



Ministère de l'Éducation Nationale

UNIVERSITÉ MONTPELLIER II  
- Faculté des Sciences -



## Master Intégration de Compétences

# RAPPORT DE STAGE

effectué à

L'Unité Mixte de Recherche « botanique et bioinformatique de  
l'Architecture des Plantes » AMAP.

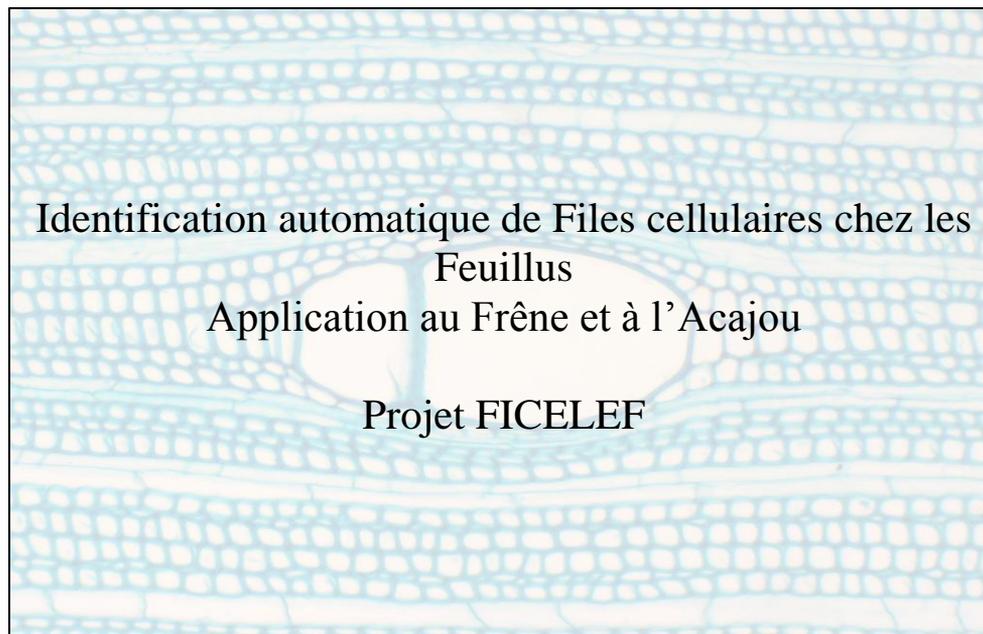
du 15 mars au 7 septembre 2010

par

Guilhem Brunel

Directeur du stage de l'entreprise : Philippe Borianne et Gérard Subsol

Directeur du stage de l'Université : Jean-François Pons



# REMERCIEMENTS

Je tiens tout d'abord à remercier mes tuteurs Philippe Borianne de l'AMAP et Gérard Subsol du LIRMM qui m'ont encadré tout au long de ce stage. Ils m'ont permis d'une part de découvrir la discipline du traitement d'image et d'autre part de me guider intuitivement, avec patience sur une bonne démarche de recherche.

Je voudrais aussi remercier Christine Heinz et Yves Caraglio pour m'avoir transmis les connaissances en architecture des plantes et en anatomie, nécessaires à la bonne compréhension du contexte biologique de ma mission.

Un grand merci aussi à Michaël Gueroult et Gille Caraux pour leurs disponibilités et leurs judicieux conseils lors de certains choix cruciaux à la bonne réussite de mon stage.

Mes remerciements vont aussi Jean François Pons, mon tuteur à l'université, pour m'avoir suivi tout au long de mon stage.

Mais aussi à Olivier Taugourdeau en thèse à l'AMAP pour ses conseils avisés en statistique, en biologie et pour sa bonne humeur quotidienne.

Je tiens enfin à remercier l'ensemble de l'UMR AMAP pour leur accueil dans la bonne humeur durant ces six mois.

# RESUME

Le projet **Ficelef** concerne la reconnaissance automatique de **Files Cellulaires** chez les **Feuillus**. Il a été le produit original d'application et développement informatique en traitement d'images, statistiques, mathématiques sur un sujet aujourd'hui indispensable à la compréhension de l'homme concernant le développement et l'architecture des plantes. Ces besoins ont été émis par l'unité mixte de recherche « *botAnique et bioinforMatique de l'Architecture des Plantes* » spécialiste dans ce domaine. Les résultats obtenus auront permis une approche nouvelle du problème déjà réfléchi dans le passé. Une chaîne de traitement aura été mise en place, fonctionnelle sous le logiciel libre de traitement d'images ImageJ. Le plugin fourni pourra cependant suivre autant d'améliorations techniques que d'innovations pouvant élargir sa portée au niveau des espèces et/ou structures traitées.

Mots-clés : *traitement d'image, file cellulaire, plugin, ImageJ, classification, détection automatique.*

# SUMMARY

The Ficelef project concerns the automatic recognition of cell lines among angiosperm. It was the original product of application and computer development in image processing, statistics, mathematics on a subject now necessary to the human understanding of plants' development and structure. These needs were brought to light by the mixed unit of research AMAP: an expert in the field. Acquired results had allowed a new approach on this matter already reflected in the past. A processing chain has been set up, functional under free image processing software: ImagesJ. The provided plugin will, however, be able to follow as much technical improvements as innovations that can enlarge his range on others tree species and/or treated structures.

Key word : *image processing, cell lines, plugin, ImageJ, classification, Autodiscovery.*

# SOMMAIRE

1	Introduction .....	1
1.1	Présentation de l'entreprise .....	2
1.2	Contexte Biologique .....	4
1.2.1	L'architecture et le développement des plantes.....	4
1.2.2	Le bois .....	4
1.2.3	Protocole d'acquisition.....	9
1.3	sujet de stage.....	11
1.4	Planning prévisionnel .....	12
2	Matériels et Méthodes.....	13
2.1	Matériels.....	13
2.1.1	Environnement de travail.....	13
2.1.2	Supports numériques .....	15
2.2	Pré-étude.....	16
2.2.1	Etat de l'art.....	16
2.2.2	Nouvelle étude prospective sur la segmentation .....	18
2.3	Méthode mise en œuvre .....	25
2.3.1	Segmentation .....	27
2.3.2	Detection des files : Caractérisation.....	36
2.3.3	Typage des cellules.....	47
3	Résultats.....	50
3.1	Intégration sous ImageJ.....	50
3.2	Résultats obtenus .....	52
3.2.1	Résultats.....	52
3.3	Discussion .....	55

3.4	Perspectives.....	57
3.4.1	Perspectives à court terme .....	57
3.4.2	Perspectives à long terme .....	57
4	Conclusion.....	58

## 1 INTRODUCTION

Ce rapport est consacré à la présentation du travail effectué dans le cadre de mon stage de fin d'étude en Master BioInformatique (Université Montpellier II) au sein de l'UMR botAnique et bioinforMatique de l'Architecture des Plantes (AMAP) du 15 mars au 7 septembre 2010.

Pour la bonne compréhension du stage à réaliser, on posera explicitement le contexte d'étude proposé, aussi bien à un niveau organisationnel que scientifique, nécessaire à la bonne compréhension du travail à réaliser.

Le sujet traite d'un travail de reconnaissance automatique de file cellulaire sur des images microscopiques de bois chez les feuillus. Ce travail est à l'heure actuelle réalisé de façon manuelle par les botanistes et par les anatomistes, ce qui est aussi bien long et fastidieux, que limitant quant à la quantité de données récupérées. Lors du précédent stage un travail d'automatisation des files cellulaires avait été mis en place pour les résineux.

L'automatisation et la standardisation de ce protocole seraient donc un gain important de temps mais aussi une perspective nouvelle quant à l'étude statistique de l'organisation des files cellulaires. Cette problématique a déjà fait l'étude de travaux de chercheurs sans obtenir de résultats réellement satisfaisants, ce qui démontre la réelle difficulté du problème.

On peut diviser le sujet en plusieurs étapes successives : amélioration de la lisibilité de l'image, segmentation de l'image, reconnaissance des files et caractérisation des régions extraites. On aura orienté volontairement, en accord avec les tuteurs encadrant le stage, une méthode de travail prospective. En effet le sujet étant large et très peu étudié dans la littérature, le critère de temps rentre considérablement en jeu. Finalement, à l'issue du stage, la solution une solution non optimale mais couvrant tous les aspects du travail sera adoptée, plutôt qu'une solution partielle plus robuste.

Toutes les méthodes existantes à la résolution de chaque étape seront testées et discutées. La solution proposée pourra être reprise et sera donc largement améliorable, mais proposera d'ores et déjà une trame complète et opérationnelle.

La chaîne de traitement proposée présente une façon originale de procéder et fait appel à des domaines très divers tels que le traitement d'images, la théorie des graphes, la modélisation géométrique, la bio statistique ou encore la création d'arbres de décisions.

## 1.1 PRESENTATION DE L'ENTREPRISE

Mon stage s'est déroulé au sein de l'UMR (Unité Mixte de Recherche) AMAP (botanique et bioinformatique de l'Architecture des Plantes), installée dans les locaux montpelliérains du CIRAD (Centre de Coopération Internationale de Recherche Agronomique pour le Développement).

Le CIRAD est l'institut français de recherche agronomique au service du développement des pays du Sud et de l'outre-mer français. Il privilégie la recherche en partenariat et a choisi le développement durable comme ligne de force de son action à travers le monde. Cette démarche prend en compte les conséquences écologiques, économiques et sociales, à long terme, des processus de transformation des sociétés et des territoires du Sud. Il intervient par des recherches et expérimentations, des actions de formation, d'information et d'innovation, et des expertises. Ses compétences relèvent des sciences du vivant, des sciences humaines et des sciences de l'ingénieur, appliquées à l'agriculture et l'alimentation, à la gestion des ressources naturelles et aux sociétés.

C'est dans ce contexte que s'inscrit l'unité mixte de recherche AMAP. Ses recherches accordent une place centrale à la description, l'analyse et la modélisation de la structure, de la dynamique de développement et de la diversité des plantes et des peuplements végétaux.

L'unité privilégie les approches génériques, communes à l'ensemble des plantes, annuelles ou pérennes, actuelles ou fossiles, sauvages ou cultivées, tempérées, méditerranéennes ou tropicales. Ces approches concernent la mesure, la représentation, l'analyse et le traitement des données, la simulation informatique, l'organisation et la gestion des connaissances. Le Cirad est un EPIC, c'est-à-dire un Etablissement Public à Caractère Economique et Commercial. Son activité est donc tournée vers le « service public »

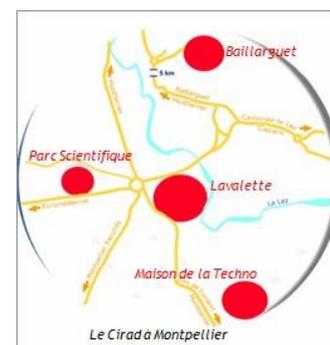
Organisme de recherche finalisée au service des pays du Sud, le Cirad intervient dans plus de 50 pays du monde. Tout au long de son histoire, il a noué et entretenu des liens étroits avec les sociétés humaines, les lieux de production agricole, animale et forestière, les milieux physiques et biologiques des pays tropicaux et subtropicaux.



L'outre-mer française représente une part importante de l'activité du Cirad. Avec plus de 500 personnes, dont plus de 100 chercheurs, le Cirad développe, avec ses partenaires, un potentiel de recherche agronomique qui constitue les fondements du futur réseau européen d'agronomie tropicale.

Paris : le siège social est à Paris pour faciliter les rapports avec le ministère de la Recherche et les partenaires de passage par la Capitale.

Montpellier : le Cirad s'étend sur 4 sites géographiques aux abords de Montferrier, ce qui représente une trentaine de bâtiments, regroupant environ 800 chercheurs.



L'organisation du Cirad est en mutation permanente. Elle est passée en 2007 de 9 à 3 départements de recherche comme le montre le schéma en annexe 1.

L'UMR AMAP est une unité multidisciplinaire : elle regroupe des disciplines aussi variées que la biologie, la botanique, l'écologie, les mathématiques appliquées et l'informatique, dont l'association constitue aujourd'hui des enjeux scientifiques et techniques majeurs pour l'agronomie et la foresterie modernes.

Elle est composée de 52 agents permanents (dont 40 chercheurs ou enseignants chercheurs et 2 chercheurs associés) dont 21 agents CIRAD répartis en Chine, en Guyane Française, au Laos, à l'INRIA de Rocquencourt et à

Montpellier, 8 agents CNRS, 8 agents INRA, 10 agents IRD (dont 3 en Guyane française), 3 enseignants chercheurs de l'Université Montpellier II et 2 chercheurs détachés au MAE en Inde, plus une quinzaine de doctorants et une quarantaine de stagiaires par an.

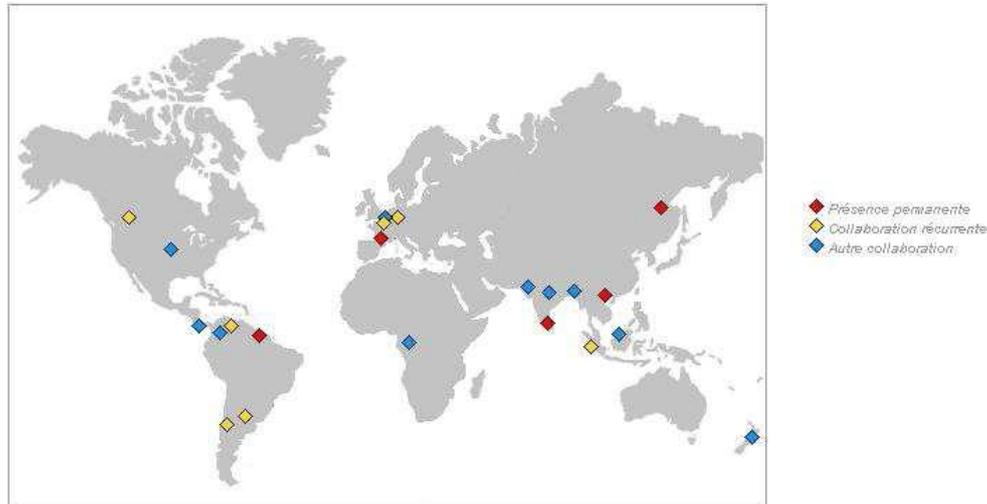


Figure 1. Localisation de l'AMAP dans le monde

Les activités de l'UMR dans le domaine des mathématiques et de l'informatique appliquées s'articulent autour des trois axes :

- la modélisation et la simulation de la croissance et de l'architecture des plantes
- l'analyse et le traitement statistique des informations
- l'ingénierie logicielle

Les recherches accordent une place centrale à la description, l'analyse et la modélisation de la structure, de la dynamique de développement et de la diversité des plantes et des peuplements végétaux. Elles privilégient les approches génériques, communes à l'ensemble des plantes, annuelles ou pérennes, actuelles ou fossiles, sauvages ou cultivées, tempérées, méditerranéennes ou tropicales. Elles concernent la mesure, la représentation, l'analyse et le traitement des données, la simulation informatique, l'organisation et la gestion des connaissances.

L'UMR contribue aussi :

- à la formation initiale (cours et encadrement dans les Universités et écoles d'ingénieur) et professionnelle ;
- à la diffusion des connaissances et recherches sous forme de logiciels le plus souvent libres mais parfois commercialisés ;
- à la coopération scientifique internationale, notamment avec des pays du sud.

L'UMR entretient des liens étroits avec des équipes de plusieurs organismes nationaux et développe un réseau diversifié de partenariats, académiques, scientifiques, technologiques et commerciaux, non seulement en Europe mais aussi en Amérique du Nord, en Argentine, au Chili, en Chine, en Inde, au Laos et en Nouvelle-Zélande.

Les problématiques de recherche abordées dans les différentes équipes proviennent de questions biologiques. Afin de répondre à ces questions, il est parfois nécessaire de créer de nouveaux outils mathématiques et/ou informatiques. Dans certains cas, ces développements sont eux-mêmes porteurs d'une problématique scientifique à part entière. Les activités dans le cadre du domaine MIA<sup>1</sup> concernent la formalisation mathématique, la modélisation ou la production logicielle. C'est dans ce même cadre que s'inscrit le projet Ficelef en réponse à une problématique concernant l'équipe d'architecture des plantes.

<sup>1</sup> Mathématiques et Informatique Appliqué.

## 1.2 CONTEXTE BIOLOGIQUE

### 1.2.1 L'ARCHITECTURE ET LE DEVELOPPEMENT DES PLANTES

Le but de cette discipline est d'analyser, de comprendre et de prédire les modes d'édification et de développement architectural des plantes, leur plasticité, leur production et d'identifier le rôle respectif des facteurs génétiques, environnementaux et culturels au cours de leur croissance et de leur ontogénie. A cette fin, différents axes d'études sont posés :

- Définir et mettre en œuvre, à partir de descripteurs essentiellement morphologiques et anatomiques, les concepts et les méthodes nécessaires à la description, à la mesure, à l'analyse et à la compréhension de l'architecture des plantes.
- Evaluer les composantes de l'expression phénotypique et génétique de l'architecture des plantes en analysant et en caractérisant leur diversité structurale et leur plasticité phénotypique.
- Analyser l'édification de l'architecture des plantes en fonction de processus physiologiques pour comprendre les mécanismes d'élaboration de leur production.
- Développer des outils de modélisation et de simulation intégrant des processus physiques, physiologiques et morphogénétiques pour prévoir le développement architectural des plantes dans des conditions agronomiques, écologiques et climatiques variées.

L'originalité de ces recherches résulte d'une approche spatio-temporelle (i.e. globale et dynamique), multi-échelles et multidisciplinaire de la structure végétale. Les méthodes développées et les résultats obtenus fournissent, aux autres domaines de la biologie végétale, des connaissances sur la plante entière, des protocoles d'observation et d'échantillonnage de leur structure et des modèles d'analyse et de représentation de l'édification de l'architecture des plantes.

### 1.2.2 LE BOIS

C'est dans ce cadre que s'inscrit l'étude du bois. Celui-ci ayant plusieurs rôles dans l'architecture des plantes :

- Assure le grossissement en diamètre des tiges
- Permet l'alimentation et le stockage dans la plante
- Permet la rigidité des tiges
- Son élaboration conditionne :
  - La production de biomasse
  - La production de matière première
  - Les caractéristiques biomécaniques
  - Les propriétés physiques du matériau (technologie du bois)

Plusieurs approches sont utilisées quant à son observation :

L'étude cambiale (du cambium, cf.1.2.2.1) consiste à connaître l'accroissement en périmètre du bois (analyse circulaire), à connaître la nature de la régularité des types cellulaires, à identifier comment s'organise la nature des nouvelles cellules (initiales fusiformes : vaisseaux, fibres, parenchyme, initiales de rayon : parenchyme). Mais aussi de savoir comment son fonctionnement change au cours des saisons, de moments de l'année. Lors de la production d'une initiale fusiforme, les biologistes réalisent une analyse radiale pour observer comment se succèdent les différents types cellulaires. On s'intéresse aussi à la proportion de tissu vasculaire (bois et liber) entre les fibres (soutien) et les rayons (réserves), mais aussi entre la taille des éléments vaisseaux (lumière) et fibres (parois). Etudier les cernes permet de caractériser la régularité d'accroissement en diamètre du tronc. On cherche aussi à caractériser le lien allongement/épaississement.

