le segment correspondant est représenté en rouge sur la dernière image). Par contre, la direction résultant de la somme est correcte (le segment correspondant est représenté en vert dans la dernière image).

### 5.4.3 Simplification du calcul pour l'épaisseur

Il n'est pas nécessaire de construire spécifiquement les éléments TDSS pour prendre en compte la variation d'épaisseur. En effet, pour le premier octant, avec :

$$A = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

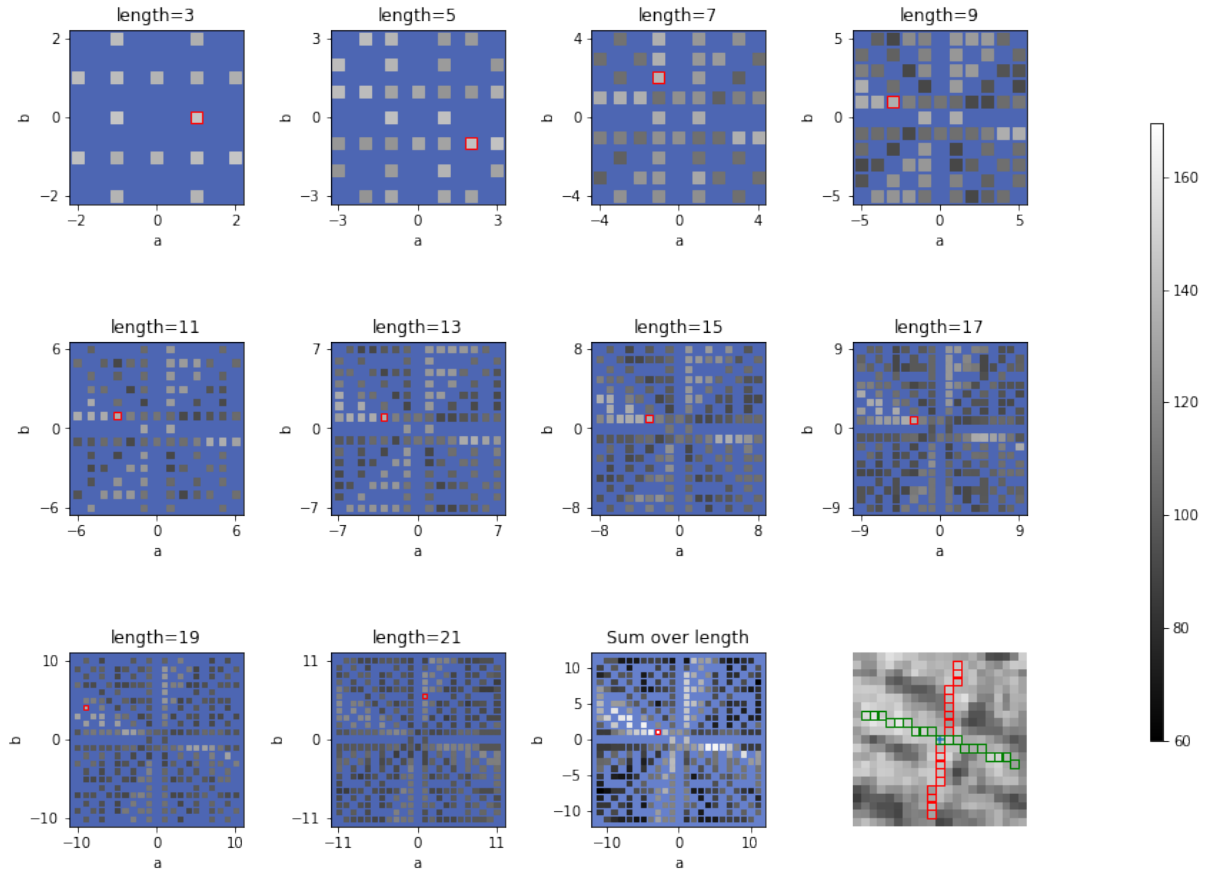


FIGURE 5.10 – Illustration de l’approche proposée avec des variations de longueurs.

nous avons la relation suivante :

$$B_{i,j,k+1} = B_{i,j,k} \oplus A$$

où  $\oplus$  est la dilatation morphologique. Un exemple de cette relation est mis en évidence dans la [figure 5.11](#) avec  $B_{6,0,0} \subset B_{6,0,1}$  et  $(a, b) = (3, 7)$ . De plus, nous avons la propriété :

$$\varepsilon_{A \oplus B}(I) = \varepsilon_A(\varepsilon_B(I))$$

qui dans notre cas d’élément structurant devient la relation :

$$\varepsilon_{B_{i,j,k+1}}(I) = \varepsilon_A(\varepsilon_{B_{i,j,k}}(I))$$

La relation reste valable pour  $B_{i,j,k} \subset B_{i,j+1,k}$  en prenant le symétrique de  $A$ , c’est-à-dire  $\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ . Cette relation est illustrée par la ligne 10 de l’[algorithme 4](#).  $A$  est à modifier en fonction de l’octant.

De plus, pour les éléments  $B_{i,0,0}$  nous pouvons utiliser l’algorithme développé par Soille et Talbot [[Soille and Talbot, 2001](#)] (voir ligne 6 de l’[algorithme 4](#)).

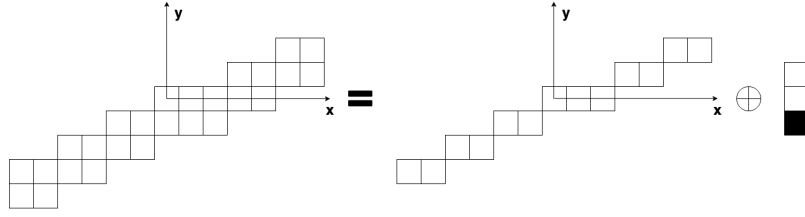


FIGURE 5.11 – Relation en dilatation morphologique et segment épais.

#### 5.4.4 Estimation de l'orientation

Dans cette section, nous faisons une proposition afin d'estimer l'orientation à partir des images intermédiaires obtenues par l'équation 5.6.

##### 5.4.4.1 Principe du calcul

Nous pouvons considérer trois variantes vis-à-vis de la méthode proposée :

- considérer toutes les longueurs et épaisseurs possibles (voir l'équation 5.6) ;
- considérer toutes les épaisseurs et la longueur est fixée à  $n$  (voir l'équation 5.7) ;
- considère toutes les longueurs et l'épaisseur est fixée à 0 (voir l'équation 5.8).

Nous avons vu la première variante avec l'équation 5.6. Cette équation se simplifie pour les deux autres variantes. Cela donne respectivement pour la deuxième et la troisième variante :

$$Y_{a,b} = \sum_{k=0}^{n-1} \varepsilon_{B_{n,k,0}}(I) + \sum_{j=1}^{n-1} \varepsilon_{B_{n,0,j}}(I) \quad (5.7)$$

$$Y_{a,b} = \frac{1}{n - \alpha + 1} \sum_{i=0}^n \varepsilon_{B_{i,0,0}}(I) \quad (5.8)$$

Dans les trois cas, nous nous retrouvons à estimer uniquement l'orientation dominante pour chaque pixel  $p$  puisque les images ne reposent que sur  $a/b$ . Inspiré par la méthode [Soille and Talbot, 2001], nous définissons à chaque point  $x$  l'orientation *claire* comme :

$$Dir^+(p) = \{ \tan^{-1}(\frac{a_i}{b_i}), | \forall (a_i, b_i) \neq (a_j, b_j) Y_{a_i, b_i}(p) \geq Y_{a_j, b_j}(p) \} \quad (5.9)$$

C'est-à-dire que l'orientation *claire* est celle de l'élément structurant pour lequel le résultat est le plus élevée pour l'érosion.

Nous introduisons une valeur supplémentaire pour nous permettre de déterminer les orientations à choisir.

$$G^+(p) = \max_{a,b} (Y_{a,b}(p)) - \min_{a,b} (Y_{a,b}(p)) \quad (5.10)$$

La valeur de  $G^+(p)$  peut être interprétée comme une intensité des structures claires au pixel  $p$ . L'orientation  $Dir^-$  des structures sombres et leur intensité  $G^-$  sont calculées