Plus généralement, l'étude de l'architecture bois cherche à modéliser l'organisation cellulaire :

- Détecter les lignées cellulaires
- Détourer les tissus
- Caractériser (dimension)
- Reconstruction 3D

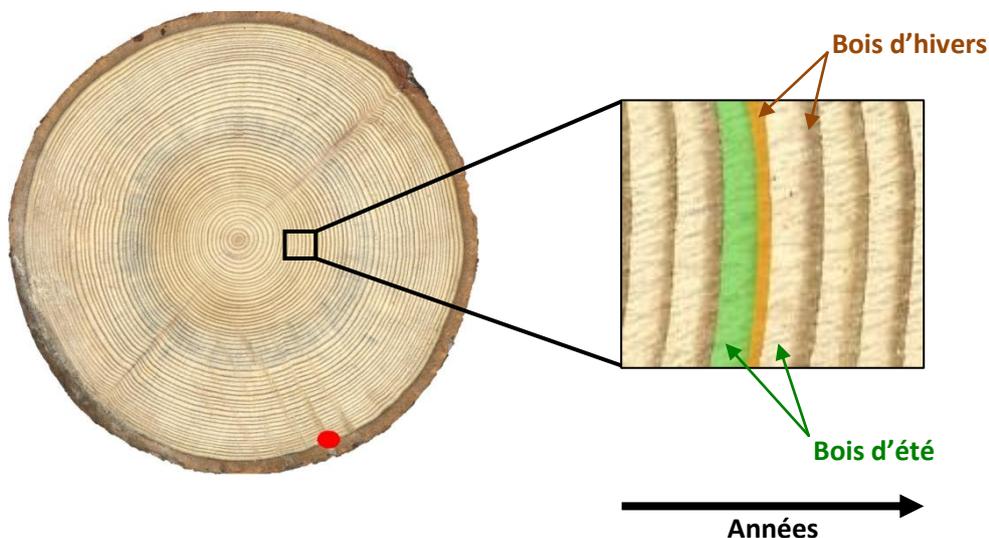
Dans le cadre de la réflexion sur le fonctionnement cambial, la production et la différenciation des cellules issues des initiales radiales et fusiformes sont le reflet des variations du contexte de croissance tant du point de vue interne (régulation des flux et échange liber/bois) qu'externe (modification de l'environnement lumineux et hydrique). Pouvoir détecter et dénombrer les cellules permet d'accéder à la mise en évidence de régularités dans la succession des types cellulaires (tant pour les initiales -agrandissement du cambium-, que pour les cellules produites -différenciation cellulaire-). Ce type de questionnement nécessite de pouvoir accéder à de l'échantillonnage suffisant pour manipuler des outils et modèles statistiques. Le côté fastidieux du comptage et typage cellulaires pourrait être allégé par une délimitation et identification (semi-)automatique des cellules.

Dans le cadre précédemment posé sur l'architecture du bois, les spécialistes en biologie végétale analysent la taille et la disposition des cellules du bois à partir d'images microscopiques de sections de troncs ou de branches. Les files cellulaires sont définies par les alignements « naturels » de cellules, facilement identifiables à l'œil nu sur l'image numérique. Il s'agit en fait de retrouver l'organisation spatiale des cellules, c'est-à-dire les motifs dominants qui se dessinent à partir des seules informations géométriques.

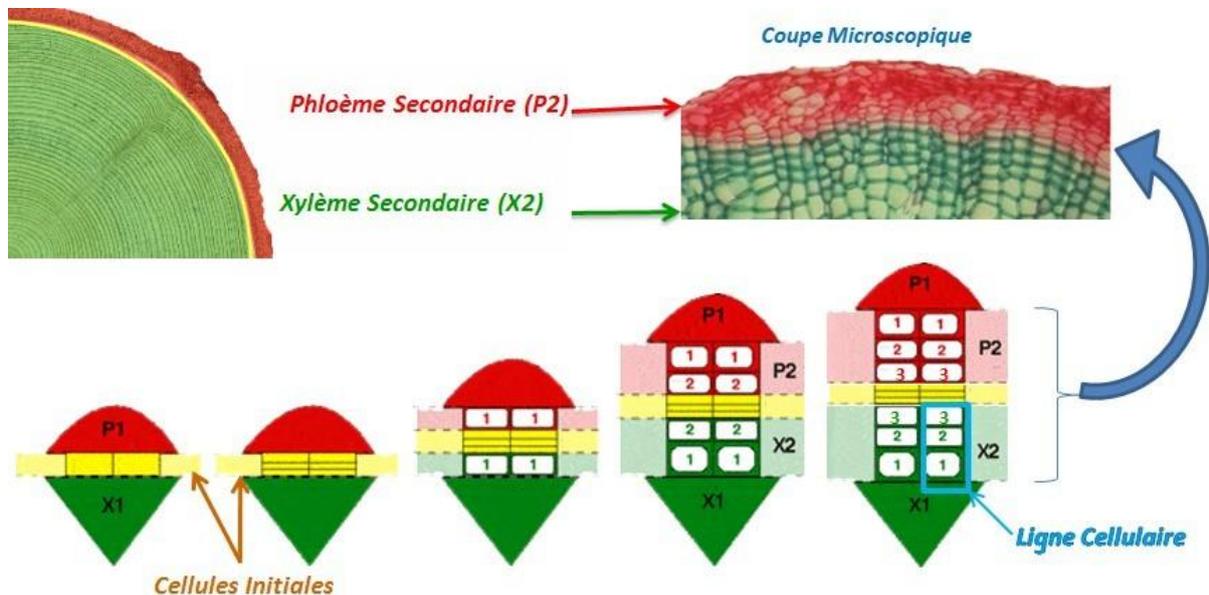
Pour bien interpréter le contenu des images sur lesquelles on travaillera ainsi que le rôle des files cellulaires, il est nécessaire de bien comprendre la formation et la structuration du bois ainsi que le protocole d'acquisition de ces images.

### 1.2.2.1 FORMATION DU BOIS

Le développement de l'arbre résulte d'une elongation et d'une croissance en diamètre, correspondant chacune à une organisation différente des tissus. Les croissances apicale (en hauteur) et radiale (en diamètre) se traduisent par une accumulation géométrique des cernes annuels d'accroissement, plutôt analogue à un empilement de cônes. Chez les espèces ligneuses et pérennes (arbres, arbustes, buissons), le fonctionnement du cambium suit un cycle saisonnier. En région tempérée, le fonctionnement s'interrompt à l'automne et reprend au printemps. Chaque année, un nouveau cylindre de bois est formé à l'extérieur du précédent.



Nous allons maintenant décrire les différents termes et concepts nécessaires à la bonne compréhension pour l'étude du bois. Nous allons illustrés ce vocabulaire grâce au schéma ci-dessous et ensuite nous en ferons l'explication.



❖ Cambium (jaune) :

Zone de cellules peu différenciées à divisions actives qui participent à l'accroissement en diamètre de la tige en se différenciant en xylème secondaire (bois) vers le centre (proportion 2x) ou en phloème secondaire, ou liber, vers l'extérieur (proportion 1x). Il est composé d'initiales fusiformes (qui se différencient en vaisseaux, fibres, parenchyme) et d'initiales de rayon (qui se différencient parenchyme).

❖ Liber ou phloème (rouge) :

Conduit la sève élaborée (eau et sucres formés dans les feuilles par la photosynthèse) vers toutes les régions de la plante.

❖ Bois ou xylème (vert) :

Conduit verticalement la sève brute (eaux et éléments minéraux en provenance des racines) du sol vers les feuilles.

Grâce au schéma précédent nous pouvons voir l'importance de l'étude des files cellulaires, car elles rendent compte directement de l'évolution du bois et donc des corrélations entre la croissance et les conditions climatiques. Une bonne compréhension des files pourraient permettre de créer des modèles selon les facteurs climatiques et ainsi simuler une croissance des cellules.

Nous allons maintenant décrire les structure microscopiques rencontrées chez les feuillus.

### 1.2.2.2 STRUCTURE DU BOIS

Voici les éléments constitutifs du bois et leur agencement spatial :

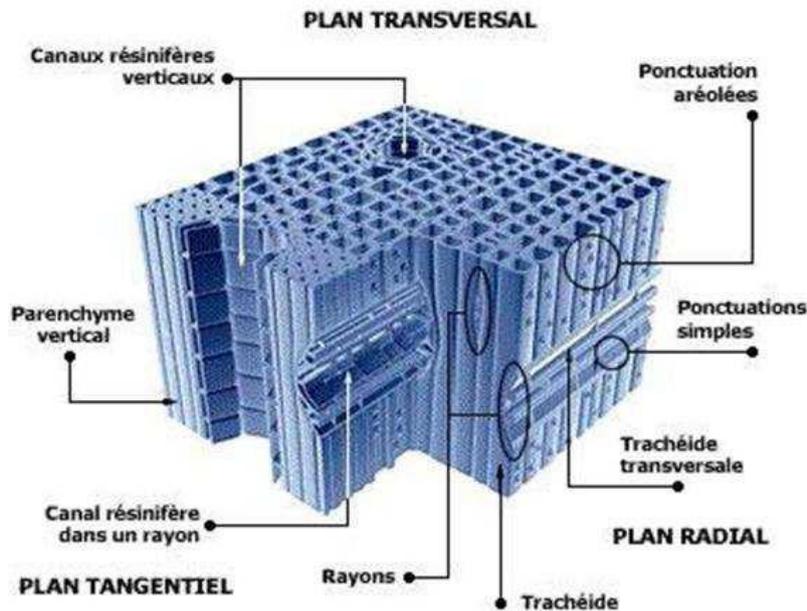


Figure 2. Eléments structurants chez le résineux

**Remarque :** Entre les résineux et les feuillus la différence anatomique principale provient de la présence de vaisseaux (grosses cellules) dans le bois. Les résineux n'en possèdent pas, la fonction de maintien et de conductance est assurée par les trachéides. Dans les feuillus, les vaisseaux conduisent et les fibres maintiennent : il y a spécialisation des cellules. Nous verrons sur des coupes 2D l'anatomie des vaisseaux.

#### **Définition :**

- ❖ **Les fibres :** (en jaune, figure 3)

Les fibres simples ou cloisonnées jouent un rôle de soutien. On désigne sous le nom de fibres, des paquets des faisceaux de fibres. Les faisceaux sont déviés par les rayons médullaires si ces rayons sont épais. Il en résulte des madures, des ondulations, particulièrement marquées dans les coupes. Les fibres ont des sections polygonales. Leurs parois sont plus ou moins épaissies suivant l'espèce et suivant l'âge de la partie considérée. Elles sont lignifiées et leur cavité peut être simple ou cloisonnée.

- ❖ **Les vaisseaux :** (en vert, figure 3)

Les parois des vaisseaux à quelques exceptions près (le frêne par exemple) sont plus minces que les parois des fibres. Elles sont encore plus faibles au niveau des ponctuations par lesquelles s'effectuent les passages osmotiques. Les vaisseaux influent par leurs dimensions et leurs distributions sur les propriétés du bois.

- ❖ **Les parenchymes :**

Court ou long, le parenchyme est formé de cellules aux parois peu épaissies. Le parenchyme est formé de cellules polyédriques à peu près de même diamètre. Dans le bois jeune, il sert de magasin de réserve à l'amidon. Dans le bois âgé, ses cellules s'épaississent et, comme les fibres, il devient un tissu de soutien. On le trouve autour des vaisseaux. Les parenchymes constituent également avec une orientation radiale, les rayons médullaires.

❖ **Les rayons** : (en rouge, figure 3)

Les rayons sont des cellules de réserve (parenchymateuses) à parois épaissies et lignifiées, qui accompagnent le tissu vasculaire. Ces cellules participent en outre à la fonction de soutien. Leur orientation est transversale et rayonnante en partant de l'axe longitudinal de l'arbre.

❖ **Les trachéides** :

Les trachéides sont des cellules allongées du xylème des plantes vasculaires servant au transport de la sève brute (composée d'eau et de sels minéraux). La formation de trachéides varie en fonction de l'endroit où ils se produisent. Ce sont de longues cellules étroites, dont les extrémités sont en biseau. Elles possèdent une paroi primaire et secondaire lignifiées et sont mortes à maturité.

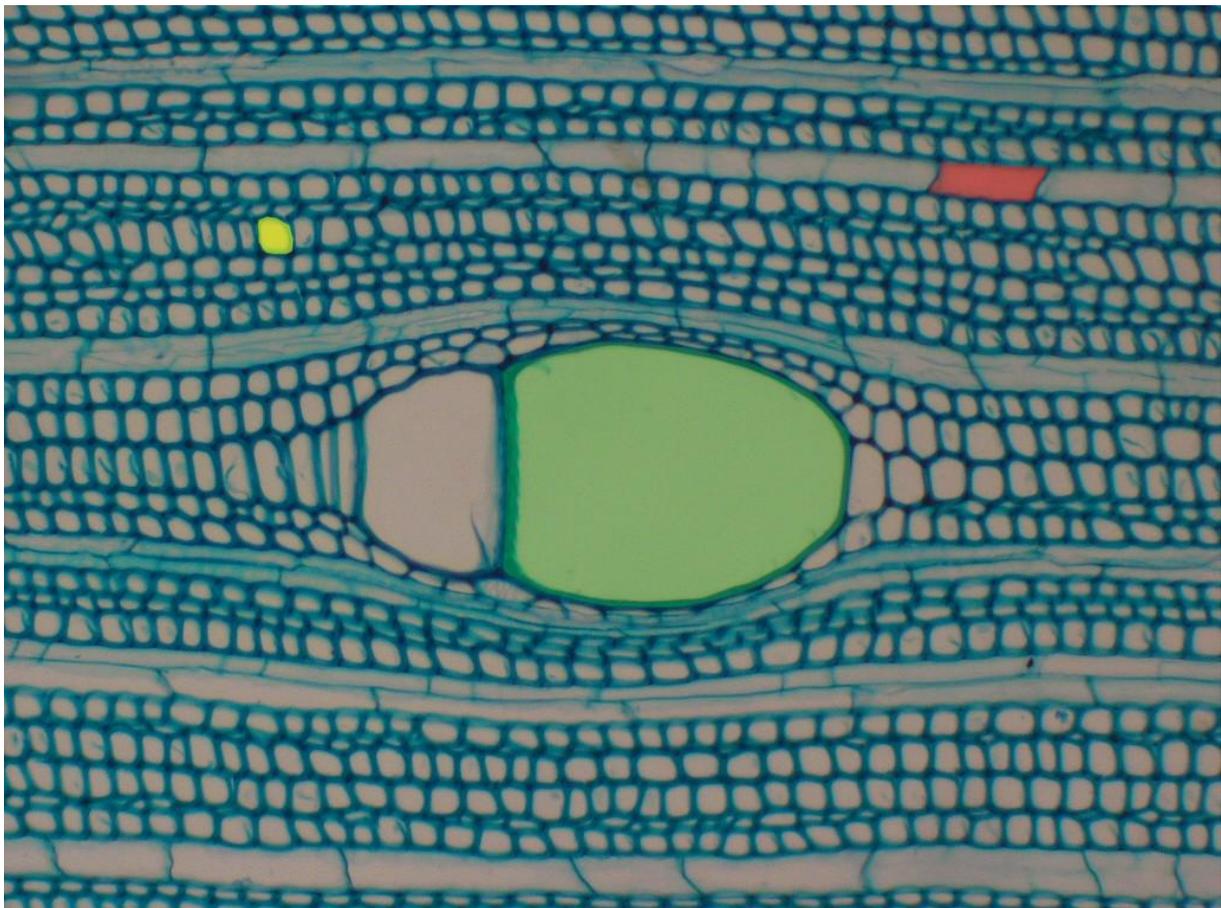


Figure 3. Coupe Transversale d'Acajou

Remarque : Lors de notre étude, nous nous intéresserons principalement à trois types cellulaires ; Les fibres, les rayons et les vaisseaux. En effet il faudrait disposer de la coupe tangentielle pour différencier les autres types de cellules.

Il nous reste à décrire ce que nous allons appeler cellule dans le reste du rapport. Une cellule est constituée d'une paroi (en cyan) et d'un lumen (en rose). Le Lumen ou la lumière d'une cellule est son espace intérieur. Ici les lumens sont vidés de leur contenu biologique.

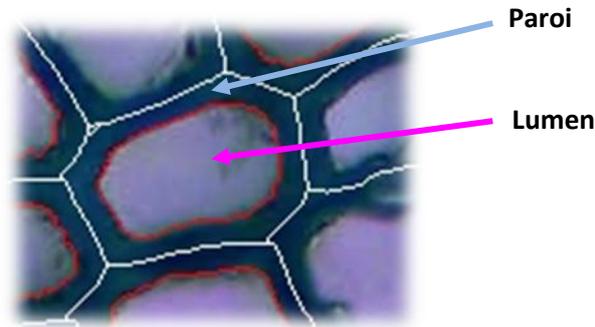


Figure 4. Cellule d'acajou

Les parois des deux cellules adjacentes sont séparées par la lamelle moyenne. Cette lamelle est très fine, nous ne pourrions pas l'observer à l'échelle de nos images.

### 1.2.3 PROTOCOLE D'ACQUISITION

L'étude des lignées cellulaires s'effectue sur des images microscopiques tirées de troncs ou branches coupés selon le plan transverse, rendant ainsi possible le suivi radial du cambium à la moelle.

Il est important de savoir comment de telles images sont obtenues. Le traitement des images est directement lié au type de préparation, par exemple le type de la coloration appliquée. Exemple d'un protocole en annexe 7.

On peut découper en trois étapes successives l'obtention d'images :

- L'échantillonnage.
- La préparation
- L'acquisition

A chaque niveau des particularités apparaissent, dépendantes de l'opérateur, de ses motivations et de ses moyens. En l'absence de protocoles très rigoureux (i.e. très automatisés), chaque étape doit être menée en connaissance de la finalité : on ne produit pas avec la même « application » des images destinées à un comptage manuel de cellules et des images utilisées pour une estimation automatique de parois cellulaires. Tout opérateur de la chaîne doit donc directement être impliqué dans le processus d'analyse et de quantification, c'est-à-dire savoir exactement comment seront exploitées les données qu'il produit ; dans le cas contraire, les données risquent de ne pouvoir être exploitées de manière optimale. A cette contrainte forte s'ajoute l'effet opérateur : la majorité des étapes du protocole d'acquisition sont laissées à l'appréciation « subjective » des opérateurs : qu'il s'agisse du prélèvement du « bon échantillon », de l'appréciation de la planéité d'une coupe après ponçage, de la mesure des temps d'immersion, de la pression exercée sur les lames ou le choix des grossissements ou des zones d'observation, tout est sujet à caution.