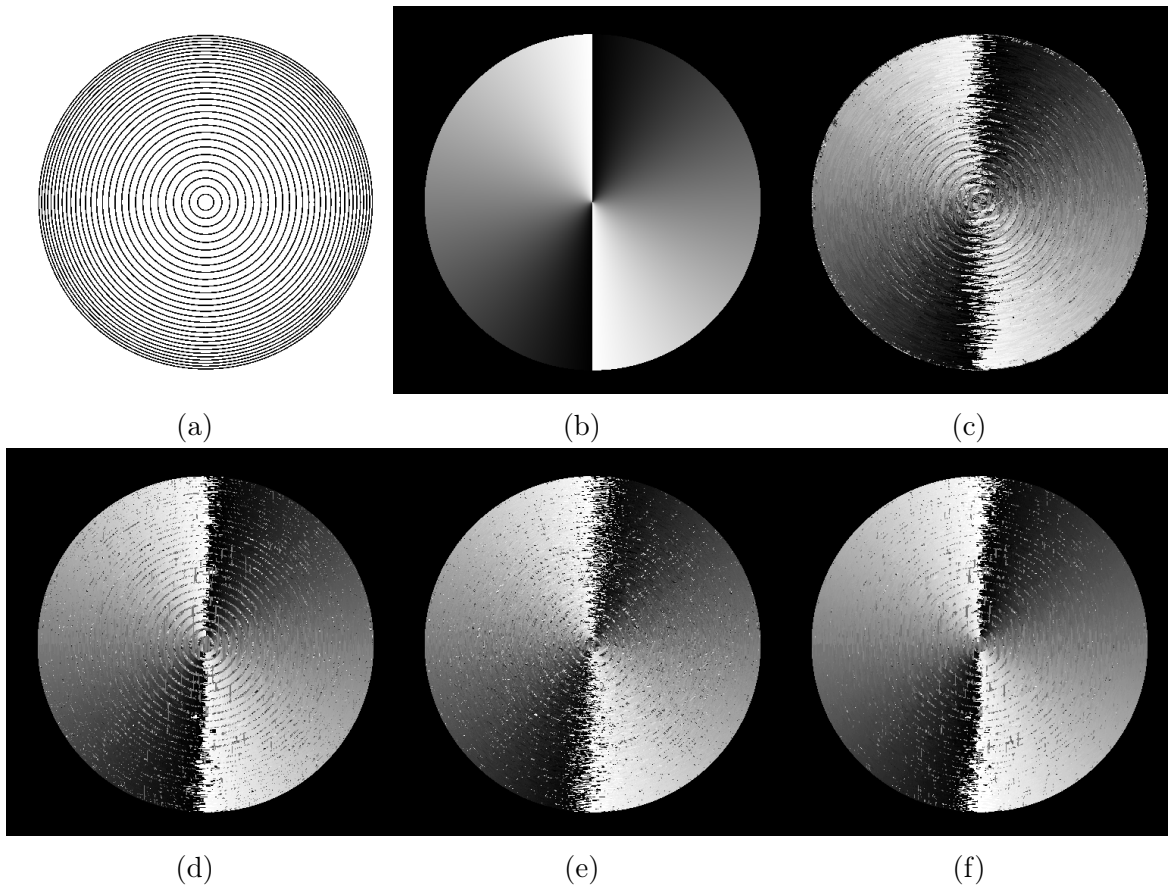


FIGURE 5.12 – Résultats de notre méthode et celle de [Soille and Talbot, 2001]. (a) Image de cercles concentriques (b) Vérité terrain. (c) [Soille and Talbot, 2001]. (d) Variations d'épaisseur. (e) Variations de longueur. (f) Les deux variations.

sur le même principe que  $G^+$  mais en utilisant  $J = 255 - I$  au lieu de  $I$ . Finalement, l'orientation au pixel  $p$  est obtenue par la formule :

$$\theta(p) = \begin{cases} Dir^+(p) & , \text{ if } G^+(x) \geq G^-(p) \\ Dir^-(p) & , \text{ sinon} \end{cases}$$

#### 5.4.4.2 Résultats préliminaires

Une analyse approfondie de la méthode pour les images en niveaux de gris n'a pas encore été faite. Néanmoins, nous proposons une première comparaison avec la méthode proposée par Soille et Talbot [Soille and Talbot, 2001]. Pour ce faire, nous considérons d'abord des objets circulaires (voir la figure 5.12) puisque la vérité terrain (orientation) peut être facilement déterminée. Cette information est utile pour démontrer l'efficacité de la méthode. Ainsi, nous avons créé des images de cercles concentriques auxquelles nous avons ajouté un peu de bruit. Ces cercles sont illustrés dans la figure 5.12. Pour comparer les méthodes avec la vérité terrain, nous proposons d'utiliser la formule proposée par



	[Soille and Talbot, 2001]	Épaisseur	Longueur	Épaisseur+Longueur
<b>RMSD</b>	0,2809	0,3111	0,2436	0,2317

TABLE 5.2 – RMSD calculé pour les différentes méthodes.

Turroni et al. [Turroni et al., 2011] :

$$RMSD = \sqrt{\frac{\sum_x d^2(\hat{\theta}_x, \theta_x)}{\text{Nombre de pixels}}}, \quad d(\theta_1, \theta_2) = \begin{cases} \theta_1 - \theta_2 & \text{if } -\frac{\pi}{2} \leq \theta_1 - \theta_2 < \frac{\pi}{2} \\ \pi + \theta_1 - \theta_2 & \text{if } \theta_1 - \theta_2 < -\frac{\pi}{2} \\ \pi - \theta_1 + \theta_2 & \text{if } \theta_1 - \theta_2 \geq \frac{\pi}{2} \end{cases}$$

avec  $\hat{\theta}_x$  l'orientation estimée en  $x$  et  $\theta_x$  l'orientation théorique estimée en  $x$ .

Le [tableau 5.2](#) montre les valeurs du RMSD pour les différentes méthodes : [Soille and Talbot, 2001], notre méthode avec uniquement la variation d'épaisseur (longueur fixée), notre méthode avec uniquement la variation de longueur (épaisseur fixée à 0) et notre méthode avec la variations de l'épaisseur et de la longueur. Pour chaque méthode, nous avons fixé la taille des segments à 21 (i.e.  $\lambda = 21$  pour la méthode [Soille and Talbot, 2001]). Nous pouvons constater qu'inclure de l'épaisseur n'améliore pas les résultats, mais prendre en compte des variations de longueur (avec ou sans épaisseur) améliore l'estimation de l'orientation.

### 5.4.5 Application pour le filtrage d'images

Dans le cas d'images en niveaux de gris, en plus estimer des caractéristiques géométriques locales, notre outil permet également d'effectuer des filtrages de l'image.

En effet, nous pouvons remarquer qu'au cours du processus, il est possible d'extraire des images filtrées. Il s'agit des images résultantes du calcul de  $\max_{a,b}(Y_{a,b}(x))$  pour  $I$  et  $J = 255 - I$ .

Le [tableau 5.3](#) montre un exemple d'application de filtre. De gauche à droite, nous pouvons voir l'image originale, les images résultantes après le filtre par [Soille and Talbot, 2001] ( $\lambda = 21$ ) et par notre filtre avec uniquement la variation de longueur ( $i = 21$ ). Puisque notre méthode prend en compte différents éléments structurants, nous pouvons constater qu'il préserve mieux les structures de petite taille.

La deuxième ligne du [tableau 5.3](#) montre les cercles de la [figure 5.12](#) auxquels nous avons ajouté du bruit gaussien additif de moyenne 0 et d'écart-type 0,1 afin d'obtenir une image en niveaux de gris. En haut à droite un zoom sur la partie centrale de l'image. Comme Soille et Talbot fixent le paramètre de longueur, leur approche ne parvient pas à filtrer les zones où cette longueur ne convient pas.

La dernière ligne du [tableau 5.3](#) montre un dernier exemple. Nous pouvons voir que notre filtre préserve mieux les structures circulaires et est plus lisse aux niveaux des intersections.

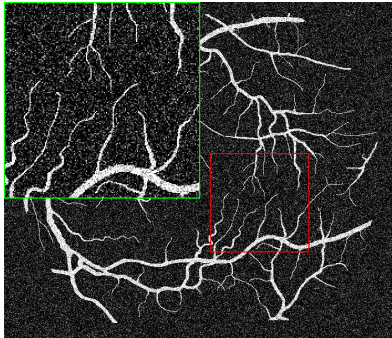
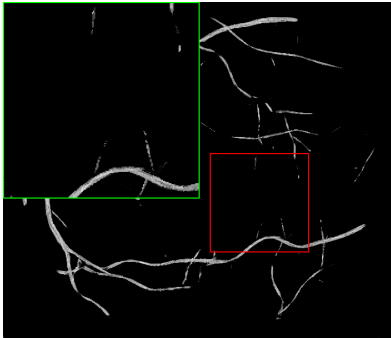
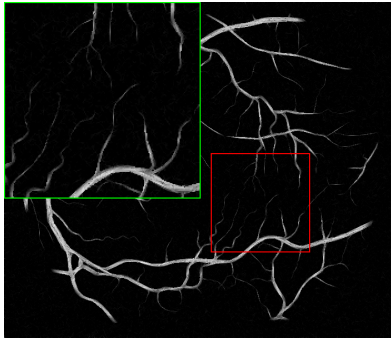
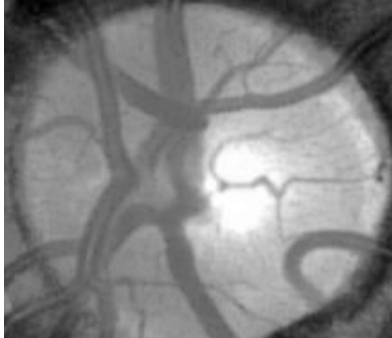
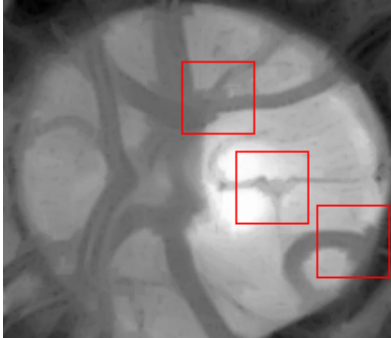
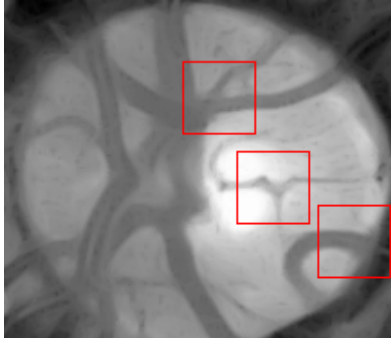
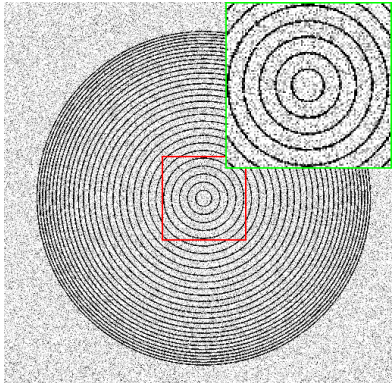
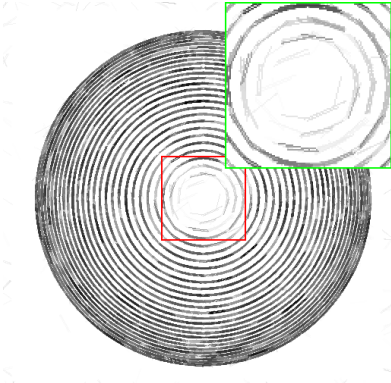
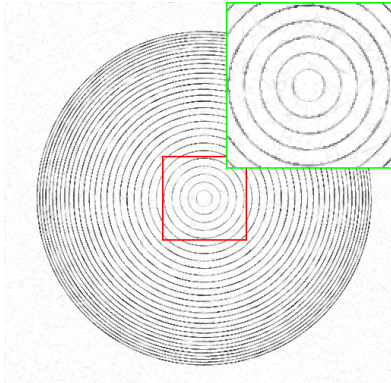
Image d'entrée	[Soille and Talbot, 2001]	Notre méthode (épaisseur fixe)
		
		
		

TABLE 5.3 – Image d'entrée (les images de vaisseaux de fond d'œil proviennent de [Hoover et al., 2000]), le filtrage selon [Soille and Talbot, 2001] et le filtrage avec notre méthode uniquement des variations de longueurs (épaisseur fixée à 1).

## 5.5 Conclusion

Dans ce chapitre, nous avons présenté un outil fondé sur la géométrie discrète et la morphologie mathématique afin d'estimer des caractéristiques locales d'un objet. Les principales caractéristiques que l'outil proposé permet d'évaluer sont : la longueur, la largeur (ou épaisseur selon le contexte) et l'orientation. D'autres caractéristiques pourraient être calculées.

Cet outil a d'abord été développé pour des images binaires. Notre outil se déploie en une panoplie de variantes qui permettent de s'adapter au cadre applicatif. Nous avons

proposé une variante qui permet de calculer l'orientation de manière plus précise, notamment dans le cas de cercles concentriques. Cependant, aucune évaluation quantitative n'a encore été faite pour le moment.

Nous avons ensuite essayé d'étendre notre outil aux images en niveaux de gris. Pour ce faire, nous avons proposé d'utiliser deux opérateurs usuels de la morphologie mathématique : l'érosion et la dilatation. Tout comme pour le cas binaire, cet outil offre la possibilité de s'adapter et de varier selon le contexte. Dans le cas des images en niveaux de gris, nous nous sommes intéressés particulièrement au calcul de l'orientation. Cependant, dans le cas d'image en niveaux de gris l'outil a une fonction supplémentaire : le filtrage. En effet, en plus de pouvoir estimer des caractéristiques locales, l'outil offre l'opportunité d'effectuer un filtrage directionnel. Il faut encore évaluer davantage l'outil pour les images en niveaux de gris. Mais, les premières évaluations qui ont été faites montrent des résultats intéressants.

# 6

## Conclusion

### Sommaire

---

<b>6.1 Contributions</b> . . . . .	184
<b>6.2 Perspectives</b> . . . . .	186

---

Ce chapitre synthétise les principaux résultats développés pendant mes années de thèse, et met l'accent sur les perspectives de recherche qui apparaissent dans la continuité directe des contributions présentées dans ce manuscrit.

Les travaux réalisés ont abordé les problématiques de l'analyse d'images RVB brutes de coupes transversales de grumes de bois pour estimer la qualité du bois. Comme nous avons pu déjà l'indiquer, le but de cette recherche est de valoriser le matériau bois dans un contexte plus global qu'est le réchauffement climatique qui accroît la demande de ce matériau. L'objectif est de réussir à extraire des informations visibles sur ce type d'images, afin d'estimer la qualité de la grume. Parmi ces informations, nous pouvons citer : les zones d'aubier et de duramen, la position du centre géométrique de la grume et de la moelle, ainsi que la détection des cernes de croissance.

Cette analyse est rendue difficile en raison de plusieurs facteurs :

- le choix de l'appareil d'acquisition ;
- les conditions d'acquisition ;
- le temps accordé.

En outre, cette analyse repose sur l'usage d'outils diversifiés issus du traitement d'images tels que la segmentation, le calcul de l'orientation ou bien de filtrage. Cependant, les recherches existantes qui abordent ces problématiques dans les mêmes conditions (type d'images, cadre applicatif, ...) sont très limitées. Cette thèse a relevé des défis et a fourni des contributions dans l'extraction d'informations nécessaires à l'évaluation de la qualité du bois à partir d'images RVB brutes de coupe transversale de grume de bois.

## 6.1 Contributions

Dans les paragraphes suivants, nous fournissons un bref résumé des travaux effectués dans cette thèse, avant d'énumérer les perspectives futures dans la section suivante.

Nous avons proposé de nouvelles méthodes pour estimer des caractéristiques nécessaires à l'estimation de la qualité du bois (voir le [chapitre 3](#)). Concernant la **segmentation de la coupe transversale** dans une image RVB, nous avons proposé d'évaluer les performances de quelques réseaux de neurones convolutifs [[Decelle and Jalilian, 2020](#)]. Nous avons également adapté des méthodes plus *classiques*, comme la méthode du K-Means ou celle des contours actifs. Toutes ces méthodes n'avaient pas encore été évaluées pour effectuer la segmentation des grumes. Les deux dernières méthodes (K-Means et contour actif) ont montré des résultats moins bons que les quatre réseaux de neurones. Ces mêmes réseaux ont des disparités entre eux. Certains ont une robustesse quel que soit le jeu d'images utilisé. D'autres ont des résultats différents selon le jeu d'images utilisé.

Nous avons proposé une méthode innovante pour estimer **la localisation de la moelle** dans une coupe transversale [[Decelle et al., 2021b](#)] (voir le [chapitre 4](#)). Cette méthode est basée sur **l'algorithme des colonies de fourmis**. L'avantage d'utiliser une telle méthode est d'avoir une solution algorithmique quant aux choix de l'accumulation des normales aux tangentes des cernes. Cette contribution a été comparée à différentes

méthodes de l'état de l'art ayant été conçues pour le même type d'image. Notre contribution montre des résultats aussi précis. Mais le point fort de notre méthode est le temps de calcul. Elle effectue l'estimation de la localisation de la moelle de manière rapide, proche du réseau de neurones convolutifs de l'état de l'art. Cette contribution prouve également l'intérêt de prendre en compte le choix des normales aux tangentes à accumuler. L'estimation de l'orientation des tangentes aux cernes n'est pas le seul élément clef dans la détection de la moelle.

Par la suite, nous avons proposé une **couche d'attention** (voir [section 6.2](#)), pour les réseaux de neurones convolutifs, basée sur des outils morphologiques (cf. [chapitre 3](#)). L'idée d'utiliser des couches d'attention n'est pas nouvelle. Cependant, l'utilisation des outils de morphologie mathématique dans ce cadre est limité. Notre couche d'attention a pour objectif d'éliminer les informations peu pertinentes au sein des cartes de caractéristiques du réseau, mais aussi de mettre en valeur les informations pertinentes de ces mêmes cartes. Un autre objectif d'une couche d'attention est de potentiellement réduire le nombre de paramètres d'un réseau de neurones. Cet objectif répond à l'enjeu de la rapidité d'évaluation de la qualité du bois. La couche d'attention proposée a été évaluée dans le cadre de la **segmentation du duramen**. Les résultats sont très encourageants : le nombre de paramètres a pu être réduit d'un facteur 10, le temps de calcul a marginalement augmenté (de l'ordre d'une dizaine de millisecondes) et les résultats sont meilleurs que d'autres couches d'attention qui ont été évaluées.

Après quoi, nous avons proposé une première approche pour identifier les limites de cernes de croissance (voir le [chapitre 3](#)). L'approche présentée permet de répondre à un enjeu : les conditions d'acquisition. En effet, ces conditions font qu'un certain nombre d'éléments, principalement dus au sciage, perturbent les outils de traitement d'images. L'outil développé permet d'effectuer un **filtrage bilatéral directionnel** tout en gardant les contours. À la suite de ça, nous avons proposé d'utiliser une décomposition par un quadtree, combinée avec l'outil de géométrie discrète développé au [chapitre 5](#), afin de mesurer au mieux la largeur moyenne des cernes dans une image RVB de coupe transversale de grume de bois. Bien que les résultats ne soient pas encore satisfaisant, ils restent prometteurs en perfectionnant les outils utilisés. En effet, une estimation visuelle des limites de cernes semble montrer de bons résultats. De plus, la méthode proposée montre que l'estimation de l'orientation des cernes reste un sujet important pour les problématiques abordées dans ce manuscrit.