#### 1.2.3.1 L'ECHANTILLONNAGE

Il s'agit de rondelles obtenues par section transversale de branches ou tronc. Avant d'en extraire une coupe à observer, l'angle de section est ajusté par ponçage perpendiculairement à l'axe vertical de la branche/tronc. Les rondelles sont ensuite échantillonnées de deux façons différentes : par découpe ou par ponçage et ne suivent pas le même conditionnement.

### 1.2.3.1.1 PREPARATION

---

#### ❖ En ce qui concerne la découpe :

La rondelle trempe au préalable dans l'eau. S'il s'agit d'une rondelle de tronc, le préparateur sectionne des cubes au ciseau. S'il s'agit d'une tige (quelques centimètres au maximum) on garde la section entière. La découpe se fait grâce à un microtome<sup>2</sup> (*Image ci contre*) de calibre 15-30µm. Les contenus cellulaires sont tués à l'eau de javel (rayons vidés). Le bois est déshydraté à l'alcool. Puis vient la coloration : plusieurs types de colorants sont appliqués selon ce que l'on souhaite observer (par exemple : safranine pour les parois, carmino-vert de Mirande pour, en vert la lignine et en rouge la cellulose ; il existe aussi des colorants pour identifier contenu cellulaire...). Une combinaison de ceux-ci est possible. Pour notre étude, seule la safranine a été utilisée. Les échantillons obtenus sont fixés sur une lame de verre à la glycérine, placée sous microscope et nécessitant un éclairage par-dessous.



#### ❖ En ce qui concerne le ponçage :

Une rondelle d'environ 5mm est prélevée, placée sur de la patte à modeler, puis poncée jusqu'à l'obtention d'une section de quelques µm. Directement placée sous microscope, cette préparation ne comporte pas de coloration et nécessite un éclairage du dessus.

### 1.2.3.1.2 ACQUISITION

---

Les échantillons sont finalement photographiés par l'intermédiaire d'une caméra numérique (*Olympus DP71*) montée sur microscope. Plusieurs grossissements sont utilisés, pour observer un cycle annuel de la croissance, ou plus en détail dans un cycle, ou un ensemble de cycles. L'image est capturée par un logiciel d'acquisition (*Olympus cell^A*) où l'opérateur peut y ajuster la couleur, le contraste, la balance couleur, etc., mais aussi choisir la résolution, le format d'export.



La diversité des méthodes de préparations et d'acquisitions, ajoutée à la diversité biologique, nous donne un jeu d'images avec une très forte variabilité à tous les niveaux. Nous présenterons plus en détail cette variabilité dans le paragraphe 2.1.2 *Supports numériques*.

---

<sup>2</sup> Microtome : appareil utilisé pour produire des rubans de coupe de très faible épaisseur.

### 1.3 SUJET DE STAGE

Le projet **FICLEF** (pour **Files cellulaires** dans les **Feuillus**), nom de mon sujet de stage, a pour but de travailler sur des images de coupes transverses de troncs ou branches chez les Feuillus.

Afin de comprendre le développement d'un arbre, les spécialistes en biologie végétale analysent la taille et la disposition des cellules du bois à partir d'images microscopiques de sections de tronc ou de branches.

Les files cellulaires sont définies par les alignements « naturels » de cellules, facilement identifiables à l'œil sur l'image numérique. Il s'agit en fait de retrouver l'organisation spatiale des cellules, c'est-à-dire les motifs dominants qui se dessinent à partir des seules informations géométriques. Actuellement, la délimitation et l'identification des cellules et des files cellulaires s'effectuent manuellement. Or, pour bien comprendre le développement d'une espèce d'arbre, il faut recueillir des observations sur un très grand nombre d'échantillon. Il devient donc indispensable d'automatiser cette reconnaissance par des méthodes de traitement d'images.

Du fait de l'hétérogénéité multiple des images, la segmentation des cellules des fibres, des vaisseaux pose un réel problème. L'idéal serait d'avoir un processus de segmentation auto paramétrable, donnant un résultat satisfaisant sur l'intégralité des images.

Une méthode a déjà été développée pour l'extraction de files chez les résineux, leur structure est la plus simple parmi les arbres (en opposition au feuillus).

Toutefois, la méthode, actuellement implémentée, ne permet pas de correctement traiter les images de feuillus (image ci-dessous) du fait de la présence de vaisseaux (grosses cellules) qui déforment, voire interrompent les files.

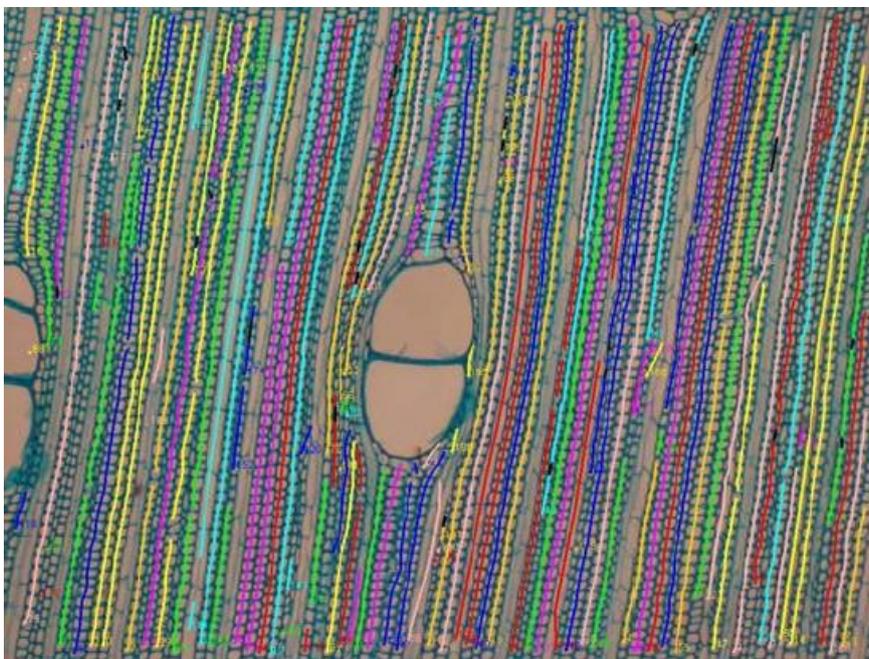


Figure 5. Détection des files cellulaires sur une coupe d'Acajou

#### Le stage permettra :

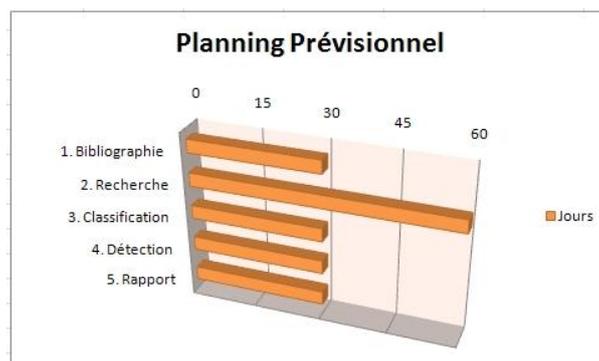
- De recenser les différents problèmes et solutions potentiellement intéressants.
- De mettre en place une solution opérationnelle en Java, sous forme de plugin dans le logiciel libre ImageJ.

## 1.4 PLANNING PREVISIONNEL

Durant les premières semaines du stage, mes tuteurs et moi-même avons mis en place des objectifs à atteindre durant une période de temps donnée. Le suivi de l'avancement du projet a été fait lors de réunions régulières hebdomadaires voire mensuelles avec les tuteurs pour discuter des avancés significatives du travail et des problèmes rencontrés, pour évaluer ensemble les options possibles, fixer les priorités ou acter des choix forts. J'ai aussi eu l'occasion de faire des présentations orales tout au long de mon stage au LIRMM<sup>3</sup> devant l'équipe d'imagerie du laboratoire (équipe ICAR<sup>4</sup>), ainsi qu'une pré-soutenance à l'occasion de ma candidature au sujet de thèse de M<sup>me</sup> Anne-Muriel Arigon Chifolleau sur l'étude phylogénétique du VIH<sup>5</sup>, en juin.

Le planning prévisionnel prévoyait cinq grands axes :

- Mars : découverte du sujet et appréhension du contexte biologique, comprenant une étude bibliographique et du stage précédent, des rencontres et des discussions avec les acteurs principalement concernés par le sujet, en particulier deux botanistes et un thésard en botanique et architecture des plantes; j'ai également consacré cette période à approfondir mes connaissances en traitement d'images, à maîtriser mon environnement de travail et à mener quelques tests.
- Avril et Mai : tester des nouvelles méthodes de prétraitement de l'image (à leurs implémentations), de les comparer aux méthodes existante. Rechercher de nouveaux algorithmes de segmentation. Et pour finir, la mise en place d'un graphe d'adjacence supportant les données extraites.
- Juin : création d'une méthode de classification des cellules, selon deux axes de recherche : une formalisation d'un arbre de décision avec les biologistes, ou l'utilisation de méthode issu du datamining.
- Juillet : Mise en place d'un algorithme de parcours de graphe pour la détection des files cellulaires.
- Aout : formalisation du rapport de stage en vue d'un dépôt fin aout ; les quinze derniers jours seront consacrés à la préparation de la présentation orale ainsi qu'une pré-soutenance de stage et à la préparation de l'après stage.



<sup>3</sup> Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

<sup>4</sup> ICAR : descriptif de l'équipe et de leurs recherches, <http://www.lirmm.fr/icar/index.html>

<sup>5</sup> VIH : virus de l'immunodéficience Humaine

## 2 MATERIELS ET METHODES

### 2.1 MATERIELS

#### 2.1.1 ENVIRONNEMENT DE TRAVAIL

Le sujet d'étude sur la détection des files cellulaires impose la création d'un plugin à intégrer au sein d'un projet intégré au logiciel libre de traitement d'image ImageJ (<http://rsbweb.nih.gov/ij/>) [15], le tout devant être implémenté en java. Ce logiciel est communément utilisé par la communauté scientifique car il possède un grand nombre de plugins en open source, ce qui permet de tester et utiliser facilement des outils mis en place pour la communauté d'utilisateurs.

Extrait de l'article consacré à ImageJ sur Wikipedia <sup>6</sup>:

« ImageJ est un logiciel de traitement et d'analyse d'images écrit en Java, ce qui en fait un logiciel utilisable sur différents systèmes d'exploitation. [...] La plupart des opérations courantes de traitement d'images sont réalisables avec ImageJ. [...] L'ajout personnalisé de fonctions est possible grâce aux plugins à écrire en java. [...] ImageJ a été initialement développé pour des applications biomédicales. Il permet par exemple de faire des analyses de gels d'électrophorèse, ou de la détection et analyse de tumeurs. Son usage s'est depuis étendu à d'autres domaines, comme la science des matériaux (détermination de tailles de grains, traitement d'images obtenues par microtomographie X par exemple). »

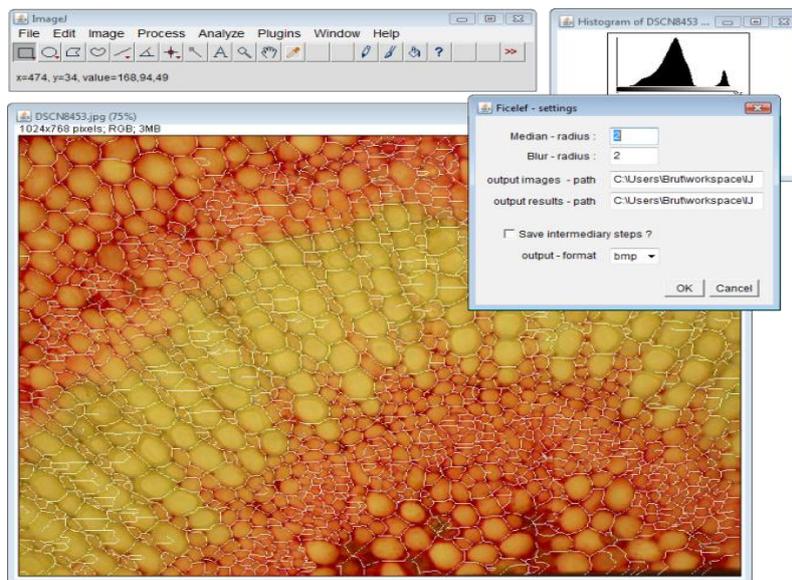


Figure 6. Interface d'ImageJ

De plus, mes tuteurs ont souhaité utiliser un gestionnaire de projet collaboratif nommé Redmine<sup>7</sup> open source permettant à tous de partager documents et codes sources centralisés sur les serveurs de l'AMAP. Ainsi le projet Ficelef est accessible à l'adresse suivant : <http://amap-dev.cirad.fr/projects/ficelef>. On y trouve un wiki, un gestionnaire de document, un SVN supportant les sources du plugin régulièrement mis à jour, etc.

<sup>6</sup> Wikipedia : encyclopédie libre, <http://fr.wikipedia.org/>

<sup>7</sup> Redmine est un système Open Source de gestion complète de projet en mode web, développé en Ruby sur la base du framework Ruby on Rails : <http://www.redmine.org/>



Figure 7. Interface du Redmine

J'ai choisi de tenir un cahier informatique et non un cahier de labo. Ainsi chaque protocole d'étude ou résultat obtenu ont fait l'objet d'un compte-rendu précisant les différents paramètres utilisés ainsi que les conclusions issues des tests. Cette méthode de travail permet de partager plus facilement les résultats avec mes encadrants et de pouvoir modifier les documents au fur et à mesure de l'avancement des tests.

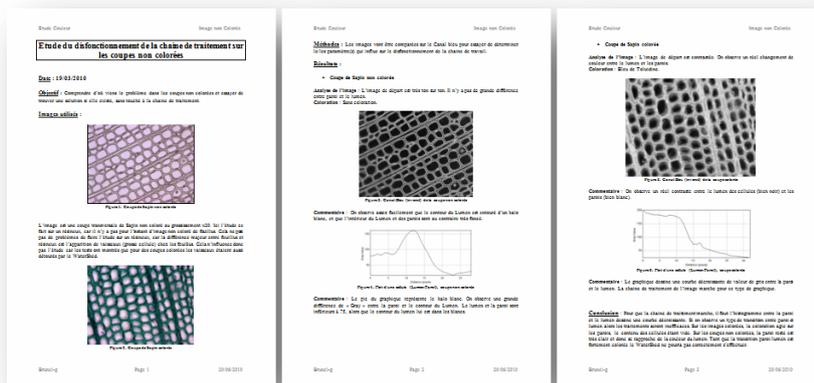


Figure 8. Compte rendu "Image Non Colorée"

Le développement du plugin à proprement parler s'est réalisé sous l'environnement de développement Java Eclipse<sup>8</sup> et à l'aide de nombreuses bibliothèques permettant le déploiement du plugin automatiquement (Ant<sup>9</sup>) ou de l'utilisation du SVN (Subclipse<sup>10</sup>), etc.

<sup>8</sup> Eclipse : IDE Open Source, essentiellement dédié au langage Java. <http://www.eclipse.org/>

<sup>9</sup> Ant : « Ant est un projet open source [...] qui vise le développement d'un logiciel d'automatisation des opérations répétitives tout au long du cycle de développement logiciel » (Wikipedia) ; <http://ant.apache.org/>

<sup>10</sup> Subclipse : Système de gestion de versions : <http://subversion.tigris.org>

### 2.1.2 SUPPORTS NUMERIQUES

Le travail à réaliser a pour support un ensemble d'images numériques. Les images à traiter proviennent essentiellement du labo d'accueil ; elles sont issues d'un protocole par découpe et coloration.

Rappelons tout d'abord la notion d'image numérique. C'est une matrice rectangle de points, chaque point correspondant classiquement au centre géométrique d'un élément de surface appelé Pixel (*Picture element*) ; par abus de langage, on appelle Pixel indifféremment le point ou l'élément de surface qui lui est associé. Pour notre étude, la plupart des images ont une résolution entre 1024\*768 et 2048\*1536 pixels en couleur. L'acquisition est faite par des microscopes électroniques qui ont un grossissement de x100 à x400.

Les images présentent une très grande variabilité due à différents facteurs.

- La préparation : selon ce que veut observer le biologiste et selon le protocole d'étude, la coloration des tissus biologiques sera différentes.
- Le grossissement : il dépend du champ d'étude que veut obtenir le biologiste. Pour observer un petit élément structurant le champ pourra être faible pour pouvoir le comparer avec d'autres éléments, ou très fort pour le voir en détail.
- L'essence : selon les différentes essences d'arbre, l'organisation cellulaire est différente. On peut ainsi différencier les résineux qui ont une organisation cellulaire simple, des feuillus qui ont une organisation plus complexe. Il existe aussi une forte variabilité au sein d'un même grand ensemble d'essence. Par exemple pour l'Acajou, les vaisseaux apparaissent comme une déformation continue des cellules au sein d'une file, alors que pour *Angustifolia*, ces derniers semblent parachutés au milieu des files.

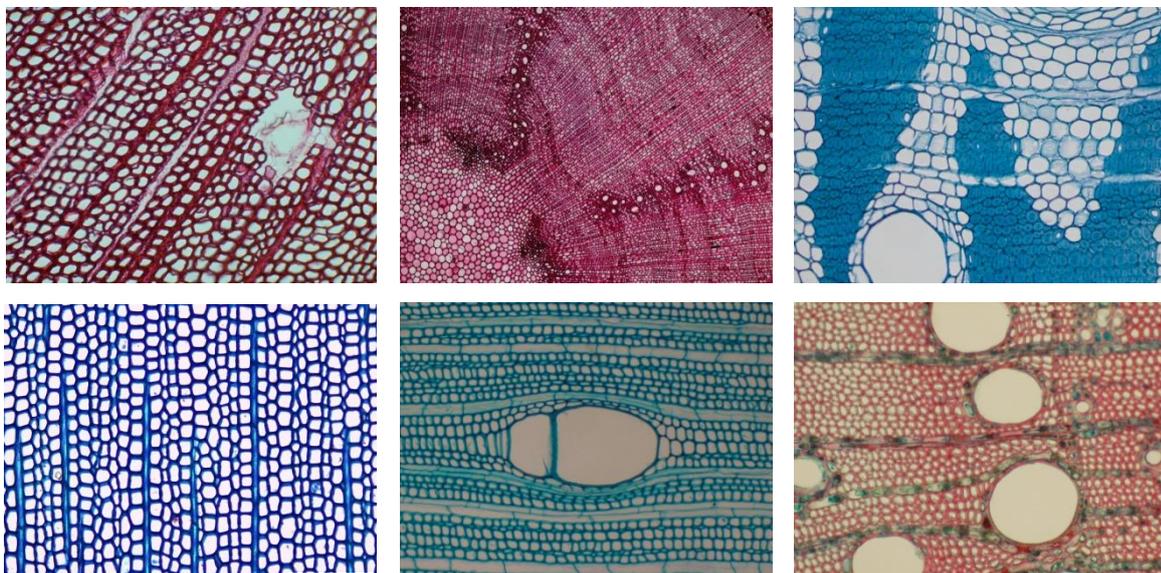


Figure 9. De gauche à droite: P. caribensis (résineux coloration carmine rouge), Chêne vert (feuillus), B. utiles (feuillus), Sapin (résineux, coloration bleu de toluidine, x400), P. Acajou (feuillus) et F. Angustifolia (feuillus).

Cette forte variabilité rend difficile la phase de prétraitement d'image car il faut trouver une méthode générique suffisamment robuste pour arriver à traiter tous les cas indépendamment. L'hétérogénéité des éléments structurants même au sein d'un sous-groupe d'essence est la principale source de difficultés, car il devient difficile de trouver des règles globales pour traiter tous les cas. Une étude essence par essence pourrait

être envisagée pour certaine espèce tant leur différence est grande par rapport à l'ensemble des autres groupes.

Dans le cadre du stage, je travaillerais sur des images de feuillus, principalement de l'acajou et du frêne. Quelques images de résineux (sapin et pin noir) seront également utilisées pour comparer et évaluer ma méthode avec celle déjà existante. L'utilisation d'un échantillon varié m'a permis d'avoir des tests de référence et de proposer une méthode plus générique englobant résineuse et feuillue (Cf. Annexe 3).

## 2.2 PRE-ETUDE

### 2.2.1 ETAT DE L'ART

Le sujet a déjà été abordé par des chercheurs mais n'a pu aboutir à cause de certains problèmes rencontrés. L'identification des cellules est rendue difficile par l'hétérogénéité de l'image et par la présence de différents artefacts biologiques : dédoublement local des files (Cf. Annexe 2), cellules d'interférences provenant de plans différents.

Voici un essai de reconnaissance de files sur une image d'Acajou grâce à un partitionnement par un diagramme de Voronoï<sup>11</sup> réalisé sur une image segmentée et localement retouchée notamment aux abords du vaisseau. On y voit le parcours des files (en jaune) perturbé aux alentours du vaisseau central, celles-ci s'arrêtent ou prennent un mauvais chemin. De plus certaines cellules ne sont pas reconnues.

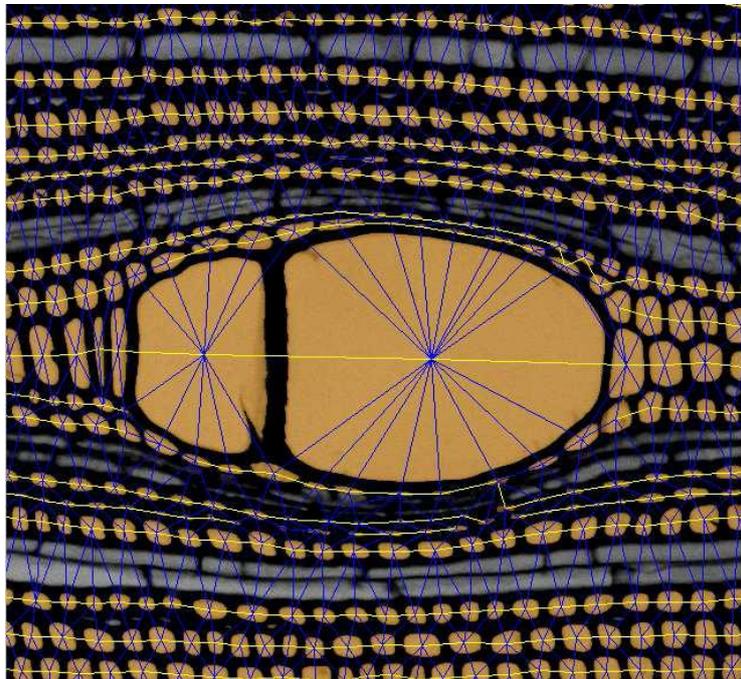
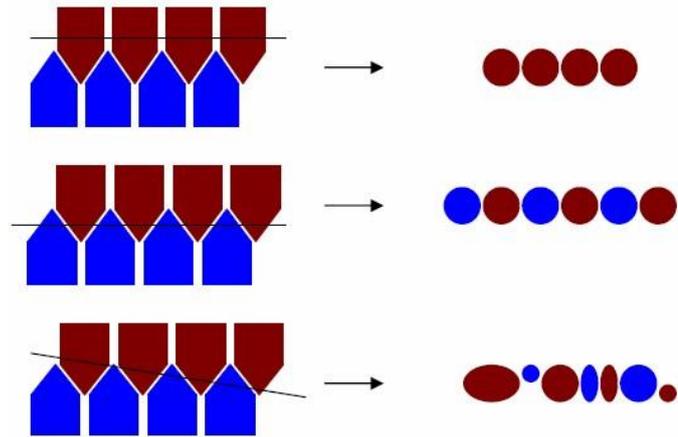


Figure 10. Reconnaissance de files sur l'Acajou

<sup>11</sup> Un diagramme de Voronoï représente une décomposition particulière d'un espace métrique déterminée par les distances à un ensemble discret d'objets de l'espace, en général un ensemble discret de points.

Un problème majeur expliquant l'intrusion de cellules n'appartenant pas au même étage cellulaire est le plan de coupe comme l'illustre le schéma ci-contre. Les cellules ont une forme fusiforme ; elles sont généralement disposées en quinconce pour former des emboîtements globalement réguliers, présentant toutefois des perturbations locales : à gauche, schématisation des empilements cellulaires ; à droite, représentation de la ligne cellulaire correspondant au trait de coupe noir.



### 2.2.1.1 LE PROJET FICELER

Lors du stage précédant une étude sur la détection chez les résineux a été mise en place sous forme de plugin ImageJ. Grâce aux documents laissés par Pol Kennel [13], j'ai eu accès à toutes les méthodes qu'il avait testé ainsi qu'à leurs résultats. J'ai pu ainsi me faire une idée précise des pistes à suivre.

La méthode développée comporte trois parties ordonnées : Le traitement de l'image, le typage des cellules et pour finir la détection des files cellulaires [12].

La méthode a été développée pour les résineux et donnent des résultats satisfaisants. Mais appliquée à des images de feuillus, les résultats ne sont pas aussi concluants. Ceci s'explique par la déformation des alignements et des cellules aux abords des gros vaisseaux.

### 2.2.1.2 LITTÉRATURE

Dans l'article « *Automated Arabidopsis plant root cells segmentation based on SVM classification and region merging* » [1] de Monica Marcuzzo & all en 2009, la démarche principale du traitement de l'image est commune à toutes les références. En premier lieu il y a une phase de prétraitement de l'image pour améliorer la lisibilité de l'image grâce à un seuillage par le filtre Otsu. Les bassins sont extraits grâce à l'algorithme de ligne de partage des eaux. Et ensuite les bassins sont typés par une méthode originale, la SVM classification. Parmi les méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik, les SVM constituent la forme la plus connue. SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995. Cette méthode est donc une alternative récente pour la classification. Cette méthode repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonction dites noyau (kernel) qui permettent une séparation optimale des données.

Les résultats publiés dans cet article semblent donner de bon résultat mais les méthodes à mettre en place semblent très couteuse en temps de calcul et à mettre en place. Nous nous essayerons les méthodes de prétraitement d'image dans notre partie de prospection.

---

### 2.2.1.3 LIMITES

Les méthodes que nous venons d'évoquer comporte des limites. Je vais évoquer celle du projet ficeler car j'ai pu en parler directement avec Pol Kennel (ancien stagiaire) et aussi parce que j'ai accès au code source de son projet. Ce qui n'est pas le cas des études bibliographiques.

Une première limite que nous pouvons citer apparait quand on lance le plugin. Une fenêtre de paramètre s'ouvre avec des paramètres à donner. Les informations pour remplir ces champs proviennent du traitement d'image (rayon à définir, etc.), ce qui n'a pas de sens pour un biologiste. Pendant le projet Ficelef nous porterons donc une attention très soutenue à ce problème. En effet le logiciel est destiné à un public de biologistes peu familier avec le traitement d'image. Il faudra donc minimiser leur intervention sur les paramètres de ces algorithmes.

La labellisation des cellules donne de nombreuses structures de type inconnues, ce qui peut provoquer un biais sur la détection des files qui va être en partie basé sur ces informations. Au vu du graphe d'adjacence nous avons choisi de mettre le typage en dernier lieu, car dans toutes les étapes de la méthode c'est celle qui peut produire le plus de biais et parce qu'une approche purement géométrique du graphe semble pouvoir être suffisante pour créer les files.

Sur une même, le résultat de la méthode va différer à chaque fois. Les différences ne sont pas énormes, sur une quarantaines de files créées, il va y en avoir cinq qui vont différer à chaque itération. Nous allons devoir prendre ce problème en compte et faire en sorte que la méthode soit reproductible à chaque fois.

Grâce à l'étude des limites du projet Ficeler nous avons une idée plus précise sur les pistes à suivre. L'étude faite l'an dernier nous a donc beaucoup apporté :

- en termes d'objectifs à atteindre et comment les atteindre
- en termes d'ergonomie pour les biologistes
- un gain de temps considérable grâce aux études faites sur le traitement d'image

---

## 2.2.2 NOUVELLE ETUDE PROSPECTIVE SUR LA SEGMENTATION

Quant à ma pré-étude, elle a consisté :

- à tester plusieurs méthodes de segmentation automatique en parcourant la bibliographie tout en cherchant un prétraitement améliorant celle-ci à chaque fois.
- à trouver et d'identifier les faiblesses des différentes méthodes.
- à cibler les zones à difficulté récurrentes et de comprendre quelles sont les raisons biologiques.
- A suivre un module<sup>12</sup> de traitement d'image au LIRMM pour accroître mes connaissances en traitement d'image et trouver de nouvelles méthodes.

---

<sup>12</sup> Module Image, « ce module s'adresse aux étudiants de l'école doctorale I2S et aux chercheurs intéressés par les techniques de traitement d'image et de vision par ordinateur », <http://www.lirmm.fr/ModuleImage/>

### 2.2.2.1 ESPACE DE COULEUR

Nous nous sommes posé la question de l'espace de couleurs le mieux adapté lors des traitements réalisés précédemment. En effet certains espaces apportent d'autres informations (contraste, saturation...) et permettent parfois un meilleur traitement (la segmentation notamment). De même les différents canaux R, G, B (pour Rouge, Vert et Bleu) ont de la même façon été testés.

Nous nous sommes notamment penchés sur le HSV, reconnu pour faire ressortir certains aspects colorimétriques. HSV est l'acronyme pour *Hue Saturation Brightness* en anglais ou TSB en français pour Teinte Saturation brillance où :

- teinte : le type de couleur (comme rouge, bleu, jaune...)
- saturation : l'intensité de la couleur
- brillance : l'éclat de la couleur

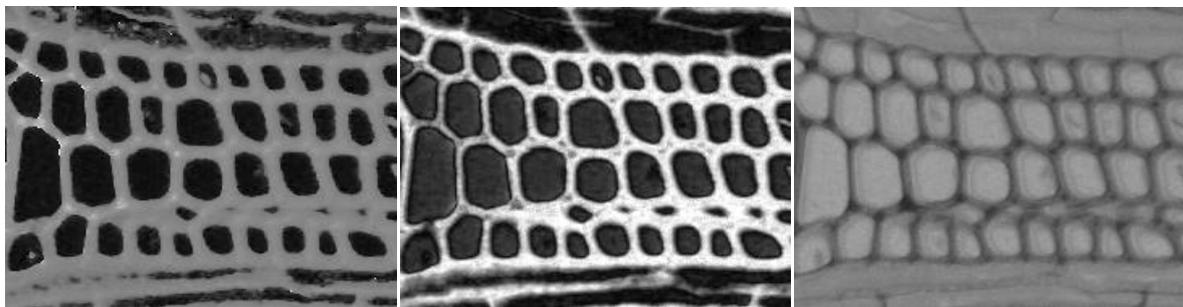


Figure 11. De gauche à droite : Teinte, Saturation, Brillance. Sur Acajou

Bien que l'image de la saturation fasse bien ressortir le contraste lumen/paroi, cette méthode n'a pas été conservée car elle est entièrement dépendante de la coloration de l'image. Pour la coloration au *bleu de toluidine* la méthode est très efficace, alors que pour la coloration au *carmino rouge* la méthode commence à montrer ses limites. Il se peut que de nouvelles colorations soient utilisées par les biologistes. Cette technique se basant uniquement sur la coloration, il se peut que la méthode devienne obsolète. Plutôt que d'imposer un protocole très rigide au niveau de la coloration, nous avons décidé de chercher d'autres méthodes avant de conserver celle-ci.

### 2.2.2.2 LES FILTRES

L'application de filtres sur une image a pour but d'améliorer leur qualité et/ou d'en extraire de l'information. La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information. En effet, de nombreux paramètres entrent en compte, les principaux étant :

- la résolution d'acquisition et le mode de codage utilisé lors de la numérisation, qui déterminent le degré de précision des éventuelles mesures de dimensions
- les réglages optiques utilisés, (dont la mise au point) qui déterminent par exemple la netteté de l'image
- les conditions d'éclairage, qui déterminent une partie de la variabilité des images traitées
- le bruit de la chaîne de transmission d'image

Il existe trois grands types de filtres :

- Les opérateurs de traitement d'images
- Les Filtres linéaires : Lissage, détection de contours etc.
- Les opérateurs morphologiques

**Le filtre médian** réduit le bruit dans l'image active en remplaçant chaque pixel par la valeur médiane des pixels voisins. Le filtre prend en paramètre  $v$ , qui représente la taille en pixel du voisinage.

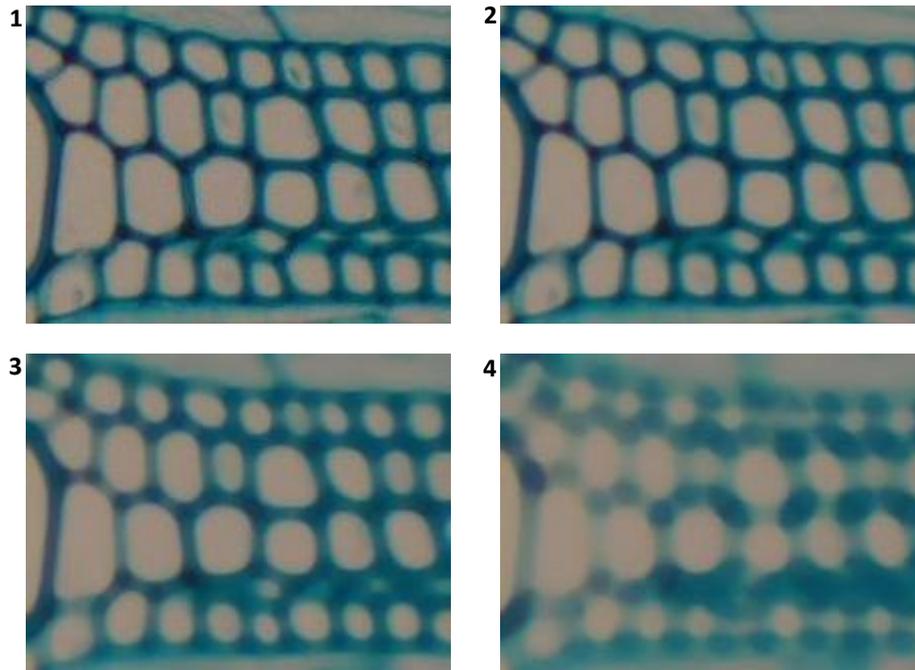


Figure 12. Application d'un médian avec  $v=0$  (1),  $v=2$  (2),  $v=6$  (3) et  $v=10$  (4). Sur Acajou

Le paramétrage des filtres est une étape primordiale car en dessous d'un certain seuil les filtres ne vont pas avoir suffisamment d'effet pour améliorer l'image et dépasser un seuil maximal l'image va se dégrader et donc nous allons perdre de l'information. Une étude sur le choix des paramètres sera faite sur chaque filtre pour trouver des éléments dans l'image qui nous permettront de les fixer automatiquement.

**La soustraction** fait partie des opérateurs logiques, dit point à point. Une opération point à point consiste à appliquer une fonction identique sur chaque pixel d'une image numérique. Ici l'opérateur va soustraire les valeurs d'une image aux valeurs d'une seconde image.

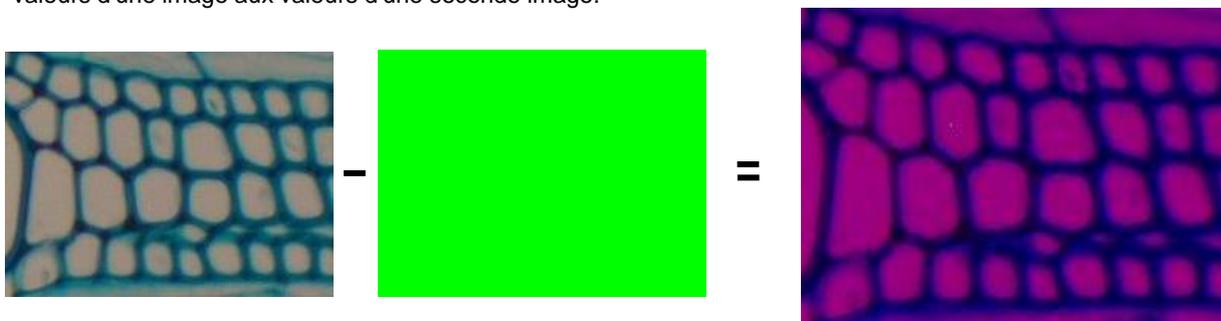


Figure 13. Soustraction de deux images

La soustraction se fait canal par canal. Ici nous avons soustrait une image vert (0,255,0) à notre image d'acajou. Ce qui a eu pour effet de mettre le canal vert de la nouvelle image à 0. Il n'y a donc plus de vert, il reste donc les valeurs des deux autres canaux.

**Le filtre flou** est le résultat d'un flou par une fonction gaussienne. Il s'agit d'un effet largement utilisé dans des logiciels graphiques, généralement pour réduire le bruit d'image ou les détails. Tout comme le filtre médian il prend en paramètre une taille de voisinage, à partir du quel sera calculée notre fonction gaussienne.