Dans le dernier chapitre de ce manuscrit, nous nous sommes légèrement détachés des problématiques (voir le [chapitre 5](#)). En effet, ce chapitre se concentre sur un **filtre directionnel basé sur la géométrie discrète**, à savoir les segments naïfs symétriques, afin d'estimer des caractéristiques locales d'un ensemble connexe (ou d'une image binaire) [[Decelle et al., 2021a](#)]. Évidemment, cet outil n'est pas sans lien avec le sujet, puisque parmi les caractéristiques de l'outil développé, il y a le calcul d'une orientation locale utilisée pour calculer la largeur moyenne des cernes, mais aussi d'une largeur locale. L'outil offre une panoplie de variations permettant de s'adapter facilement à un contexte particulier. Nous avons aussi proposé une redéfinition des segments épais. L'outil présenté est pertinent

pour analyser des cercles concentriques qui rappellent la forme des cernes de croissance. Dans ce chapitre, nous avons aussi proposé une première extension de l'outil pour les images en niveaux de gris. Cet outil est encore en cours d'amélioration puisque qu'un certain nombre de restrictions, par rapport à sa version binaire, ont dû être mises en place pour traiter ce type d'images. Ceci étant dit, l'outil permet d'estimer une orientation, mais aussi d'effectuer un lissage d'une image bruitée. Malgré l'absence de résultats quantitatifs, quelques résultats qualitatifs ont mis en lumière que le fait d'approfondir cet outil pourrait s'avérer pertinent, aussi bien pour le traitement de coupe transversale, mais aussi dans un traitement d'images de manière générale.

## 6.2 Perspectives

Tout d'abord, l'une des pistes à étudier est d'évaluer les performances des différentes contributions sur d'autres essences de bois : cèdre, if, chêne merisier, pin sylvestre, ... De fait, la grande partie des évaluations réalisées ont porté sur le douglas. Nous pouvons donc espérer que nos méthodes proposées soient tout aussi performantes sur d'autres essences. Par exemple, le duramen du douglas est rouge/rose alors que celui de l'if est très rouge et celui du cèdre tend vers le marron. Les réseaux de neurones entraînés sur le douglas pour détecter le duramen doivent être ré-entraînés afin de prendre en compte ces variations colorimétriques des essences. Les rayons ligneux, rayon de cellules en direction de la moelle, sont très marqués sur l'essence du chêne. Ces derniers peuvent perturber les algorithmes proposés, principalement dans le calcul de l'orientation et dans la suppression des marques dues au sciage.

Nous pensons aussi que des efforts doivent être mis en œuvre pour améliorer la détection de la moelle, par exemple en déterminant de manière automatique certains paramètres de la méthode. En effet, bien que notre méthode offre de bonnes performances, elle nécessite un nombre non-négligeable de paramètres, certains ayant un effet marginal, d'autres étant cruciaux. Ces derniers doivent faire l'objet d'une détermination automatique pour permettre de mieux s'adapter aussi bien au choix de l'appareil d'acquisition qu'aux conditions d'acquisition, et même au temps qui est accordé pour effectuer le traitement de l'image.

Nous voudrions également voir si l'utilisation de l'algorithme des colonies de fourmis pour créer un espace d'accumulation, de manière générale, peut être pertinent dans d'autres cas. C'est-à-dire que le choix des accumulations serait fait avec un algorithme similaire à celui proposé. Par exemple, est-il possible d'effectuer une transformée de Hough pour la détection des droites en utilisant un tel algorithme ?

La méthode proposée est implémentée en C++ [Decelle et al., 2021b]. Son implémentation est actuellement séquentielle et peu optimisée. Dédiée à l'expérimentation et à la mise en application avec nos partenaires de l'INRAE, l'optimisation de la mémoire et des temps de calcul n'a été prise en compte que partiellement. Dans l'objectif d'une utilisation industrielle, une suite immédiate du travail effectué consiste à étudier les possibilités de parallélisation qui semblent prometteuses. D'autant plus que cela pourrait permettre de prendre un peu plus de temps de calcul dans l'estimation de l'orientation ou la suppression

des marques de sciage.

Concernant la couche d'attention, notre étude n'a porté que sur la segmentation du duramen, il serait pertinent d'évaluer ses performances sur des jeux de données génériques (par exemple les jeux MNIST ou COCO). Notre couche d'attention, reposant sur deux outils de morphologie mathématique, à savoir l'érosion et la dilatation, pourrait aussi faire l'étude d'améliorations via l'utilisation d'autres outils de morphologie mathématique comme l'ouverture ou la fermeture. Une analyse plus approfondie de cette couche d'attention devrait aussi être à faire : quels sont les éléments structurants que le réseau de neurones a appris ? D'ailleurs, est-ce que notre couche d'attention est pertinente dans la segmentation de la grume ? Voici des exemples de questions pour lesquelles il faudrait porter une attention particulière.

Nous voulons également améliorer l'outil de géométrie discrète que nous avons développé, aussi bien dans le cas d'images binaires que d'images en niveaux de gris.

Dans le cas binaire, le sujet qui focalise notre attention est le temps de calcul. Ce dernier est très coûteux, ce qui limite son usage en pratique. Il faut d'une part étudier minutieusement les points qui pourraient donner lieu à une diminution de la complexité en temps. D'autre part, la parallélisation est aussi à regarder de près. Cela a pour but d'avoir un temps de calcul expérimental beaucoup plus bas, et par la suite avoir une application sur des images de plus grandes tailles, une mise en œuvre dans un cadre applicatif (calcul de l'orientation, etc.).

Dans le cas d'images en niveaux de gris, il nous faut essayer de s'approcher plus du cas binaire. C'est-à-dire qu'il faut réussir à étendre les segments symétriques de droites discrètes de manière automatique. Il faut aussi voir pour intégrer les segments omis dans le cas en niveaux de gris. En effet, dans notre cas, nous n'avons que pris en compte les segments avec  $\mu$  mis à 0. Pour cela, l'utilisation de nouveaux filtres morphologiques (*rank-max filters*) est à envisager. Toujours dans un objectif d'une application industrielle, il faut implémenter les algorithmes efficaces pour calcul de l'érosion et de la dilatation par des segments de droites discrètes (basé sur les propriétés périodiques de ces segments à l'aide d'un histogramme par exemple).

Un autre point que nous devons souligner est le suivant. Bien que nous ayons réussi à détecter un ensemble d'éléments utiles pour l'estimation de la qualité du bois, une question demeure : est-il possible d'estimer la qualité du bois à partir de ces éléments détectés ? Autrement dit, il reste à faire un travail de classification de la qualité de la grume à partir des éléments estimés par les différentes approches présentées. Des méthodes de classification, comme les machines à vecteurs de support (SVM), les arbres de décisions ou les forêts d'arbres décisionnels (Random Forest), perceptron multicouche (MLP), ... doivent être testées pour faire cette classification.

Enfin, d'un point de vue plus technique, il s'agit aussi de mettre en œuvre un pipeline complet, de la segmentation de la grume à l'analyse des cernes, afin d'évaluer la qualité d'une grume en incorporant l'ensemble des contributions.



## Revue internationale à comité de relecture

### Soumis

Longuetaud, F., Pot, G., Mothe, F., Barthelemy, A., Decelle, R., Delconte, F., Ge, X., Guillaume, G., Mancini, T., Ravoajanahary, T., Butaud, J.-C., Collet, R., Debled-Rennesson, I., Marcon, B., Ngo, P., Roux, B., Viguier, J. Treetrace project database on traceability and quality assessment of round woods : Douglas fir sampling. 2022. *Annals of Forest Science*. cf. *chapitre 3*

## Conférence internationale à comité de relecture

### En cours

Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., et Longuetaud, F. Ligth U-NET with a new Morphological Attention Gate Model Application To Analyse Wood Sections.

### Publiés

Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., & Longuetaud, F. Ant Colony Optimization for Estimating Pith Position on Images of Tree Log Ends. In *Image Processing On Line (IPOL)*. 2022 cf. *chapitre 4*

Digital Straight Segment Filter for Geometric Description. Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., et Longuetaud, F. In *International Conference on Discrete Geometry and Mathematical Morphology (DGMM)*. 2021 cf. *chapitre 5*

Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., et Longuetaud, F. Pith Estimation on Tree Log End Images. In *Reproducible Research on Pattern Recognition (RRPR)*. 2021 cf. *chapitre 4*

Decelle, R., & Jalilian, E. Neural Networks for Cross-Section Segmentation in Raw Images of Log Ends. In *IPAS 2020-Fourth IEEE International Conference on Image Processing, Applications and Systems*. 2020, December cf. *chapitre 3*

Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., et Longuetaud, F. A new algorithm to automatically detect the pith on rough log-end images. In *21st International Nondestructive Testing and Evaluation (NDTE) of Wood Symposium*. 2019, September.

## Communications orales

Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., et Longuetaud, F. Description géométrique à l'aide de segments discrets. GDR IM, GT-GDMM, LORIA, France 2021.

Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., et Longuetaud, F. Détection automatique de la moelle sur des images d'extrémités de grumes de Douglas brutes de sciage. GDR Bois, Épinal, France 2019

## **Prix**

Prix du meilleur article étudiant, DGMM 2021.

Prix du meilleur Poster Session D, GDR Bois, Épinal, France 2019.

## **Lien Git**

<https://gitlab.com/Ryukhaan/treetrace>



# Acronymes

- ANR** Agence Nationale de la Recherche. 15
- APN** appareil photographique numérique. 16, 18, 19, 25, 72, 87
- CieLAB** CIELAB color space. 7, 9, 26, 27, 73, 97, 99, 101, 104, 105
- CO2** dioxyde de carbone. 21
- CT** computerized tomography. 21
- EMD** Earth Mover’s Distance. 70
- FWF** Fonds zur Förderung der wissenschaftlichen Forschung. 15
- GES** gaz à effet de serre. 21
- HSV** Hue Saturate Value. 9, 25, 73, 78, 80, 85, 105, 106
- INRAE** Institut national de recherche pour l’agriculture, l’alimentation et l’environnement. 15, 186
- LIDAR** Laser Imaging Detection And Ranging. 7, 16–18
- LORIA** Laboratoire Lorrain de Recherche en Informatique et ses Applications. 15
- LPB** Local Binary Patterns. 69
- RVB** Rouge Vert Bleu. 15, 16, 18, 21, 22, 24, 25



# Glossaire

**aubier** Partie du tronc la plus proche de l'écorce. C'est la partie tendre, jeune et vivante du bois qui transporte la sève. 22

**binarisation** La binarisation est un cas particulier de segmentation où l'on cherche à décomposer l'image en deux régions.. 28

**bois de trituration** Le bois de trituration est destiné à la fabrication de panneaux de fibres, pâtes de cellulose, panneaux de particules, etc.. 21

**cerne** Le bois qui est formé chaque année. 16, 21, 29, 64, 111

**dendrochronologie** Science traitant de l'étude et de la datation des cernes de croissance des arbres. 21, 74, 90, 96, 111

**duramen** Bois ayant subi une duraminisation. 27, 29

**grume** Une grume est un tronc coupé, ébranché et encore pourvus de son écorce. 15, 16, 30, 64, 65

**moelle** Centre biologique du bois. 64, 70, 74



# Bibliographie

- [Abhishek and Hamarneh, 2021] Abhishek, K. and Hamarneh, G. (2021). Matthews correlation coefficient loss for deep convolutional networks : Application to skin lesion segmentation. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 225–229. IEEE.
- [Abu-Ain et al., 2013] Abu-Ain, W., Abdullah, S. N. H. S., Bataineh, B., Abu-Ain, T., and Omar, K. (2013). Skeletonization algorithm for binary images. *Procedia Technology*, 11 :704–709.
- [Achanta et al., 2010] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2010). Slic superpixels. Technical report.
- [Anderson et al., 2005] Anderson, R., Kingsbury, N., and Fauqueur, J. (2005). Determining multiscale image feature angles from complex wavelet phases. In *International Conference Image Analysis and Recognition*, pages 490–498. Springer.
- [Andreu and Rinnhofer, 2003] Andreu, J.-P. and Rinnhofer, A. (2003). Modeling knot geometry in norway spruce from industrial ct images. In *Scandinavian Conference on Image Analysis*, pages 786–791. Springer.
- [Anwar et al., 2020] Anwar, S., Khan, S., and Barnes, N. (2020). A deep journey into super-resolution : A survey. *ACM Computing Surveys (CSUR)*, 53(3) :1–34.
- [Arthur and Vassilvitskii, 2006] Arthur, D. and Vassilvitskii, S. (2006). k-means++ : The advantages of careful seeding. Technical report, Stanford.
- [Badrinarayanan et al., 2017] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12) :2481–2495.
- [Basantia et al., 2018] Basantia, N., Nollet, L. M., and Kamruzzaman, M. (2018). *Hyper-spectral Imaging Analysis and Applications for Food Quality*. CRC Press.
- [Bayer, 1976] Bayer, B. (1976). Colour filter array. *United States of America patent*, 3971065.
- [Beaulieu and Dutilleul, 2019] Beaulieu, J. and Dutilleul, P. (2019). Applications of computed tomography (ct) scanning technology in forest research : A timely update and review. *Canadian journal of forest research*, 49(10) :1173–1188.
- [Becker et al., 1987] Becker, M., Geremia, F., and Schipfer, R. (1987). Bilan de santé actuel et rétrospectif du sapin (*abies alba mill.*) dans les vosges. etude écologique et dendrochronologique. In *Annales des sciences forestières*, volume 44, pages 379–402. EDP Sciences.



- [Berenstein and Lavine, 1988] Berenstein, C. A. and Lavine, D. (1988). On the number of digital straight line segments. *IEEE Trans. on pattern analysis and machine intelligence*, 10(6) :880–887.
- [Berglund et al., 2013] Berglund, A., Broman, O., Grönlund, A., and Fredriksson, M. (2013). Improved log rotation using information from a computed tomography scanner. *Computers and electronics in agriculture*, 90 :152–158.
- [Borianne et al., 2011] Borianne, P., Pernaudat, R., and Subsol, G. (2011). Automated delineation of tree-rings in x-ray computed tomography images of wood. In *2011 18th IEEE International Conference on Image Processing*, pages 437–440. IEEE.
- [Borsellino and Poggio, 1973] Borsellino, A. and Poggio, T. (1973). Convolution and correlation algebras. *Kybernetik*, 13(2) :113–122.
- [Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6) :679–698.
- [Cerda et al., 2007] Cerda, M., Hitschfeld-Kahler, N., and Mery, D. (2007). Robust tree-ring detection. In *Pacific-Rim Symposium on Image and Video Technology*, pages 575–585. Springer.
- [Chan et al., 2005] Chan, R. H., Ho, C.-W., and Nikolova, M. (2005). Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on image processing*, 14(10) :1479–1485.
- [Chan and Vese, 2001a] Chan, T. and Vese, L. (2001a). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2) :266–277.
- [Chan and Vese, 2001b] Chan, T. F. and Vese, L. A. (2001b). Active contours without edges. *IEEE Transactions on image processing*, 10(2) :266–277.
- [Chen et al., 2013] Chen, Q., Li, D., and Tang, C.-K. (2013). Knn matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9) :2175–2188.
- [Chen et al., 2014] Chen, S., Pande, A., and Mohapatra, P. (2014). Sensor-assisted facial recognition : an enhanced biometric authentication system for smartphones. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 109–122.
- [CIPA, 2020] CIPA (2020). *Total Shipments of Digital Still Cameras*. Camera & Imaging Products Association.
- [Conner et al., 1998] Conner, W. S., Schowengerdt, R. A., Munro, M., and Hughes, M. K. (1998). Design of a computer vision based tree ring dating system. In *1998 IEEE Southwest Symposium on Image Analysis and Interpretation (Cat. No. 98EX165)*, pages 256–261. IEEE.
- [Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE.
- [De Rivaz, 2001] De Rivaz, P. F. C. (2001). *Complex wavelet based image analysis and synthesis*. PhD thesis, University of Cambridge.

- 
- [Decelle and Jalilian, 2020] Decelle, R. and Jalilian, E. (2020). Neural networks for cross-section segmentation in raw images of log ends. In *2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS)*, pages 131–137. IEEE.
- [Decelle et al., 2021a] Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., and Longuetaud, F. (2021a). Digital straight segment filter for geometric description. In *International Conference on Discrete Geometry and Mathematical Morphology*, pages 255–268. Springer.
- [Decelle et al., 2021b] Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., and Longuetaud, F. (2021b). Pith estimation on tree log end images. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 101–120. Springer.
- [Dimitrov and Golparvar-Fard, 2015] Dimitrov, A. and Golparvar-Fard, M. (2015). Segmentation of building point cloud models including detailed architectural/structural features and mep systems. *Automation in Construction*, 51 :32–45.
- [Dorigo and Blum, 2005] Dorigo, M. and Blum, C. (2005). Ant colony optimization theory : A survey. *Theoretical computer science*, 344(2-3) :243–278.
- [Dorigo et al., 1996] Dorigo, M., Maniezzo, V., Colorni, A., et al. (1996). Ant system : optimization by a colony of cooperating agents. *IEEE Transactions on Systems, man, and cybernetics, Part B : Cybernetics*, 26(1) :29–41.
- [Dorksen-Reiter and Debled-Rennesson, 2006] Dorksen-Reiter, H. and Debled-Rennesson, I. (2006). *Convex and Concave Parts of digital Curves*, pages 145–159. Springer Netherlands.
- [Easton Jr, 2010] Easton Jr, R. L. (2010). *Digital Image Processing I*. September.
- [Entacher et al., 2007] Entacher, K., Planitzer, D., and Uhl, A. (2007). Towards an automated generation of tree ring profiles from ct-images. In *Image and Signal Processing and Analysis (ISPA). 5th International Symposium on*, pages 174–179. IEEE.
- [Even and Ngo, 2020] Even, P. and Ngo, P. (2020). Live extraction of curvilinear structures from lidar raw data. In *XXIV ISPRS Congress*, volume 2, pages 211–219.
- [Fabijańska and Danek, 2018] Fabijańska, A. and Danek, M. (2018). Deepdendro—a tree rings detector based on a deep convolutional neural network. *Computers and electronics in agriculture*, 150 :353–363.
- [Fabijanska et al., 2017] Fabijanska, A., Danek, M., Barniak, J., and Piórkowski, A. (2017). Towards automatic tree rings detection in images of scanned wood samples. *Computers and Electronics in Agriculture*, 140 :279–289.
- [FCBA, 2020] FCBA (2020). *Memento 2020*. Champs-sur-Marne, France.
- [Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395.
- [Franchi et al., 2020] Franchi, G., Fehri, A., and Yao, A. (2020). Deep morphological networks. *Pattern Recognition*, 102 :107246.
- [Freeman et al., 1991] Freeman, W. T., Adelson, E. H., et al. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9) :891–906.