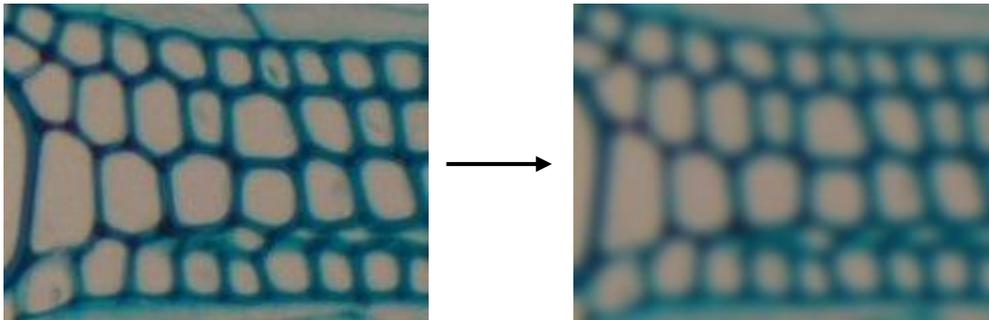


Figure 14. Résultat d'un flou, taille du masque = 3

Nous avons choisi ici une fenêtre de 3\*3 pixels pour notre noyau gaussien, ce qui nous donne la formule suivante :

$$\begin{pmatrix} g(-1,-1) & g(0,-1) & g(1,-1) \\ g(-1,0) & g(0,0) & g(1,0) \\ g(-1,1) & g(0,1) & g(1,1) \end{pmatrix}$$

$$\text{Où } g(x, y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Avec  $\sigma = 2$ , comme l'utilise les fonctionnalités d'ImageJ par défaut. La taille du masque a une importance significative sur l'effet de sur/sous-segmentation. De manière générale plus l'image est zoomée et de grande résolution, plus on augmente le masque pour éliminer plus de détails.

### 2.2.2.3 LA SEGMENTATION

La segmentation consiste à créer des partitions dans l'image en identifiant des pixels similaires. Le critère de similarité permet de contrôler l'aspect final de la segmentation. La segmentation a pour objectif de différencier des zones d'intérêt (par exemple objets / fond). C'est généralement une première étape d'un traitement plus complexe comme le filtrage adaptatif ou la reconnaissance de forme. Les méthodes de segmentation étant sensibles au bruit, il est nécessaire de commencer par nettoyer l'image en appliquant les filtres usuels d'atténuation de bruit (comme le médian ou un flou gaussien).

Cette approche se distingue, par exemple, des segmentations par contours ou par seuillage dans lesquelles les régions créées ne possèdent pas toutes ces propriétés.

### 2.2.2.3.1.1 LE SEUILLAGE

---

L'opération dite de "seuillage simple" consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur (appelée seuil, en anglais treshold) et à la valeur maximale les pixels ayant une valeur supérieure. Ainsi le résultat du seuillage est une image binaire contenant des pixels noirs et blancs, c'est la raison pour laquelle le terme de binarisation est parfois employé. Le seuillage permet de mettre en évidence des formes ou des objets dans une image. Toutefois la difficulté réside dans le choix du seuil à adopter.

Généralement le seuil est trouvé grâce à un clustering d'ordre 2 (méthode décrite dans le prochain paragraphe) sur l'histogramme de l'image. Cette méthode peut s'avérer dans certain cas très efficace. Dans notre cas la complexité des images (coloration et zoom) ne nous permet pas de trouver rapidement le seuil optimal pour traiter notre image.

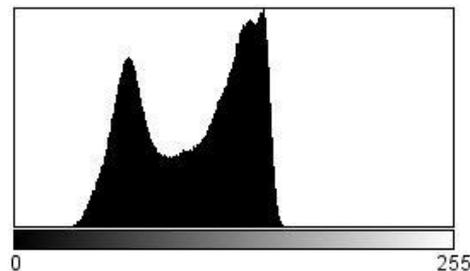


Figure 15. Histogramme en niveau de gris de l'image d'acajou

Le seuillage nécessite au préalable un passe en niveau de gris (8bit). Cette transition est faite naturellement dans ImageJ par une fonction linéaire sur les trois canaux. Cette méthode ne permet pas de maximiser la différence entre les différents canaux (R,G,B). Nous avons essayé de résoudre ce problème par une méthode statistique, que nous exposerons au paragraphe 2.2.2.3.1.4 Analyse en composante principale.

### 2.2.2.3.1.2 LE CLUSTERING

---

Une autre approche intéressante de segmentation, du point de vue région, est le clustering, méthode largement employée dans le domaine du partitionnement de données quelles qu'elles soient. Le clustering est une méthode statistique d'analyse de données qui a pour but de regrouper un ensemble de données en différents paquets homogènes, en ce sens que les données de chaque sous-ensemble partagent des caractéristiques communes, qui correspondent le plus souvent à des critères de proximité que l'on définit en introduisant des mesures de distance.

Plusieurs types d'algorithmes de clustering existent ; nous avons retenu pour nos tests celui des k-means, car simple à comprendre et d'une efficacité éprouvée : il permet de répartir un ensemble de données en  $k$  classes.

Nous n'avons pas retenu cette méthode car elle présente au finale un grand nombre d'inconvénient. Notamment le nombre de classe à définir. En effet nous ne pouvons pas connaître à l'avance le nombre de classe optimaux pour notre image. Il existe des méthodes comme les méthodes bayésiennes (ex : Calinski Harabasz [9]) pour déterminer ce critère, mais ces méthodes impliquent d'avoir testé itérativement un grand nombre de classe et de les comparer par la suite. Un autre point négatif du clustering est la non reproductivité des résultats. Pour un même nombre de classe avec des graines aléatoire (géré par le plugin), on ne va pas recréer les mêmes groupes. Pour trouver les graines optimales, il faut utiliser d'autres méthodes bayésiennes. Ces méthodes sont conteuses en temps de calcul et complexe.

### 2.2.2.3.1.3 L'ALGORITHME DE LIGNE DE PARTAGES DES EAUX (LPE)

La ligne de partage des eaux (LPE ou watershed en anglais) utilise la description des images en termes géographiques. Une image peut en effet être perçue comme un relief si l'on associe le niveau de gris de chaque point à une altitude. Il est alors possible de définir la ligne de partage des eaux comme étant la crête formant la limite entre deux bassins versants. Cet algorithme est détaillé au chapitre *Ligne de Partages des Eaux*.

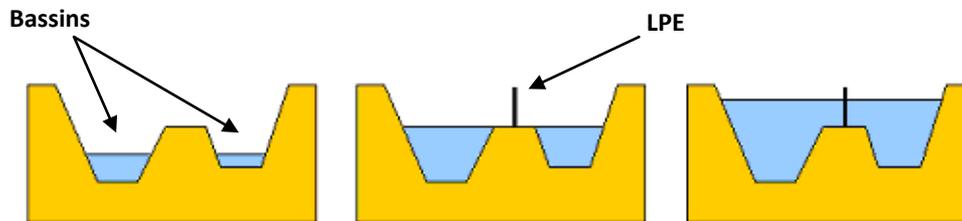


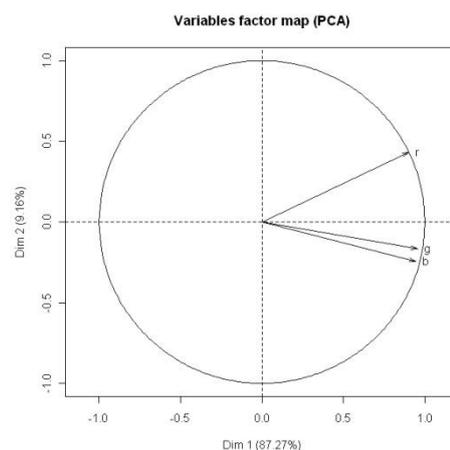
Figure 16. LEP entre deux bassins

Cette méthode nous fournit des résultats intéressants, mais certains points restent à préciser. En effet le défaut majeur de cette méthode est la sur-segmentation qui est due à beaucoup de minimaux locaux présents dans une image. Nous avons donc dû remédier à ce problème en « lissant » l'histogramme de notre image en appliquant plusieurs filtres que nous détaillerons par la suite.

### 2.2.2.3.1.4 L'ANALYSE EN COMPOSANTE PRINCIPALE (ACP)

Le passage d'une image couleur (32bit) en une image en niveau de gris (8 bit) est réalisé par une équation linéaire. Les 3 canaux (de 8bit) de l'image couleur sont divisés par  $\frac{1}{3}$  et additionnés pour ne former plus qu'un canal de 8bit. Cette méthode ne permet pas de maximiser les différences des canaux et donc du contraste de l'image. Pour cela nous avons fait une étude statistique, l'ACP [2]. L'ACP est une méthode de la famille de l'analyse des données et plus généralement de la statistique multi variée, qui consiste à transformer des variables liées entre elles (dites "corrélées" en statistique) en nouvelles variables indépendantes les unes des autres (donc "non corrélées"). Ces nouvelles variables sont nommées "composantes principales", ou axes. Elle permet au praticien de réduire l'information en un nombre de composantes plus limité que le nombre initial de variables. Dans notre cas trouver les ratios de chaque canal.

Cette étude m'a poussé à rencontrer M. Gilles Carraux (Enseignant Chercheur au LIRMM) pour conforter mes idées et exposé mon problème à un spécialiste. De cette discussion en est sortie qu'on ne pouvait pas travailler uniquement sur les trois dimensions (RVB) mais qu'il faudrait prendre en compte le voisinage de chaque pixel. Le schéma ci-contre nous montre le résultat de l'ACP qui représente l'importance des trois canaux de l'image d'acajou.



Cette méthode prometteuse a été laissée de côté car elle était couteuse en temps et que dans la même période lors d'un cours du module image, j'ai appris de nouvelles méthodes qui se sont avérées très efficaces.

#### 2.2.2.4 LES DIFFERENTES PREPARATIONS DES IMAGES

Comme nous l'avons vu lors de la discussion sur la préparation des images, il existe un grand nombre de préparations des images. Nous avons mené des tests sur deux grands types de préparation : Coupé coloré et Coupé non coloré.

Les méthodes que nous avons testées ne marchent pas sur les coupes non colorées. Nous allons essayer de comprendre pourquoi ces méthodes ne donnent pas les mêmes résultats.

La première piste fût de regarder les différences sur l'histogramme des deux préparations. Nous n'avons pas vu de différence pouvant amener à l'échec de nos méthodes. La seconde piste fût de regarder localement les profils de chacune des images et notamment à l'interface de la paroi et du lumen.

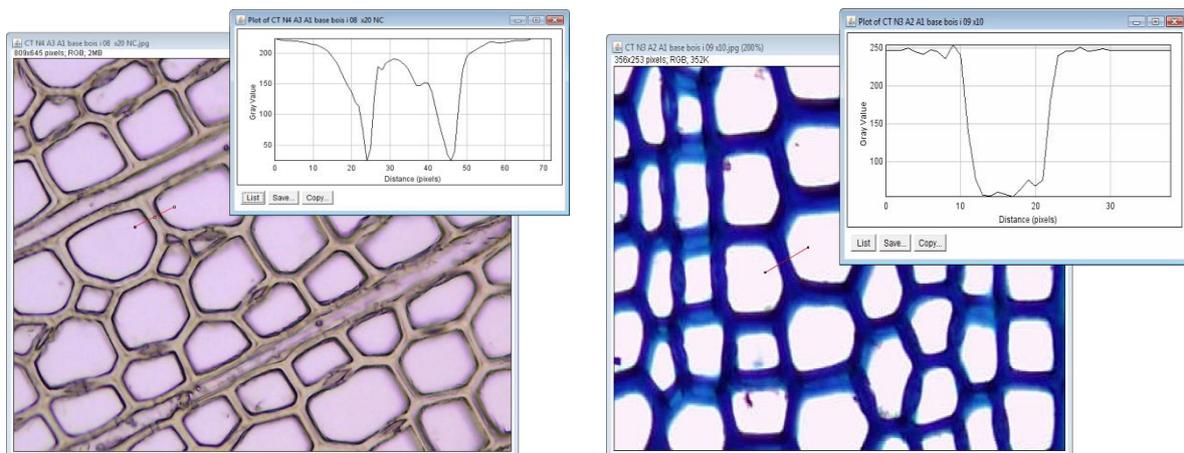


Figure 17. De gauche à droite : Profil d'une coupe non coloré, Profil d'une coupe coloré. Chez le sapin

Comme on peut le voir sur les profils, il y a un inversement local du profil chez les coupes non coloré. Le profil est en **W** alors que pour les images colorées le profil est en **U**.

Cette différence s'explique par la faible différence de contraste entre la paroi et le lumen, ainsi que par le liseré noir qui détoure le lumen.

#### 2.2.2.5 CONCLUSION

Toutes les méthodes classiques existantes pour la segmentation des images cellulaires ne sont pas satisfaisantes car pas assez robustes et pas entièrement automatiques.

Tout au long de cette pré étude, il est apparu clairement qu'un traitement générique pouvant traiter aussi bien les images produites par le protocole avec coloration (Acajou) et sans coloration (Epicéa) n'était pas possible.

Le problème majeur reste la variabilité existant entre toutes les images aussi bien d'un point de vue général que d'un point de vue local, une variabilité aussi bien protocolaire qu'humaine.

L'approche qui donne les meilleurs résultats est la segmentation par LPE après avoir réalisé un travail de débruitage de l'image L'enchaînement des méthodes a en particulier pu résoudre le problème de la séparation entre les trachéides et le parenchyme. Nous allons maintenant détailler cet enchaînement de filtre et leur paramètre.

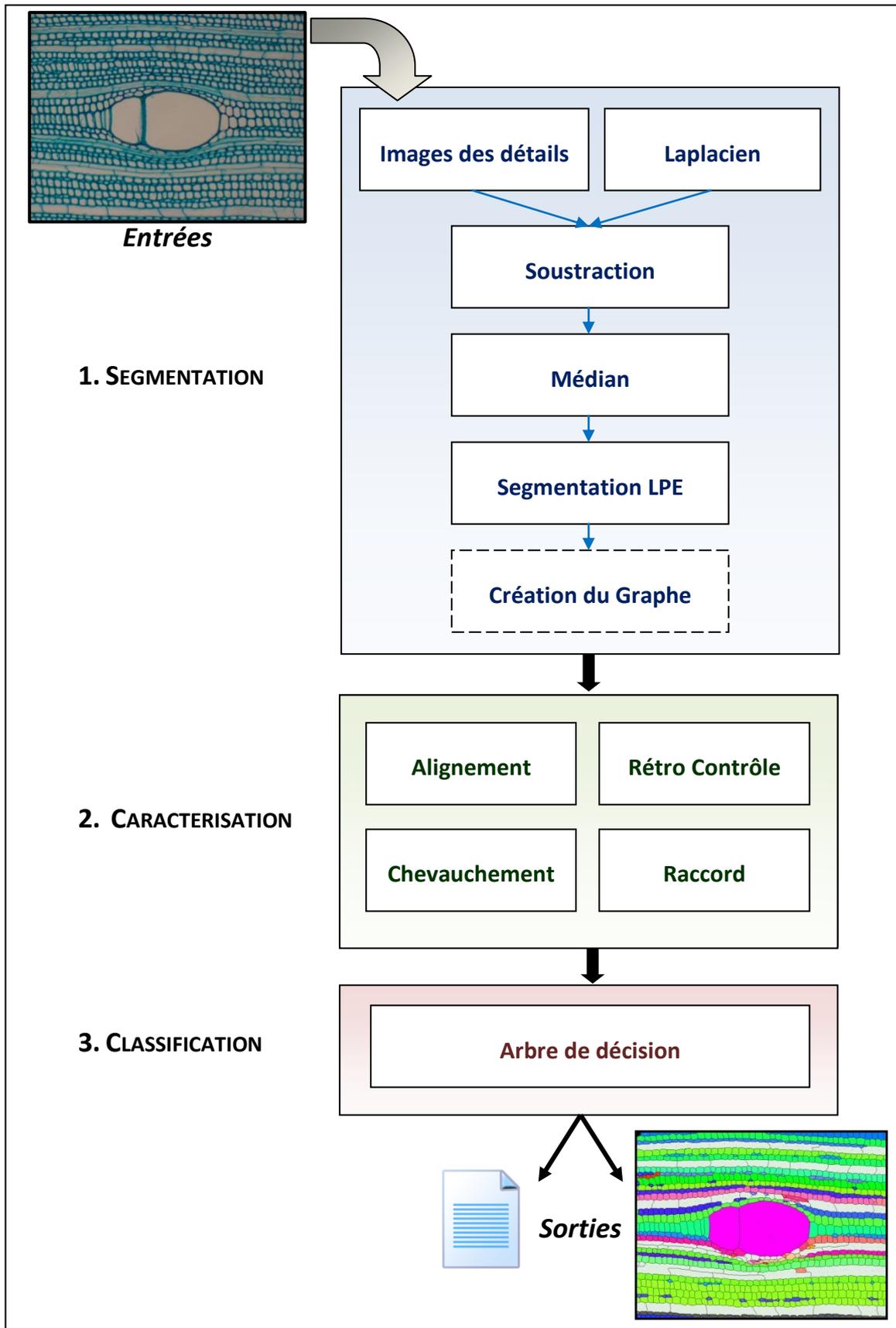


Figure 18. Chaîne de traitements établie pour le plugin ficelef

## METHODE MISE EN ŒUVRE :

Nous allons maintenant décrire la chaîne de traitement adoptée pour la détection automatique des files cellulaires. Chaque étape sera détaillée et brièvement discutée notamment sur l'aspect paramétrage, après avoir présenté synthétiquement l'enchaînement général des algorithmes (Schéma ci-dessus : Figure 18. Chaîne de traitements établie pour le plugin ficelef).

La chaîne de traitement comporte trois grands axes :

- La segmentation en bassin de l'image
- La caractérisation des files cellulaires
- La classification des types cellulaires

Par rapport au planning prévisionnel on peut remarquer que la partie typage des cellules est passée après la détection de files. En effet suite à des réunions avec les biologistes, on a pu constater que le typage n'est pas une information sûre. On peut notamment citer que pour une même cellule, les biologistes ne typeront pas de la même façon la cellule si on leur demande séparément. Exemple (Cf. Figure 19. Cellule à problème), la cellule en vert sera une fois typée en tant que fibre et une autre fois en tant que vaisseau. Après concertation avec les deux biologistes, la cellule sera finalement considérée comme étant un vaisseau à cause de la granularité de la paroi.

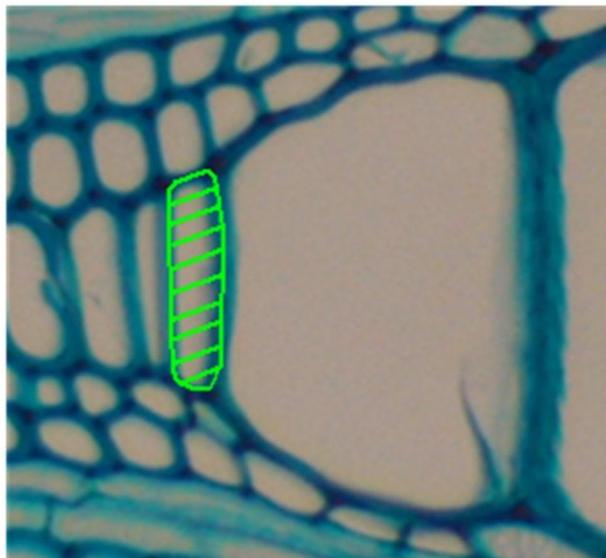


Figure 19. Cellule à problème

La classification est donc l'étape la moins fiable et peut induire un biais. Construire les files à partir des types cellulaires n'a donc pas semblé être une idée judicieuse et nous avons donc décidé de reporter cette étape à la fin de nos traitements.

Nous allons maintenant détailler la chaîne de traitements complète prenant en entrée une image prise par le biologiste et produisant en sortie une image résultat identifiant les files et un fichier de données avec les caractéristiques.

Nous illustrerons le résultat de chaque traitement sur un fragment de l'image teste d'Acajou :

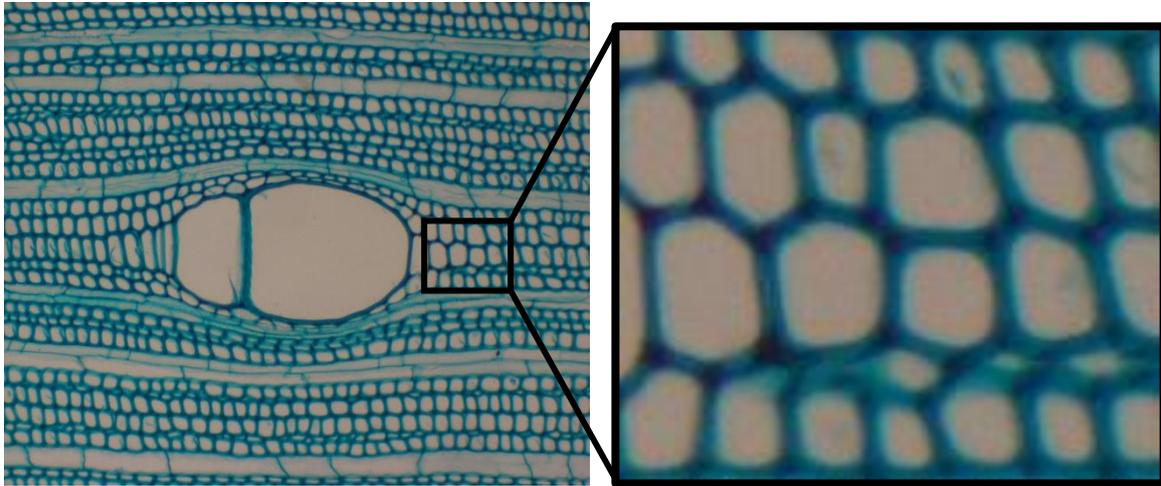


Figure 20. Fragment de l'image d'Acajou, utilisé pour nos tests

### 2.3.1 SEGMENTATION

L'étape de segmentation a pour but d'identifier et extraire les bassins (cellules) de l'image, pour pouvoir construire notre graphe d'adjacence des bassins. Le graphe d'adjacence est à l'interface entre le traitement d'image et la modélisation géométrique. Nous avons choisi de le mettre dans la modélisation informatique car il portera toutes les informations pour la détection des files, nous détaillerons cette partie dans le paragraphe « Création du graphe d'adjacence » ci-après.

La segmentation automatique des images fournies se décompose en deux grandes parties qui sont :

- Le traitement de l'image
- La segmentation en régions

#### 2.3.1.1 TRAITEMENT DE L'IMAGE

Les traitements appliqués dans cette partie ont non seulement pour but d'améliorer la lisibilité de l'image en supprimant au maximum le bruit et en atténuant les variations locales, mais aussi de soustraire un maximum de détails pouvant perturber la segmentation et de mettre en avant les éléments biologiques. Un bon travail à ce niveau évite une trop grande sur-segmentation<sup>13</sup> car élimine les minimums locaux identifiés par l'algorithme de partage des eaux comme nous le verrons dans la partie suivante.

Nous allons maintenant détailler les différents filtres utilisés avant la création du graphe d'adjacence.

<sup>13</sup> Phénomène d'arête surnuméraire qui apparaît lors de la segmentation en région issu de la LPE.

### 2.3.1.1.1 IMAGE DES DETAILS

Le premier filtre appliqué est le filtre Images des détails qui à l'origine a pour but de faire ressortir les détails de l'image. D'un point de vu biologique les cellules apparaissent comme des détails, éléments caractéristique de l'image alors que les parois sont considérées comme un fond de l'image. L'image des détails est obtenue grâce à la soustraction de l'image native et de l'image ayant subie un flou gaussien. Cette méthode n'est pas utilisée dans les méthodes classiques de traitement d'image de cellules, elle est utilisée couramment dans l'extraction de points d'intérêt dans des images couleur. J'ai pu apprendre cette méthode pendant une conférence donnée par M. Montesinos<sup>14</sup> lors du module image du LIRMM.

Suite aux études expérimentales menées sur cette méthode, j'ai pu déterminer automatiquement la taille du masque de voisinage du flou gaussien en fonction des dimensions de l'image. On a pu remarquer que plus l'image est de grande taille, plus la taille du voisinage doit être importante pour uniformiser les couleurs.

**IMAGE DES DETAILS = IMAGE NATIVE – IMAGE FLOUTEE**

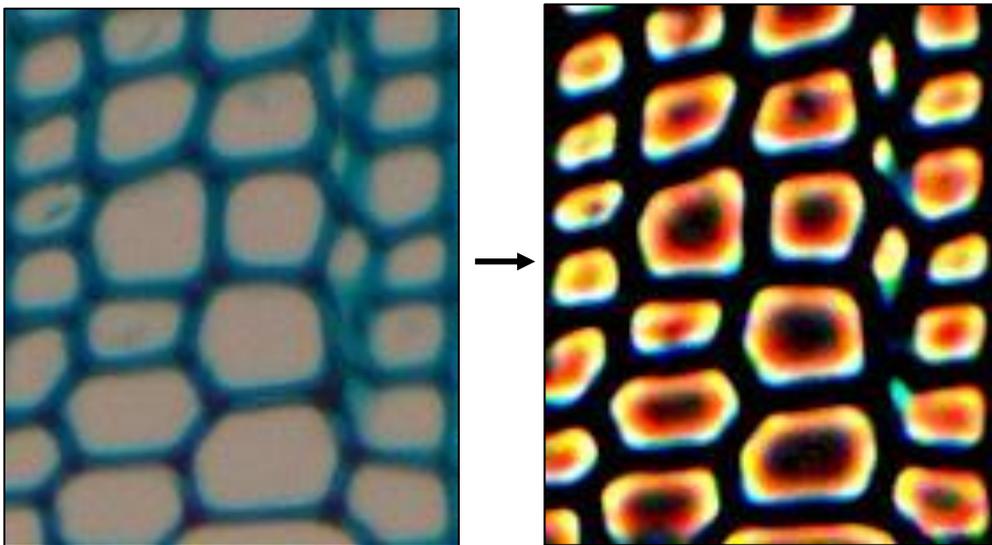


Figure 21. Résultat du filtre Image des détails

Les résultats à la sortie du filtre sont très satisfaisants, on obtient une image des lumens. La méthode est paramétrée automatiquement et n'est pas influencée par la couleur des images. Elle fait bien apparaître les lumens des cellules pour l'ensemble des images que ce soit chez les feuillus ou les résineux. Les parois cellulaires se retrouvent colorées en noire sauf pour quelques exceptions où il reste encore un peu de couleur. La prochaine étape aura pour but de corriger cette coloration. Cette méthode se base sur une différence de contraste de l'image.

Remarque sur le filtre gaussien : le filtre gaussien est un filtre natif du logiciel ImageJ, après une étude poussée de ce filtre, nous avons remarqué que quand la taille du voisinage atteint un certain seuil, l'image est réduite avant d'appliquer le flou puis remise à sa taille initiale pour éviter des temps de calcul trop grands. Si on applique indépendamment le même flou sur deux images identiques, on peut avoir des différences. Ces différences sont ponctuelles et infimes. Mais ce phénomène peut induire la non reproductibilité de notre méthode.

Extrait traduit du manuel d'ImageJ : « Pour augmenter la vitesse, sauf pour les petits rayons du flou, les lignes (lignes ou des colonnes de l'image) sont réduites avant la convolution et élargies à la taille d'origine par la suite. »

<sup>14</sup> M. Philippe Montesinos – Chercheur au LGI2P, EMA, Nîmes. <http://www.lgi2p.mines-ales.fr/~montesin/>

### 2.3.1.1.2 FILTRE PASSE BANDE (LAPLACIEN)

---

Pour corriger la coloration des parois, nous avons choisi de faire un filtre Laplacien par les flous. La méthode consiste à créer deux images floutées, une forte et l'autre faiblement. Et de les soustraire pour obtenir l'effet d'un filtre passe bande. Pour le paramétrage des flous, j'ai fait une étude sur les différentes informations que l'on peut trouver à partir de l'image (dimension de l'image, taille métrique du pixel, taille des éléments biologique, etc.) mais je n'ai pas pu identifier d'élément permettant de régler automatiquement les paramètres. Néanmoins j'ai pu fixer empiriquement des valeurs qui semblent optimales pour l'ensemble de mon jeu d'essai.

**LAPLACIEN = IMAGE FLOUTEE (FORT) – IMAGE FLOUTEE (FAIBLE)**

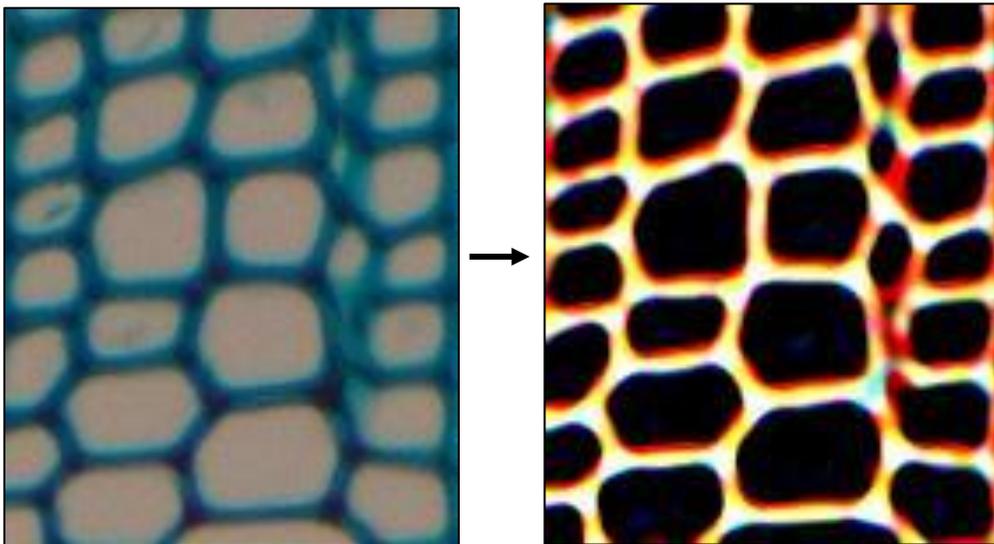


Figure 22. Résultat du filtre Laplacien par les flous

Ce filtre permet d'avoir les parois des cellules en couleur et les lumens en noir, on obtient donc l'image des parois. Ce qui peut être considéré comme le filtre inverse de l'image des détails. Ce filtre est considéré comme robuste car il arrive à détourner correctement les parois avec une nuance très proche des lumens et donc laisser de la coloration dans celle-ci alors que beaucoup d'autres filtres n'y arrivent pas.

### 2.3.1.1.3 SOUSTRACTION

---

Comme nous l'avons vu précédemment dans l'image des détails, il restait des imperfections dans les parois. Nous allons maintenant soustraire à l'image des lumens, l'image des parois. Cette étape donne une image des lumens améliorés. Dans le même temps nous passons d'une image 32bit<sup>15</sup> à une image 8bit (représenté en niveau de gris). Cette transformation est faite par une fonction linéaire. L'approche statistique (ACP<sup>16</sup>) n'a pas été gardée dans le laps de temps du stage, mais reste une piste envisagée.

---

<sup>15</sup> Une image en 32bit est codée sur trois canaux de 8bit (Rouge Vert Bleu). Il donne la couleur de l'image.

<sup>16</sup> Analyse en Composante Principale (ACP)

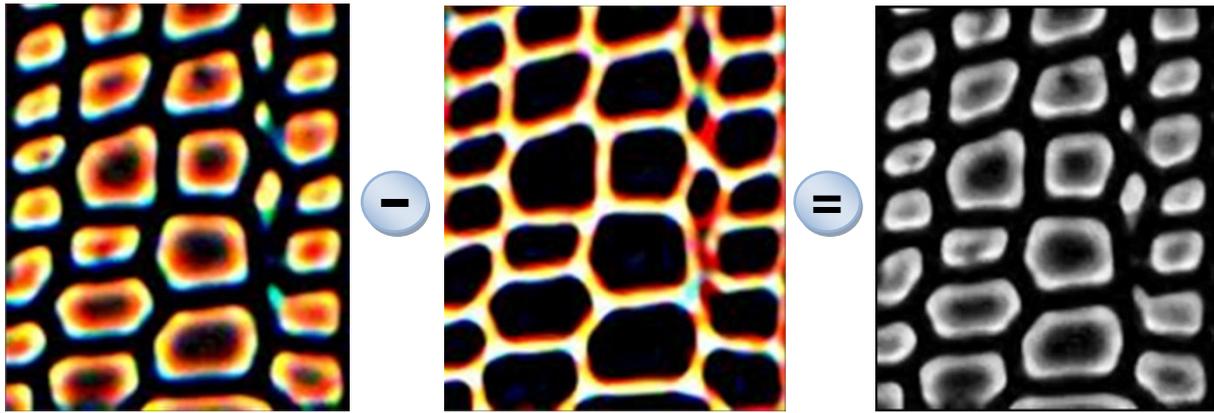


Figure 23. Schéma et résultat de la soustraction

Nous obtenons une image en niveau du gris avec les lumens en clair et les parois en foncé. Cette méthode permet donc de contraster d'avantage la différence parois/lumen. Sur l'image on peut encore voir du bruit, c'est-à-dire une granularité de l'image, qui peut être gênant lors de la segmentation de l'image. Néanmoins les étapes préalables ont permis de réduire considérablement le bruit général de l'image, le bruit restant est un bruit résiduel (ponctuel).

#### 2.3.1.1.4 FILTRE MEDIAN

Comme nous venons de le voir le bruit restant sur l'image peut être considéré comme bruit impulsif, c'est-à-dire très fort et ponctuel. Ce type de bruit peut être réduit par l'utilisation d'un filtre Médian. Son fonctionnement consiste à remplacer la valeur d'un pixel par la valeur médiane de l'ensemble des pixels de son voisinage défini par un masque carré de taille fixée.

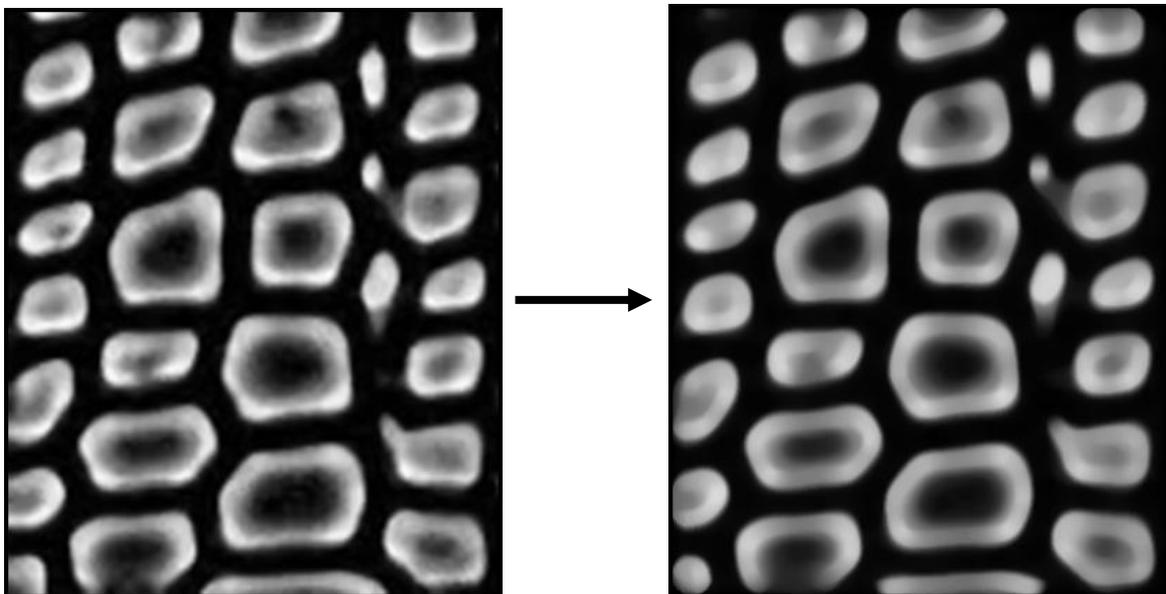


Figure 24. Résultat du filtre médian

Les résultats montrent bien que les derniers pixels bruités ont été corrigés, l'image a été lissée. On peut encore se pencher sur la taille du masque à utiliser dans le cas d'un traitement tout automatique. Après plusieurs essais il semble qu'un voisinage de taille égal à trois fonctionne sur toutes les images du jeu d'essai.

Remarque : par expérience nous avons choisi le filtre médian à d'autres filtres de réduction de bruit, comme le filtre Meanshift ou filtre de Kalman par exemple. Son temps de calcul étant plus faible pour des résultats aussi satisfaisants.

### 2.3.1.2 SEGMENTATION EN REGION

Cette étape a pour but de détourner le plus précisément possible les bassins, c'est-à-dire idéalement la séparation doit être fait entre deux parois de cellules voisine. Cette étape est cruciale car elle sera la base du graphe d'adjacence et des études faites sur les cellules. Nous avons choisi de conserver l'algorithme de Ligne de Partage des Eaux (LPE) ou Watershed [3]. Nous allons décrire les grands principes de cet algorithme dans cette partie comme on le trouve dans la littérature.

#### 2.3.1.2.1 L'ALGORITHME DE LIGNES DE PARTAGE DES EAUX (LPE)

L'idée de base derrière toutes les méthodes de segmentation reposant sur la ligne de partage des eaux est de considérer une image en niveaux de gris comme un relief topographique. Il s'agit alors de calculer la ligne de partage des eaux du dit relief. Les bassins versants ainsi obtenus correspondent aux régions de la partition.