- [Freyburger et al., 2009] Freyburger, C., Longuetaud, F., Mothe, F., Constant, T., and Leban, J.-M. (2009). Measuring wood density by means of x-ray computer tomography. *Annals of forest science*, 66(8) :804–804.
- [Funt and Bryant, 1987] Funt, B. V. and Bryant, E. C. (1987). Detection of internal log defects by automatic interpretation of computer tomography images. *Forest Products Journal*, 37(1) :56–62.
- [G. Hardy, 1989] G. Hardy, E. W. (1989). *An introduction to the theory of numbers*.
- [Gazo et al., 2020] Gazo, R., Vanek, J., Abdul\_Massih, M., and Benes, B. (2020). A fast pith detection for computed tomography scanned hardwood logs. *Computers and Electronics in Agriculture*, 170 :105107.
- [George et al., 2020] George, P., Maurin, E., and Trouy, M.-C. (2020). *L’essence du bois*. L’essence du bois. Editions du Patrimoine.
- [Getreuer, 2012] Getreuer, P. (2012). Chan-Vese Segmentation. *Image Processing On Line*, 2 :214–224.
- [Girshick, 2015] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.
- [Glasbey, 1993] Glasbey, C. A. (1993). An analysis of histogram-based thresholding algorithms. *CVGIP : Graphical models and image processing*, 55(6) :532–537.
- [Gonzalez and Woods, 2018] Gonzalez, R. C. and Woods, R. E. (2018). Digital image processing.
- [Gonzalez-Hidalgo et al., 2018] Gonzalez-Hidalgo, M., Massanet, S., Mir, A., and Ruiz-Aguilera, D. (2018). Improving salt and pepper noise removal using a fuzzy mathematical morphology-based filter. *Applied Soft Computing*, 63 :167–180.
- [Granlund and Knutsson, 1994] Granlund, G. H. and Knutsson, H. (1994). *Signal processing for computer vision*. Springer Science & Business Media.
- [Hast, 2014] Hast, A. (2014). Simple filter design for first and second order derivatives by a double filtering approach. *Pattern Recognition Letters*, 42 :65–71.
- [He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.
- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Hodges et al., 1990] Hodges, D. G., Anderson, W. C., and McMillin, C. W. (1990). The economic potential of ct scanners for hardwood sawmills. *Forest Products Journal* 40 (3), pages 65–69.
- [Hong et al., 1998] Hong, L., Wan, Y., and Jain, A. (1998). Fingerprint image enhancement : algorithm and performance evaluation. *IEEE transactions on pattern analysis and machine intelligence*, 20(8) :777–789.
- [Hoover et al., 2000] Hoover, A., Kouznetsova, V., and Goldbaum, M. (2000). Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical imaging*, 19(3) :203–210.

- 
- [Horowitz, 1974] Horowitz, S. L. (1974). Picture segmentation by a directed split-and-merge procedure. In *IJ CPR*, pages 424–433.
- [Hough, 1962] Hough, P. V. (1962). Method and means for recognizing complex patterns. US Patent 3,069,654.
- [Howard et al., 2017] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*.
- [Hu and Li, 2012] Hu, J. and Li, S. (2012). The multiscale directional bilateral filter and its application to multisensor image fusion. *Information Fusion*, 13(3) :196–206.
- [Hu et al., 2018] Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141.
- [IGN, 2020] IGN (2020). *Le memento inventaire forestier - édition 2020*. Institut national de l’information géographique et forestière, Saint-Mandé, France.
- [Jain and Chandrasekaran, 1982] Jain, A. and Chandrasekaran, B. (1982). Dimensionality and sample size considerations in pattern recognition practice. In *Classification Pattern Recognition and Reduction of Dimensionality*, volume 2 of *Handbook of Statistics*, pages 835–855. Elsevier.
- [Jammalamadaka and Sengupta, 2001] Jammalamadaka, S. R. and Sengupta, A. (2001). *Topics in circular statistics*, volume 5. world scientific.
- [Johansson et al., 2013] Johansson, E., Johansson, D., Skog, J., and Fredriksson, M. (2013). Automated knot detection for high speed computed tomography on pinus sylvestris l. and picea abies (l.) karst. using ellipse fitting in concentric surfaces. *Computers and electronics in agriculture*, 96 :238–245.
- [Jones, 1943] Jones, L. A. (1943). The historical background and evolution of the colorimetry report. *J. Opt. Soc. Am.*, 33(10) :534–543.
- [Kang et al., 2018] Kang, G., Li, J., and Tao, D. (2018). Shakeout : A new approach to regularized deep neural network training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5) :1245–1258.
- [Karsa and Shmueli, 2018] Karsa, A. and Shmueli, K. (2018). Segue : A speedy region-growing algorithm for unwrapping estimated phase. *IEEE transactions on medical imaging*, 38(6) :1347–1357.
- [Kass et al., 1988] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes : Active contour models. *International journal of computer vision*, 1(4) :321–331.
- [Kennel et al., 2015] Kennel, P., Borianne, P., and Subsol, G. (2015). An automated method for tree-ring delineation based on active contours guided by dt-cwt complex coefficients in photographic images : Application to abies alba wood slice images. *Computers and Electronics in Agriculture*, 118 :204–214.
- [Khan et al., 2020] Khan, S., Lee, D.-H., Khan, M. A., Gilal, A. R., Iqbal, J., and Waqas, A. (2020). Efficient and improved edge detection via a hysteresis thresholding method. *Current Science (00113891)*, 118(6).

- [Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.
- [Kingsbury, 2001] Kingsbury, N. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10(3) :234–253.
- [Kirsch, 1971] Kirsch, R. A. (1971). Computer determination of the constituent structure of biological images. *Computers and biomedical research*, 4(3) :315–328.
- [Klette and Rosenfeld, 2004] Klette, R. and Rosenfeld, A. (2004). *Digital geometry – Geometric methods for digital picture analysis*. Morgan Kaufmann.
- [Knutsson, 1982] Knutsson, H. (1982). *Filtering and reconstruction in image processing*. PhD thesis, Linköping University Electronic Press.
- [Koplowitz et al., 1990] Koplowitz, J., Lindenbaum, M., and Bruckstein, A. (1990). The number of digital straight lines on an  $n^* n$  grid. *IEEE Trans. on Information Theory*, 36(1) :192–197.
- [Krähenbühl et al., 2014] Krähenbühl, A., Roussel, J.-R., Kerautret, B., Debled-Rennesson, I., Mothe, F., and Longuetaud, F. (2014). Segmentation robuste de nœuds à partir de coupes tangentielles issues d’images tomographiques de bois. In *Reconnaissance de Formes et Intelligence Artificielle (RFIA) 2014*.
- [Krähenbühl et al., 2012] Krähenbühl, A., Kerautret, B., Debled-Rennesson, I., Longuetaud, F., and Mothe, F. (2012). Knot detection in x-ray ct images of wood. In *International Symposium on Visual Computing*, pages 209–218. Springer.
- [Kurdthongmee, 2020] Kurdthongmee, W. (2020). A comparative study of the effectiveness of using popular dnn object detection algorithms for pith detection in cross-sectional images of parawood. *Heliyon*, 6(2).
- [Kurdthongmee and Suwannarat, 2019] Kurdthongmee, W. and Suwannarat, K. (2019). An efficient algorithm to estimate the pith location on an untreated end face image of a rubberwood log taken with a normal camera. *European Journal of Wood and Wood Products*, 77(5) :919–929.
- [Kurdthongmee et al., 2018] Kurdthongmee, W., Suwannarat, K., Panyuen, P., and Sae-Ma, N. (2018). A fast algorithm to approximate the pith location of rubberwood timber from a normal camera image. In *15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6. IEEE.
- [Lachaud, 2010] Lachaud, J. (2010). Digital shape analysis with maximal segments. In *Applications of Discrete Geometry and Mathematical Morphology*, pages 14–27.
- [Lachaud et al., 2007] Lachaud, J.-O., Vialard, A., and de Vieilleville, F. (2007). Fast, accurate and convergent tangent estimation on digital contours. *Image and Vision Computing*, 25(10) :1572–1587.
- [Laggoune et al., 2005] Laggoune, H., Guesdon, V., et al. (2005). Tree ring analysis. In *Canadian Conference on Electrical and Computer Engineering, 2005.*, pages 1574–1577. IEEE.
- [Le et al., 2014] Le, A. V., Jung, S.-W., and Won, C. S. (2014). Directional joint bilateral filter for depth images. *Sensors*, 14(7) :11362–11378.

- 
- [Lettvin et al., 1959] Lettvin, J. Y., Maturana, H. R., McCulloch, W. S., and Pitts, W. H. (1959). What the frog’s eye tells the frog’s brain. *Proceedings of the IRE*, 47(11) :1940–1951.
- [Lin et al., 2017] Lin, G., Milan, A., Shen, C., and Reid, I. (2017). Refinenet : Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934.
- [Longuetaud et al., 2004] Longuetaud, F., Leban, J.-M., Mothe, F., Kerrien, E., and Berger, M.-O. (2004). Automatic detection of pith on ct images of spruce logs. *Computers and Electronics in Agriculture*, 44(2) :107–119.
- [Lu et al., 2010] Lu, S., Su, B., and Tan, C. L. (2010). Document image binarization using background estimation and stroke edges. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(4) :303–314.
- [Ma et al., 2017] Ma, J., Du, K., Zhang, L., Zheng, F., Chu, J., and Sun, Z. (2017). A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing. *Computers and Electronics in Agriculture*, 142 :110–117.
- [Mamalet and Garcia, 2012] Mamalet, F. and Garcia, C. (2012). Simplifying convnets for fast learning. In *International Conference on Artificial Neural Networks*, pages 58–65. Springer.
- [Mamrosh and Moore, 2015] Mamrosh, J. L. and Moore, D. D. (2015). Using google reverse image search to decipher biological images. *Current protocols in molecular biology*, 111(1) :19–13.
- [Masson-Delmotte et al., 2021] Masson-Delmotte, V., Zhai, P., Pirani, A., Connors, S. L., Péan, C., Berger, S., Caud, N., Chen, Y., Goldfarb, L., Gomis, M. I., Huang, M., Leitzell, K., Lonnoy, E., Matthews, J., Maycock, T. K., Waterfield, T., Yelekçi, O., Yu, R., and (eds.), B. Z. (2021). *Climate Change 2021 : The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, chapter IPCC, 2021 : Summary for Policymakers. Cambridge University Press.
- [Matheron, 1967] Matheron, G. (1967). Eléments pour une théorie des milieux poreux.
- [Matthews, 1975] Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2) :442–451.
- [Medina-Carnicer et al., 2009] Medina-Carnicer, R., Carmona-Poyato, A., Muñoz-Salinas, R., and Madrid-Cuevas, F. J. (2009). Determining hysteresis thresholds for edge detection by combining the advantages and disadvantages of thresholding methods. *IEEE transactions on image processing*, 19(1) :165–173.
- [Merveille et al., 2017] Merveille, O., Naegel, B., Talbot, H., Najman, L., and Passat, N. (2017). 2D Filtering of Curvilinear Structures by Ranking the Orientation Responses of Path Operators (RORPO). *Image Processing On Line*, 7 :246–261.
- [Merveille et al., 2018] Merveille, O., Talbot, H., Najman, L., and Passat, N. (2018). Curvilinear structure analysis by ranking the orientation responses of path operators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2) :304–317.

- [Misra et al., 2021] Misra, D., Nalamada, T., Arasanipalai, A. U., and Hou, Q. (2021). Rotate to attend : Convolutional triplet attention module. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3139–3148.
- [Moghaddam and Cheriet, 2012] Moghaddam, R. F. and Cheriet, M. (2012). Adotsu : An adaptive and parameterless generalization of otsu’s method for document image binarization. *Pattern Recognition*, 45(6) :2419–2431.
- [Mondal et al., 2019] Mondal, R., Purkait, P., Santra, S., and Chanda, B. (2019). Morphological networks for image de-raining. In *International Conference on Discrete Geometry for Computer Imagery*, pages 262–275. Springer.
- [Mongardini and Radzikowski, 2020] Mongardini, J. and Radzikowski, A. (2020). *Global Smartphone Sales May Have Peaked : What Next ?*
- [Mumford and Shah, 1989] Mumford, D. B. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*.
- [Nasser et al., 2018] Nasser, H., Ngo, P., and Debled-Rennesson, I. (2018). Dominant point detection based on discrete curve structure and applications. *Journal of Computer and System Sciences*, 95 :177–192.
- [Nguyen et al., 2020] Nguyen, V.-T., Constant, T., Kerautret, B., Debled-Rennesson, I., and Colin, F. (2020). A machine-learning approach for classifying defects on tree trunks using terrestrial lidar. *Computers and Electronics in Agriculture*, 171 :105332.
- [Nguyen et al., 2018] Nguyen, V. T. et al. (2018). *Estimation de la qualité de bois ronds et d’arbres sur pied par Lidar terrestre*. PhD thesis, Paris, AgroParisTech.
- [Norell, 2011] Norell, K. (2011). Automatic counting of annual rings on *Pinus sylvestris* end faces in sawmill industry. *Computers and Electronics in Agriculture*, 75(2) :231–237.
- [Norell and Borgefors, 2008] Norell, K. and Borgefors, G. (2008). Estimation of pith position in untreated log ends in sawmill environments. *Computers and Electronics in Agriculture*, 63(2) :155 – 167.
- [Oktay et al., 2018] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., et al. (2018). Attention u-net : Learning where to look for the pancreas. *arXiv preprint arXiv :1804.03999*.
- [Österberg et al., 2004] Österberg, P., Ihalainen, H., and Ritala, R. (2004). Method for analyzing and classifying wood quality through local 2d-spectrum of digital log end images. In *Proceedings of International Conference on Advanced Optical Diagnostics in Fluids, Solids and Combustion, VSJ-SPIE’04, Tokyo, Japan*.
- [Otsu, 1979] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1) :62–66.
- [Paris et al., 2009] Paris, S., Kornprobst, P., Tumblin, J., and Durand, F. (2009). *Bilateral filtering : Theory and applications*. Now Publishers Inc.
- [Powers, 2020] Powers, D. M. (2020). Evaluation : from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv :2010.16061*.

- 
- [Provençal and Lachaud, 2009] Provençal, X. and Lachaud, J.-O. (2009). Two linear-time algorithms for computing the minimum length polygon of a digital contour. In Brlek, S., Reutenauer, C., and Provençal, X., editors, *Discrete Geometry for Computer Imagery*, pages 104–117. Springer Berlin Heidelberg.
- [Qiao et al., 2019] Qiao, S., Wang, H., Liu, C., Shen, W., and Yuille, A. (2019). Weight standardization. *arXiv preprint arXiv :1903.10520*.
- [Raatevaara et al., 2020] Raatevaara, A., Korpunen, H., Tiitta, M., Tomppo, L., Kulju, S., Antikainen, J., and Uusitalo, J. (2020). Electrical impedance and image analysis methods in detecting and measuring scots pine heartwood from a log end during tree harvesting. *Computers and Electronics in Agriculture*, 177 :105690.
- [Rais et al., 2017] Rais, A., Ursella, E., Vicario, E., and Giudiceandrea, F. (2017). The use of the first industrial x-ray ct scanner increases the lumber recovery value : case study on visually strength-graded douglas-fir timber. *Annals of forest science*, 74(2) :28.
- [Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- [Redmon and Farhadi, 2018] Redmon, J. and Farhadi, A. (2018). Yolov3 : An incremental improvement. *arXiv preprint arXiv :1804.02767*.
- [Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn : Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.
- [Reveillès, 1991] Reveillès, J.-P. (1991). *Géométrie discrete, calcul en nombres entiers et algorithmique*. PhD thesis, Université Louis Paster.
- [Reveillès, 1991] Reveillès, J.-P. (1991). *Géométrie discrete, calcul en nombres entiers et algorithmique*.
- [Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- [Ruder, 2016] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv :1609.04747*.
- [Ruifrok et al., 2001] Ruifrok, A. C., Johnston, D. A., et al. (2001). Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4) :291–299.
- [Samet, 1984] Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2) :187–260.
- [Sarifuddin et al., 2005] Sarifuddin, M., Missaoui, R., and Laggoune, H. (2005). Technique de détection de contours flous et bruités. In *Actes de la conférence SETIT*. Citeseer.
- [Schraml and Uhl, 2013] Schraml, R. and Uhl, A. (2013). Pith estimation on rough log end images using local fourier spectrum analysis. In *Proceedings of the 14th Conference on Computer Graphics and Imaging (CGIM'13), Innsbruck, AUT*.



- [Schraml and Uhl, 2014] Schraml, R. and Uhl, A. (2014). Similarity based cross-section segmentation in rough log end images. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 614–623. Springer.
- [Serra, 1982] Serra, J. (1982). Mathematical morphology and image analysis. *SIAM, Washington DC à paraître*.
- [Shorten and Khoshgoftaar, 2019] Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1) :1–48.
- [Singh et al., 2012] Singh, T. R., Roy, S., Singh, O. I., Sinam, T., Singh, K., et al. (2012). A new local adaptive thresholding technique in binarization. *arXiv preprint arXiv :1201.5227*.
- [Snyman et al., 2005] Snyman, J. A., Wilke, D. N., et al. (2005). *Practical mathematical optimization*. Springer.
- [Sobel and Feldman, 1968] Sobel, I. and Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272.
- [Soille, 1994] Soille, P. (1994). Generalized geodesy via geodesic time. *Pattern Recognition Letters*, 15(12) :1235–1240.
- [Soille et al., 1996] Soille, P., Breen, E. J., and Jones, R. (1996). Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5) :562–567.
- [Soille and Misson, 2001] Soille, P. and Misson, L. (2001). Tree ring area measurements using morphological image analysis. *Canadian Journal of Forest Research*, 31(6) :1074–1083.
- [Soille and Talbot, 2001] Soille, P. and Talbot, H. (2001). Directional morphological filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11) :1313–29.
- [Som et al., 1993] Som, S., Davis, J., Wells, P., and Svalbe, I. (1993). Morphology methods for processing tomographic images of wood. *Digital Image Computing : Techniques and Applications (DICTA)*, 8 :564–571.
- [Stängle et al., 2015] Stängle, S. M., Brüchert, F., Heikkilä, A., Usenius, T., Usenius, A., and Sauter, U. H. (2015). Potentially increased sawmill yield from hardwoods using x-ray computed tomography for knot detection. *Annals of forest science*, 72(1) :57–65.
- [Stern, 1858] Stern, M. (1858). Über eine verallgemeinerung der kreistheilung. *Journal für die reine und angewandte Mathematik*, 55 :193–220.
- [Sun et al., 2017] Sun, M., Song, Z., Jiang, X., Pan, J., and Pang, Y. (2017). Learning pooling for convolutional neural network. *Neurocomputing*, 224 :96–104.
- [Sundari and Kumar, 2014] Sundari, P. M. and Kumar, S. B. R. (2014). An approach for dendroclimatology using image processing techniques. In *2014 World Congress on Computing and Communication Technologies*, pages 234–236. IEEE.
- [Taylor et al., 1984] Taylor, F. W., Wagner, F. G., McMillin, C. W., Morgan, I. L., and Hopkins, F. F. (1984). Locating knots by industrial tomography—a feasibility study. *Forest Products Journal* 34 (5) : 42-46.