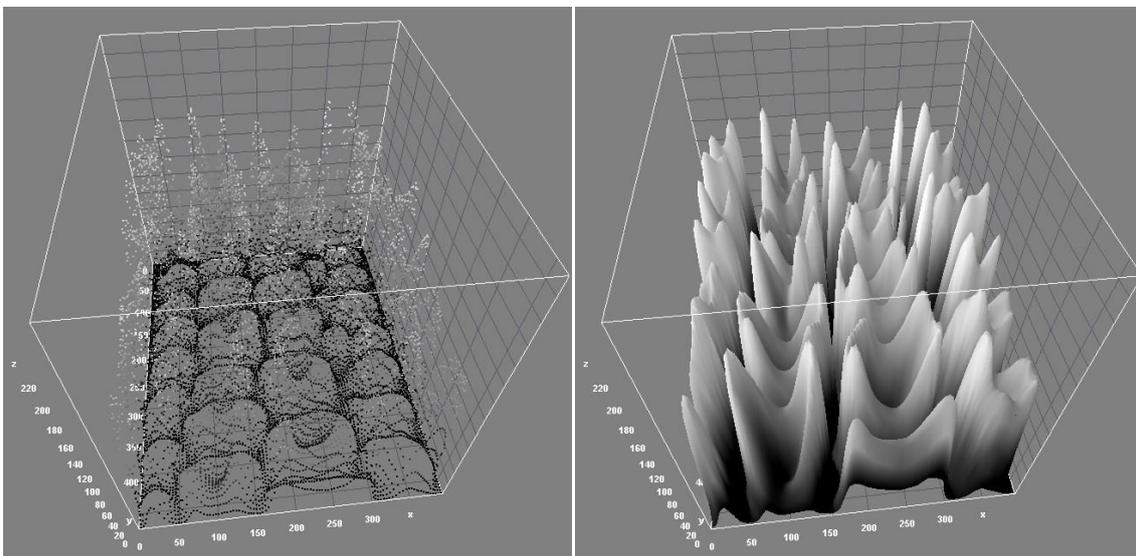


Figure 25. Relief topographique de notre partie de l'image d'acajou

Si on considère l'image à niveaux de gris comme une fonction de  $\mathbb{R}^2$  dans  $\mathbb{R}$  suffisamment régulière, on peut alors définir proprement la ligne de partage des eaux. Cependant, dans le cas discret (fonctions de  $\mathbb{Z}^2$  dans  $\mathbb{Z}$ ), qui correspond mieux aux images considérées, il n'existe pas à ce jour de définition consensuelle de cette transformation. Plusieurs algorithmes ont été proposés pour la calculer, donnant autant de définitions, pas toujours équivalentes.

On peut classer les algorithmes de construction de la ligne de partage des eaux en trois catégories. Les algorithmes par inondation simulent une montée progressive du niveau d'eau à partir des minima du relief. Les algorithmes par ruissellement suivent, à partir de chaque pixel de l'image, la ligne de plus grande pente jusqu'à atteindre un minimum comme le filtre *growing-region* [10]. Finalement, les algorithmes topologiques proposent une déformation progressive du relief, préservant certaines caractéristiques topologiques, jusqu'à ce qu'il soit réduit à une structure fine correspondant à la ligne de partage des eaux comme le filtre *Split/Merge* [11].

Nous nous intéresserons à la version par inondation dont voici le principe : la montée des eaux consiste à immerger la surface topographique dans de l'eau. A chaque fois que la hauteur des eaux atteint la hauteur d'un label, un nouveau bassin versant est créé dont la couleur est celle du label. A chaque fois que deux bassins de couleurs différentes se rencontrent, on empêche leur fusion en créant une digue, nommée ligne de partage des eaux. En revanche si les deux bassins sont de même couleur, ils fusionnent.

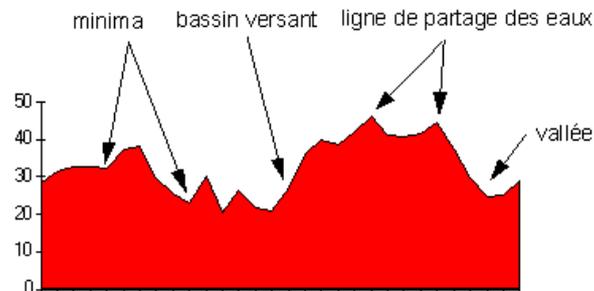


Figure 26. Illustration d'une surface topographique

Pour construire la ligne de partage des eaux, il suffit d'imaginer le procédé suivant :

- On perce un trou dans chaque minimum local du relief.
- On procède à une inondation du relief à partir des minima régionaux, de manière à ce que le niveau des eaux monte à vitesse constante et soit uniforme dans tous les bassins versants, en précisant que l'eau ne peut pénétrer dans les vallées que par ses minima.
- Chaque fois que les eaux issues de deux minima régionaux se rencontrent (les premiers points où cela se produit sont des points col ou selle du relief), on construit une digue le long de la ligne de crête, de manière à ce que les eaux issues de deux bassins versants différents ne se mélangent pas.
- Une fois le relief totalement immergé, les digues formées sont les lignes de partages des eaux (watersheds).

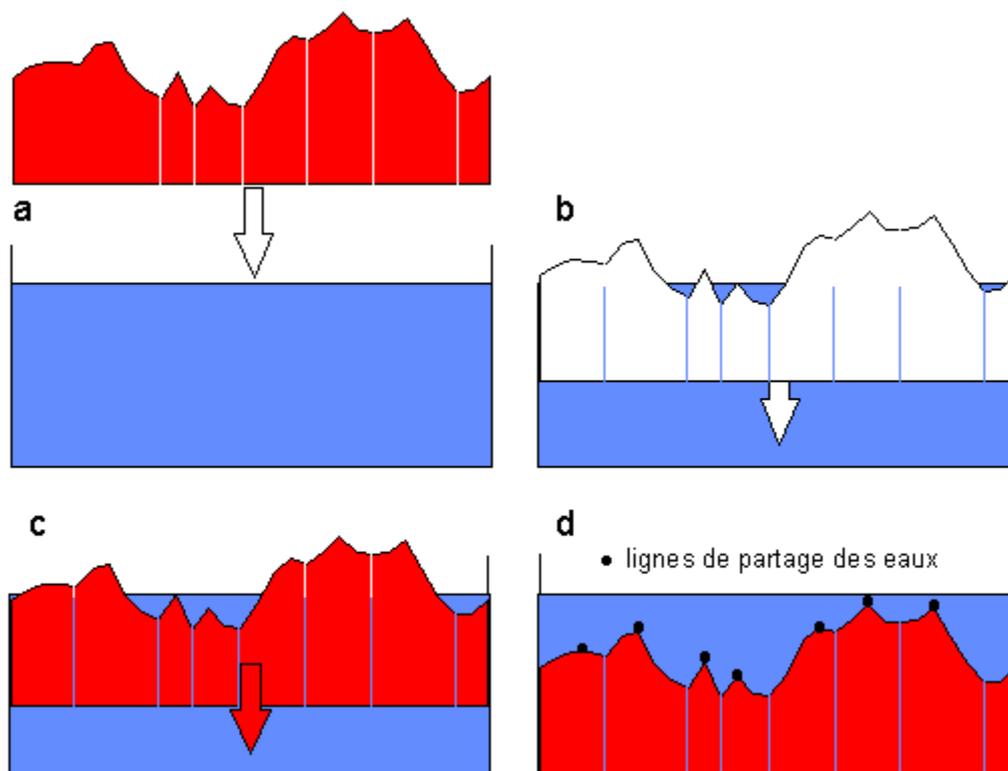


Figure 27. Illustration de l'immersion d'une surface topographique

Une des difficultés de la mise en œuvre de cette analogie intuitive est qu'elle laisse beaucoup de liberté quant à sa formalisation. Il existe deux classes principales d'implémentation : l'une est basée sur un algorithme récursif d'immersion et une autre basée sur l'utilisation de fonctions de distances géodésiques.

Nous avons repris une implémentation de Vincent and Soille [3] (1991) disponible librement au sein de la communauté de développeurs d'ImageJ. Nous l'avons adaptée à nos besoins, notamment inséré dans notre plugin. On trouvera en Annexe 4, un pseudo code de l'algorithme.

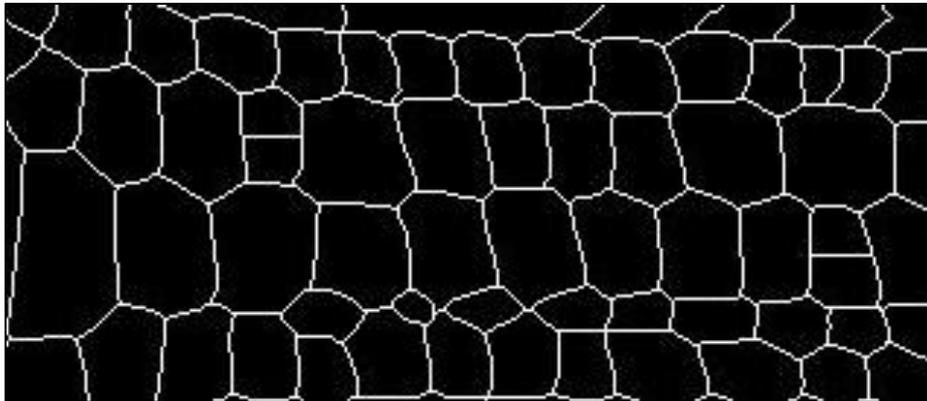


Figure 28. Résultat de l'algorithme LEP sur une portion de l'image d'acajou

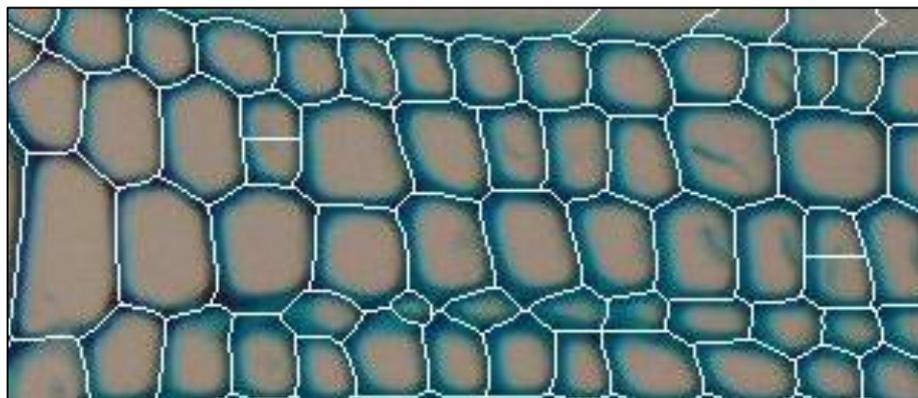


Figure 29. Superposition des lignes de partage des eaux avec l'image native

Les résultats sont satisfaisants hormis quelques zones sur-segmentées. En réalité la sur-segmentation dépend principalement du paramétrage des filtres utilisés durant le prétraitement. Comme nous l'avons vu dans la pré-étude, nous n'avons aucun moyen de remédier à la sur-segmentation restante. Néanmoins les étapes précédentes permettent de régler automatiquement la LPE ; en effet avant de faire la segmentation, j'ai choisi de borner les maxima des valeurs de l'histogramme entre 0 et 255 pour maximiser le contraste paroi/lumen et pour calibrer la LPE selon ces seuils (cette étape ne sera plus développée dans le rapport).

Il est à noter que les lignes fournies par l'algorithme n'étant pas toujours en 4-connextité<sup>17</sup>, nous avons appliqué l'algorithme de la squelettisation<sup>18</sup> sur cette image binaire. Ainsi les lignes de partage seront exploitables plus facilement lors de l'extraction des arêtes et sommets.

<sup>17</sup> Système de voisinage entre pixel, ici une grille carrée (nord, sud, est, ouest).

<sup>18</sup> La squelettisation fait partie du domaine de la morphologie mathématique : il fournit un squelette (trait d'épaisseur égale à 1 pixel) en partant d'une image binaire.

### 2.3.1.2.2 REDUCTION DES REGIONS

La réduction de régions suite à l'application d'un Watershed est une chose courante dans la littérature en traitement d'image. Nous avons choisi une méthode « originale », que nous n'avons pas retrouvée dans la littérature, pour réduire les régions. J'ai fait une méthode de voisinage qui va distinguer les arêtes et les sommets des bassins. Un sommet est un pixel qui sera commun à au moins 3 bassins, tandis qu'une arête sera la ligne (liste de pixels) qui joint deux sommets. Grâce à cette méthode nous avons pour chaque bassin (bords des cellules) une modélisation géométrique de son contour et pour chaque arête les deux bassins adjacents.



Figure 30. Résultat de caractérisation des arêtes et sommets

**Légende :** Les sommets sont identifiés par un point blanc, les arêtes ont été colorées (aléatoirement) et les bassins sont représentés en gris.

Une fois cette étape préliminaire finie, on peut grâce à l'image native comparer les arêtes selon leur profil d'intensité des pixels et supprimer les arêtes qui traversent le lumen des cellules et on met à jour les sommets. Avec ce système on remédie au problème de sur-segmentation sur l'ensemble des images.

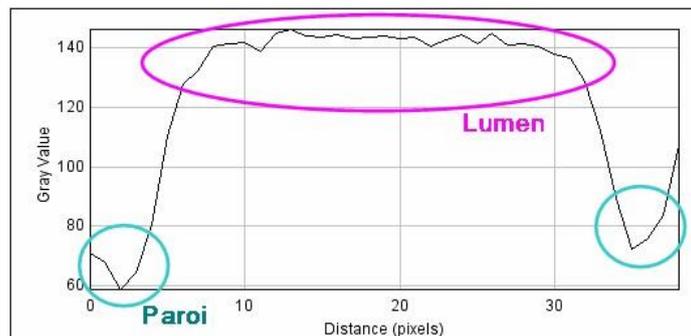


Figure 31. Profil d'une arête de sur-segmentation

Les arêtes et sommets n'appartiennent pas aux bassins. En effet si l'on prend la réalité biologique, les parois de deux cellules voisines sont séparées par la lamelle moyenne<sup>19</sup>, en bleu dans le schéma ci-dessous.

<sup>19</sup> « La lamelle moyenne est une couche composée essentiellement de composés pectiques qui sépare deux cellules végétales » Cf. Wikipedia, [http://fr.wikipedia.org/wiki/Paroi\\_pectocellulosique](http://fr.wikipedia.org/wiki/Paroi_pectocellulosique)

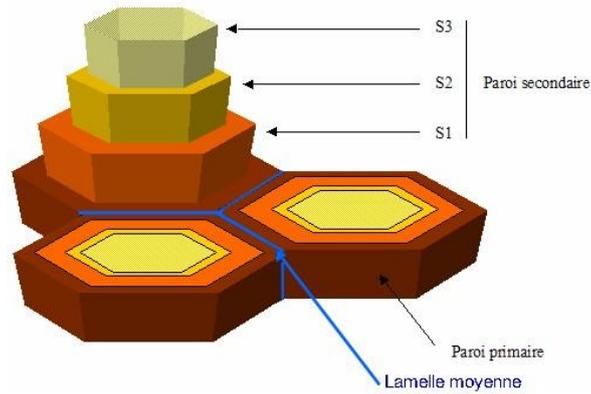


Figure 32. Morphologie paroi végétale

Nous avons donc choisi de ne pas inclure nos arêtes et sommets dans les bassins. L'adjacence sera portée par les arêtes qui auront comme identifiant l'indice des deux bassins voisins.

### 2.3.1.2.2.1 RESUME EN IMAGE DE LA METHODE

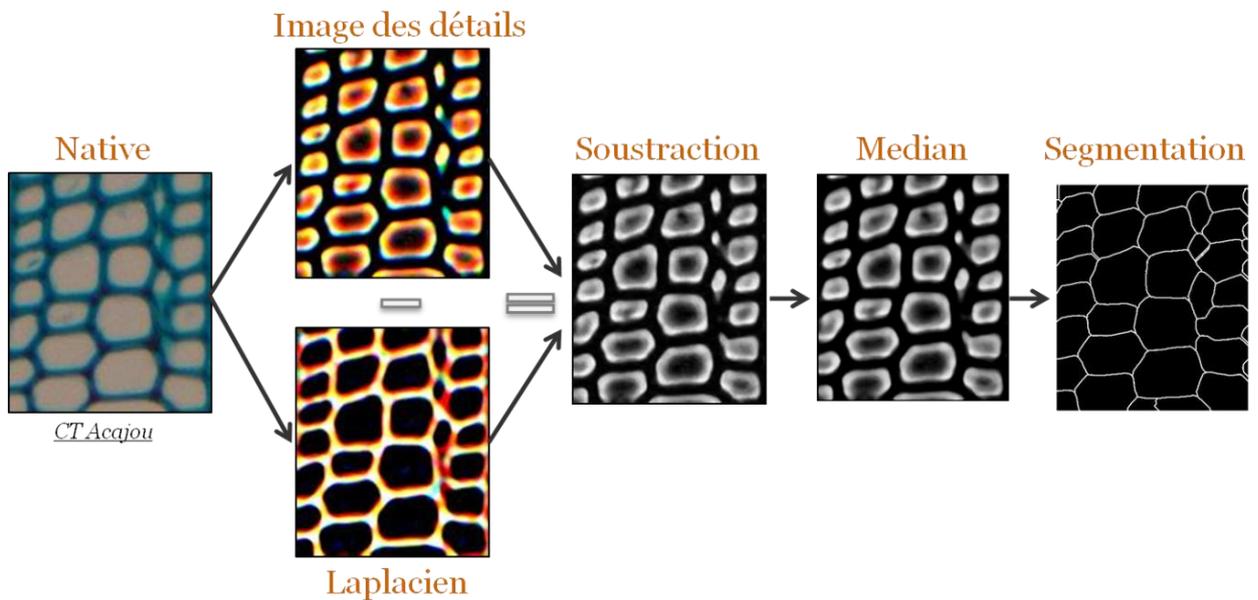


Figure 33. Traitement de l'image détail des filtres

Les difficultés rencontrées lors de l'élaboration de la chaîne de traitement est dans un premier temps de trouver les filtres optimaux parmi la multitude de disponible et déterminer leur agencement. Dans un second temps trouver leurs paramétrages. En effet, bons nombres de filtres et algorithmes nécessitent bien souvent d'être paramétrés, comme le flou gaussien par exemple où l'on définit la taille du masque. La réalisation d'une méthode automatique signifie qu'il ne faudrait dans l'idéal aucune intervention de l'utilisateur lors du traitement, ce qui implique que les paramètres soient définis automatiquement, et plus exactement fixés (en pouvant justifier le choix réalisé), soit calculés dynamiquement. Idéalement l'enchaînement des filtres devraient s'abstraire de la couleur, du niveau du zoom, de la résolution et des éléments biologiques. Le niveau de zoom reste problématique car avec un zoom trop faible il y a un sous échantillonnage des éléments biologiques qui empêche un détournage correct des cellules.

## 2.3.2 DETECTION DES FILES : CARACTERISATION

Nous avons à ce stage de la méthode une image binaire contenant le contour de chaque bassin. Nous allons dans un premier temps extraire de cette image un graphe d'adjacence des cellules, qui sera le support pour la détection des files. Nous détaillerons ensuite les méthodes (géométriques et topologiques) de détection pour la création des files. Comme nous l'avons évoqué en début de description des méthodes retenues, cette étape a été mise avant le typage cellulaire car on s'est rendu compte après plusieurs étude du graphe, qu'il serait possible de créer les files sans pour autant connaître la nature des cellules (notamment l'étude ci-dessous) et ainsi éviter les biais pouvant être induit par une mauvaise identification des types cellulaires.