- 
- [Tian et al., 2008] Tian, J., Yu, W., and Xie, S. (2008). An ant colony optimization algorithm for image edge detection. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 751–756. IEEE.
- [Tomasi and Manduchi, 1998] Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE.
- [Trouy and Triboulot, 2019] Trouy, M.-C. and Triboulot, P. (2019). *Matériau bois-Structure et caractéristiques*.
- [Turroni et al., 2011] Turroni, F., Maltoni, D., Cappelli, R., and Maio, D. (2011). Improving fingerprint orientation extraction. *IEEE Trans. on Inform. Forensics and Security*, 6(3) :1002–1013.
- [Vacavant, 2017] Vacavant, A. (2017). A novel definition of robustness for image processing algorithms. In Kerautret, B., Colom, M., and Monasse, P., editors, *Reproducible Research in Pattern Recognition*, pages 75–87, Cham. Springer International Publishing.
- [van Herk, 1992] van Herk, M. (1992). A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognition Letters*, 13(7) :517 – 521.
- [Venkatesh and Seelamantula, 2015] Venkatesh, M. and Seelamantula, C. S. (2015). Directional bilateral filters. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1578–1582. IEEE.
- [Vincent and Soille, 1991] Vincent, L. and Soille, P. (1991). Watersheds in digital spaces : an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(06) :583–598.
- [Vincent et al., 2008] Vincent, P., Larochele, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.
- [Wang et al., 2017] Wang, M., Liu, X., Gao, Y., Ma, X., and Soomro, N. Q. (2017). Superpixel segmentation : A benchmark. *Signal Processing : Image Communication*, 56 :28–39.
- [Wei et al., 2011] Wei, Q., Leblon, B., and La Rocque, A. (2011). On the use of x-ray computed tomography for determining wood properties : a review. *Canadian journal of forest research*, 41(11) :2120–2140.
- [Wimmer et al., 2021] Wimmer, G., Schraml, R., Hofbauer, H., Petutschnigg, A., and Uhl, A. (2021). Two-stage cnn-based wood log recognition. *CoRR*, abs/2101.04450.
- [Woo et al., 2018] Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). Cbam : Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19.
- [Wu, 2009] Wu, H. (2009). Global stability analysis of a general class of discontinuous neural networks with linear growth activation functions. *Information Sciences*, 179(19) :3432–3441.

- [Xu et al., 2015] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell : Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.
- [Yang et al., 2020] Yang, H.-H., Yang, C.-H. H., and Wang, Y.-C. F. (2020). Wavelet channel attention module with a fusion network for single image deraining. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 883–887. IEEE.
- [Yuan et al., 2015] Yuan, X.-c., Wu, L.-s., and Peng, Q. (2015). An improved otsu method using the weighted object variance for defect detection. *Applied surface science*, 349 :472–484.
- [Zbontar et al., 2021] Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins : Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR.
- [Zhang et al., 2015] Zhang, K., Zhang, L., Lam, K.-M., and Zhang, D. (2015). A level set approach to image segmentation with intensity inhomogeneity. *IEEE transactions on cybernetics*, 46(2) :546–557.
- [Zhou et al., 2012] Zhou, H., Feng, R., Huang, H.-h., Lin, E.-p., and Yu, J.-l. (2012). Method of tree-ring image analysis for dendrochronology. *Optical Engineering*, 51(7) :077202.
- [Zhu et al., 2021a] Zhu, B., Hofstee, P., Lee, J., and Al-Ars, Z. (2021a). An attention module for convolutional neural networks. In *International Conference on Artificial Neural Networks*, pages 167–178. Springer.
- [Zhu et al., 2021b] Zhu, L., Spachos, P., Pensini, E., and Plataniotis, K. N. (2021b). Deep learning and machine vision for food processing : A survey. *Current Research in Food Science*, 4 :233–249.

## Résumé

Dans le contexte du changement climatique et de son atténuation, du développement de la bioéconomie circulaire, et d'une pression accrue qui en découle sur la ressource en bois, un des axes de recherche prioritaire est l'optimisation de la transformation de cette ressource qui peut se faire à différents niveaux. L'objectif ici est d'extraire des informations sur la qualité du bois à partir de l'analyse de sections transversales du grumes ou billons de bois en forêt ou en scierie. Pour estimer cette qualité, plusieurs caractéristiques visibles peuvent être extraites : zones d'aubier et de duramen, position de la moelle et du centre géométrique, le nombre de cernes et leur largeur.

Dans un premier temps, nous nous intéressons à la segmentation de la grume dans l'image. Cette segmentation rend plus simple l'analyse des autres caractéristiques et permet de localiser le centre géométrique. Pour cela, nous proposons plusieurs approches. D'abord, des méthodes classiques issues du traitement d'images sont abordées, comme la méthode des K-Means ou les contours actifs. Nous utilisons également des réseaux de neurones convolutifs. Nous montrons l'avantage des réseaux de neurones par rapport à ces deux autres méthodes.

La deuxième caractéristique estimée est la zone de duramen (zone centrale plus colorée). Nous proposons une nouvelle couche d'attention pour les réseaux de neurones utilisant la morphologie mathématique moins souvent utilisée. Les couches d'attention ont permis aux réseaux d'être plus performants en se focalisant sur les informations les plus pertinentes. Dans notre cadre, l'objectif de cette couche est double : réduire la quantité de paramètres et augmenter les performances. Notre couche d'attention montre de meilleures performances par rapport à d'autres couches d'attention.

Dans un troisième temps, nous proposons d'analyser les cernes. Notre méthode est en trois grandes étapes. D'abord, un lissage directionnel pour rehausser les cernes (tout en gardant au mieux les contours) et réduire à la fois la texture intracernes et les marques de sciage. Puis, un seuillage adaptatif pour déterminer les zones de cernes potentiels. Enfin, un deuxième seuillage afin d'avoir les limites de cernes. À partir de la segmentation finale, l'analyse des cernes (nombre, largeur moyenne, etc.) est rendue possible.

Enfin, l'estimation de la position de la moelle est abordée. Nous proposons une nouvelle approche originale basée sur l'algorithme des colonies de fourmis pour estimer la position de la moelle. L'utilisation de cet algorithme permet de s'abstraire d'une étape habituelle, à savoir l'accumulation des normales aux tangentes des cernes. Notre méthode montre de nombreux avantages par rapport aux approches de l'état de l'art, réseaux de neurones inclus.

Dans une dernière partie, nous présenterons un travail en géométrie discrète : un filtre directionnel. Il estime les segments les plus longs en tout point d'un ensemble connexe. Ce travail a pour objectif d'être appliqué dans l'estimation des caractéristiques des cernes.

**Mots-clés:** Traitement d'images, Segmentation, Algorithmes de colonies de fourmis, Réseaux de neurones, Qualité du bois, Géométrie Discrète

# Abstract

In the context of climate change and its mitigation, the development of the circular bioeconomy, and the resulting increased pressure on the wood resource, one of the priority research areas is to optimize the transformation of this resource, which can be done at different levels. The objective here is to extract information on wood quality from the analysis of cross-sections of logs, located in the forest or in the sawmill. To estimate this quality, several visible characteristics can be extracted : sapwood and heartwood areas, position of the pith and geometric center, number of tree rings and their width.

First, we are interested in the segmentation of the log in the image. This segmentation makes easier the analysis of other features and allows us to locate the geometric center. For this, we propose several approaches. First, classical methods from image processing are discussed, such as the K-Means method or active contours. We also use convolutional neural networks. We show the advantage of neural networks over the other methods.

The second estimated feature is the duramen area (more colored central area). We propose a new attention layer for neural networks using the less commonly used mathematical morphology. Attention layers have allowed networks to outperform by focusing on the most relevant information. In our context, this layer has two goals : reducing the amount of parameters and increasing performance. Our attention layer shows better performances compared to other attention layers.

In the third step, we propose to analyze the tree rings. Our method is in three main steps. First, a directional smoothing to enhance the rings ( keeping the edges as well as possible) and reduce both the intra tree rings texture and the saw marks. Then, an adaptive thresholding to segment the areas of potential tree rings. Finally, a second thresholding in order to have the ring limits. From the final segmentation, the analysis of the rings (number, average width, etc.) is made possible.

Finally, the estimation of the pith position is addressed. We propose a new original approach based on the ant colony algorithm to estimate the location of the pith. Using this approach allows to skip a usual step, namely the way to accumulate normals at the tree ring tangents. Our method shows many advantages compared to state-of-the-art approaches, including neural networks.

In a last part, we will present a work in discrete geometry : a directional filtering. It estimates the maximal digital straight segments at any point of a connected set. This work is intended to be applied to estimate some tree ring features.

**Keywords:** Image processing, Segmentation, Ant colony algorithms, Neural networks, Wood quality, Discrete Geometry