*Etude faite avec mes neveux : Si l'on donne à un enfant (ici 4 et 6ans) l'image de l'acajou, il est capable instinctivement de reproduire les files, si on lui demande de dessiner les alignements qu'il voit, sans pour autant connaître la biologie.*

### 2.3.2.1 CREATION DU GRAPHE D'ADJACENCE

Comme nous venons de voir, nous avons extrait de l'image de la Lignes de Partage des Eaux une structure de données comportant une liste de bassins, représentés par des sommets et leur relation d'adjacence, matérialisée par des arrêtes. On trouvera en détail l'algorithme de détection de régions en annexe 8.

De ces données, nous allons donc créer un graphe d'adjacence. Le choix de la structure de graphe étant classique, nous avons choisi de garder la librairie JUNG<sup>20</sup> [17] utilisée dans le stage précédent car c'est une librairie performante portant bon nombre de variantes de structures et beaucoup de fonctionnalités optimisées.

Ici, l'intérêt de l'utilisation d'un graphe réside d'abord dans le fait que nous allons faire porter des informations sur les nœuds et sur les arêtes. En l'occurrence chaque nœud porte un objet « bassin versant de la LPE », alors que chaque arête porte un objet « arête » stockée dans notre structure. Le graphe créé est de type non orienté.

Les objets portés par le graphe sont donc les objets FBasin et FEdge identifiés dans notre classe FData. La description des ces classes est fournie en annexe 9. Nous allons faire une brève description des informations portés par chacun de ces objets :

- Un bassin est défini par : un identifiant, sa liste de pixels positionnés dans l'espace, sa taille, mais aussi une liste de masques portant sur différents états de l'image traitée (par exemple sur l'image d'origine, sur l'image prétraitée... etc.)
- Une arête est définie par : les identifiants des bassins qu'elle relie, la liste de pixels de la ligne d'adjacence<sup>21</sup> qui existe entre les centres de ces deux bassins ainsi que celle de la ligne de séparation<sup>22</sup> qui existe entre ces deux bassins, des informations de taille, une liste de masques des lignes d'adjacence et de séparation portés sur différents états de l'image, etc.

Nous avons choisi de garder les cellules du bord de l'image même si cela peut provoquer des effets de bord, ces derniers seront traités ultérieurement comme étant des bassins « non fiables » et on laissera au biologiste le soin de les conserver ou non pour leur analyse.

Remarque : suite aux discussions avec les biologistes, il a été évoqué que dans les futurs stages, il y aurait sûrement un pavage des images. C'est-à-dire une concaténation des images pour en obtenir une plus grande.

<sup>20</sup> Java Universal Network/Graph Framework, <http://jung.sourceforge.net/>

<sup>21</sup> Ligne d'adjacence : ligne reliant les centres de deux bassins voisins.

<sup>22</sup> Ligne de séparation : ligne séparant deux bassins voisins (équivalent à la ligne de partage).

Cela nous a conforté dans le choix de garder les cellules du bord de l'image pour rester cohérent avec cette perspective.

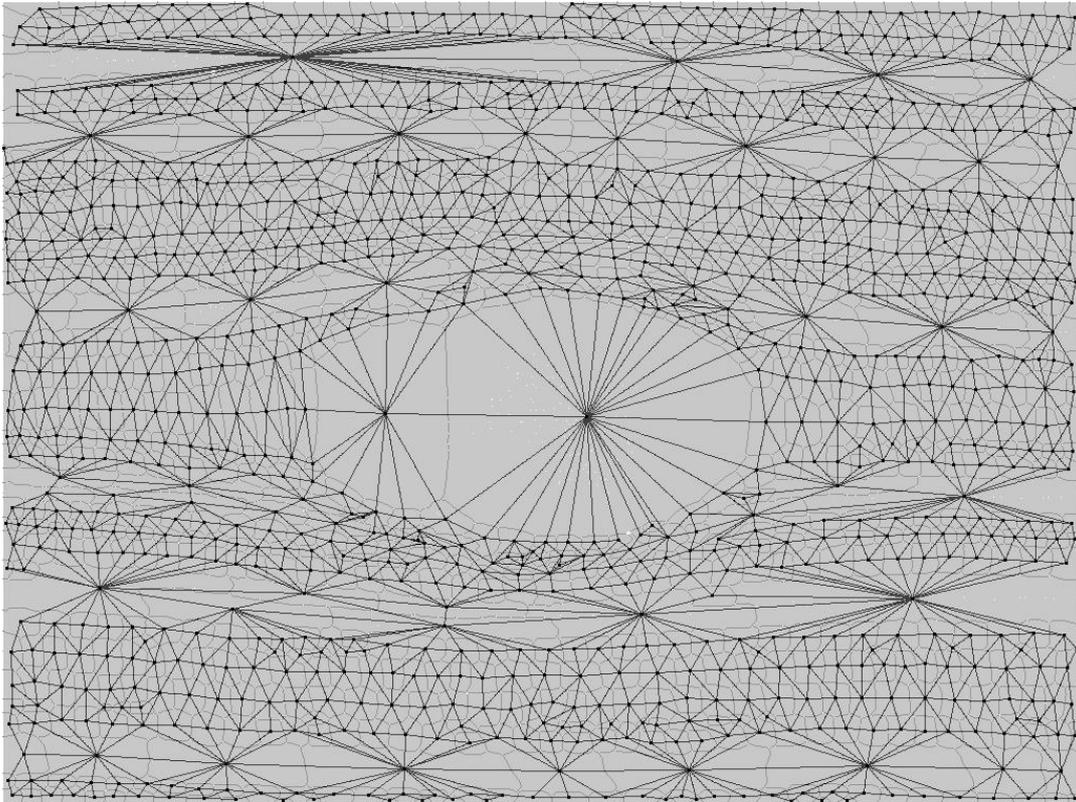


Figure 34. Superposition du graphe d'adjacence et des bassins "nettoyés"

Les résultats obtenus sont très satisfaisants. Nous retrouvons bien pour chaque cellule son contour bien défini, ainsi qu'en son centre le nœud d'adjacence.

### 2.3.2.2 MODELISATION GEOMETRIQUE

Nous expliquerons dans ce chapitre comment nous avons identifié les files cellulaires. Nous utiliserons largement le graphe réalisé précédemment, car il nous suffit, dans l'absolu, de le parcourir « correctement » pour obtenir nos files.

La première question qu'il faut se poser avant même de tracer les files est : dans quelle direction sont elles alignées ? Autrement dit, savoir comment est orientée l'image. L'orientation n'est pas constante car elle dépend de ce que cherche à voir le biologiste.

#### 2.3.2.2.1 L'ORIENTATION DES FILES

Pour réaliser cette tâche, il faut s'aider du graphe et plus particulièrement des arêtes du graphe. D'après l'article de Ronald J. [5] dont nous nous sommes largement inspiré, il suffit d'observer la distribution des angles que forme chaque arête par rapport à un axe fixé, et d'en extraire le pic le plus important. En effet, le graphe d'adjacence, vu la topologie de l'arrangement cellulaire du bois, porte une majorité d'arêtes qui suivent la direction des files. Car les cellules sont organisées en quinconce. Du coup, les directions secondaires sont réparties sur deux orientations, contrairement à la direction principale

Nous avons donc dans un premier temps calculé l'angle qui existe entre chaque arête et l'axe des abscisses afin d'obtenir une distribution angulaire.

Les arêtes obtenues sont comprises entre  $-89^\circ$  et  $90^\circ$ . Ainsi taguées, on peut observer l'histogramme de distribution des angles de l'ensemble des arêtes (un angle de  $-90^\circ$  est assigné à  $90^\circ$  pour ne pas disperser un éventuel pic).

### Histogramme des angles formés avec l'axe des abscisse

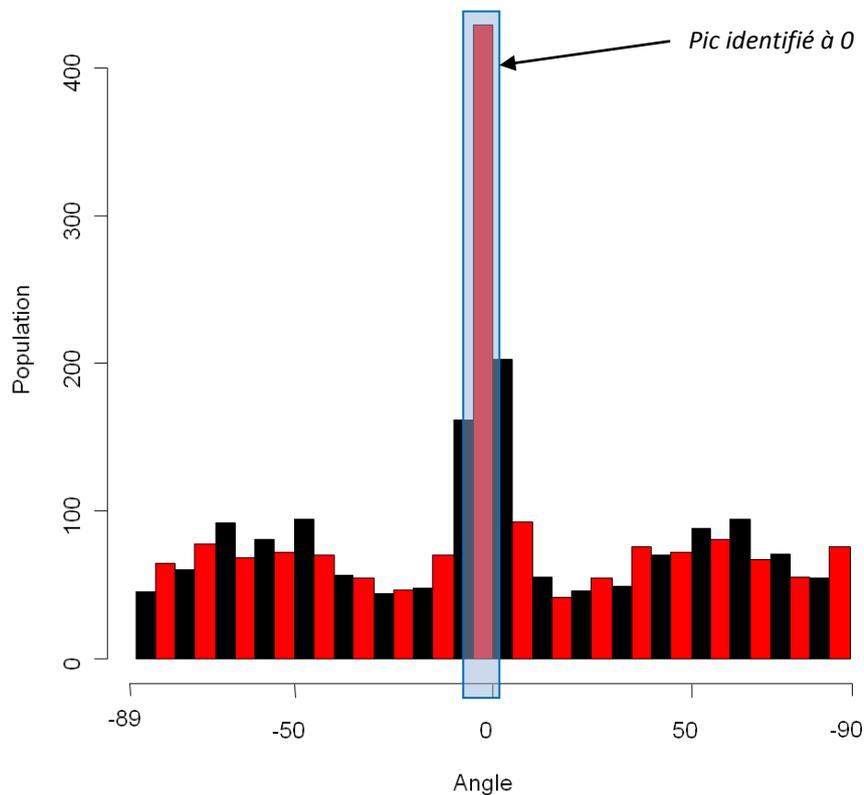
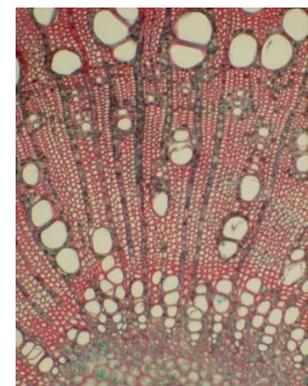


Figure 35. Distribution des angles formés par les arêtes avec l'axe des abscisses sur l'image d'acajou

De cet histogramme, il nous suffit d'extraire le pic maximum pour connaître l'orientation des files. Nous pouvons vérifier que cette direction est en accord avec l'estimation qu'en fait l'œil sur les images originales correspondantes, ou encore en faisant pivoter volontairement l'image de  $n$  degrés et vérifier le décalage du pic sur l'histogramme. L'emploi de cette méthode est robuste car il n'y aura jamais deux pics du fait de la distribution en quinconce des cellules

Cette information va nous servir à créer un vecteur principal de l'image pour commencer le parcours du graphe selon la bonne orientation.

Remarque : Cette méthode impose un niveau de zoom assez fort. Car sur un zoom faible, on pourra voir plusieurs directions de file, comme nous le voyons sur l'image de frêne ci-contre. Cela posera donc un problème lors de l'utilisation du vecteur pour le parcours. La méthode de pavage sera utilisée pour reconstruire une image contenant une grande portion de l'arbre.



### 2.3.2.3 MODELISATION TOPOLOGIQUE

La détection des files à proprement parler se base sur le parcours du graphe et en utilisant le vecteur directionnel pour démarrer nos parcours. Cette méthode comporte quatre grandes parties :

- L'alignement
- Le rétro contrôle
- Les chevauchements
- Les raccords

Nous allons maintenant détailler chacune de ces parties.

#### 2.3.2.3.1 ALIGNEMENT

Pour créer un alignement je me suis inspiré, après les avoir testés, de deux algorithmes de parcours de graphe : l'algorithme de colonies des fourmis<sup>23</sup> et l'algorithme de Dijkstra<sup>24</sup>, dont je vais expliquer succinctement leur principe.

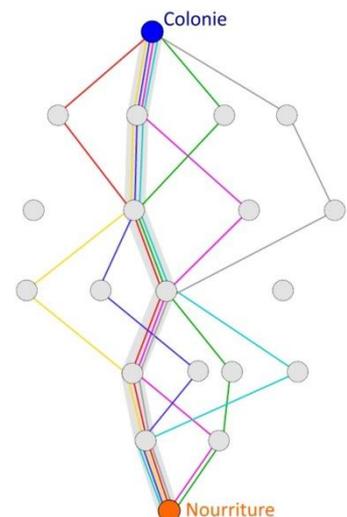
##### ❖ Algorithme de colonies des fourmis :

L'idée originale provient de l'observation de l'exploitation des ressources alimentaires chez les fourmis. En effet, celles-ci, bien qu'ayant individuellement des capacités cognitives limitées, sont capables collectivement de trouver le chemin le plus court entre une source de nourriture et leur nid.

Des biologistes ont ainsi observé, dans une série d'expériences menées à partir de 1989, qu'une colonie de fourmis ayant le choix entre deux chemins d'inégale longueur menant à une source de nourriture avait tendance à utiliser le chemin le plus court. Schéma en *Annexe 5*.

Un modèle expliquant ce comportement est le suivant :

1. une fourmi (appelée « éclairouse ») parcourt plus ou moins au hasard l'environnement autour de la colonie ;
2. si celle-ci découvre une source de nourriture, elle rentre plus ou moins directement au nid, en laissant sur son chemin une piste de phéromones ;
3. ces phéromones étant attractives, les fourmis passant à proximité vont avoir tendance à suivre, de façon plus ou moins directe, cette piste ;
4. en revenant au nid, ces mêmes fourmis vont renforcer la piste ;
5. si deux pistes sont possibles pour atteindre la même source de nourriture, celle étant la plus courte sera, dans le même temps, parcourue par plus de fourmis que la longue piste ;
6. la piste courte sera donc de plus en plus renforcée, et donc de plus en plus attractive ;
7. la longue piste, elle, finira par disparaître, les phéromones étant volatiles ;
8. à terme, l'ensemble des fourmis a donc déterminé et « choisi » la piste la plus courte.



<sup>23</sup> « Les algorithmes de colonies de fourmis sont des algorithmes inspirés du comportement des fourmis et qui constituent une famille de métaheuristiques d'optimisation », Cf. Wikipedia.

[http://fr.wikipedia.org/wiki/Algorithme\\_de\\_colonies\\_de\\_fourmis](http://fr.wikipedia.org/wiki/Algorithme_de_colonies_de_fourmis)

<sup>24</sup> L'algorithme de Dijkstra sert à résoudre le problème du plus court chemin entre deux points dans un graphe. [http://fr.wikipedia.org/wiki/Algorithme\\_de\\_Dijkstra](http://fr.wikipedia.org/wiki/Algorithme_de_Dijkstra)

Cet algorithme permet d'identifier plusieurs chemins. Le chemin qui remplit au mieux nos critères aura le plus haut « scores », tant dis que les autres perdrons de l'importance à chaque tours. Dans notre parcours nous n'essayons pas de prendre le chemin avec les arêtes les plus courtes, mais celui qui va maximiser nos critères (que nous verrons par la suite).

#### ❖ Algorithme de Dijkstra

Le principe de l'algorithme de Dijkstra, est de construire progressivement, à partir des données initiales, un sous-graphe dans lequel sont classés les différents sommets par ordre croissant de leur distance minimale au sommet de départ. La distance correspond à la somme des poids des arêtes empruntées. Permettant ainsi de à résoudre le problème du plus court chemin.

Ces deux algorithmes nécessitent de connaître le point de départ et surtout celui d'arrivé. Dans notre cas on peut supposer qu'une file va parcourir idéalement l'image de part en part et donc que l'arrivée se trouve à l'opposé de notre point de départ. Or ce n'est pas toujours le cas. Il peut y avoir par exemple des arrêts brutaux de files aux alentours des vaisseaux, ou des files d'intrusions qui ne traversent pas l'image de pars en pars.

Néanmoins ces algorithmes sont intéressants car ils m'ont permis de créer un parcours qui reprend certain de leur principe. Comme la notion de phéromones qui va nous permettre de donner un score à chacune de nos files pour essayer de créer un ensemble de files, non chevauchantes, dont les « scores » locaux et globaux sont minimaux pour l'ensemble de l'image

#### ❖ Méthode mise en œuvre :

Nous avons implémenté un parcours itératif qui va à chaque tour regarder un bassin et essayer de trouver les cellules voisines grâce au vecteur directionnel.

Pour le premier nœud on va recherche les nœuds voisins qui sont dans le sens du vecteur directeur avec une tolérance de  $\pm 35^\circ$ . Cet angle a été choisi en fonction de phénomène biologique Pour les nœuds suivants l'angle sera calculer grâce au nouveau vecteur crée à partir du nœud précédent et du nœud courant. La file s'arrêtera quand il n'y aura plus de nœuds voisins de disponibles.

Remarque : Le choix de l'angle de  $\pm 35^\circ$  vient d'une observation du comportement des cellules dans le bois. Il n'y a pas dans la nature des cellules d'une même file avec des variations supérieures à cet angle, car les files sont contraintes par les files adjacentes et les rayons.

Le choix du nœud voisin sera donc compris dans un certain angle. Dans le cas simple (ci-dessous), on trouve un voisin étant très proche de l'axe directeur, il sera alors choisi sans test supplémentaire.

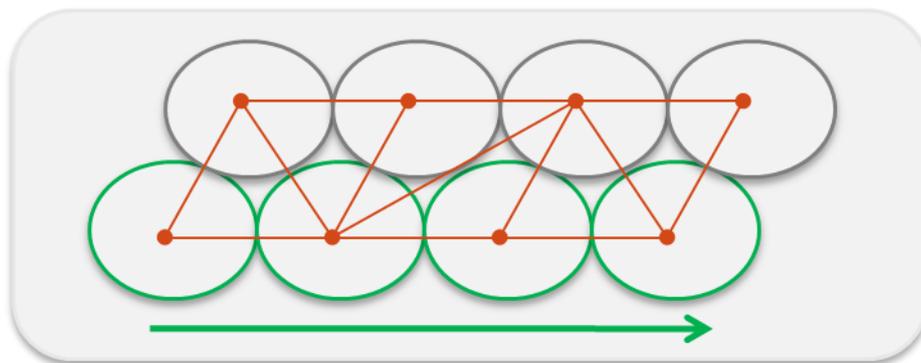


Figure 36. Alignement "simple"

Dans la réalité on va souvent trouver plusieurs voisins dans notre intervalle. Pour effectuer un choix nous regardons en premier lieu si les deux premiers voisins (N1 et N2) minimise l'angle le vecteur directeur. S'ils sont de part et d'autre de la droite formée par le nœud courant et le nœud précédant, on testera si la différence d'angle entre N1 et N2 est inférieure à 8°. Dans le cas où l'angle est supérieur à 8° on choisira celui qui a l'angle le plus faible. Dans l'autre cas on considèrera que les 2 voisins ont la même probabilité d'être dans la file. Pour différencier les deux voisins nous avons choisi de comparer les cellules voisines par rapport à la cellule courante et notamment comparer leur « ressemblance ». Lors d'une discussion avec un biologiste, nous avons évoqué notre désir de comparer des cellules ce qui n'est pas une chose aisée, il nous a alors spontanément suggéré d'utiliser la « *similarité de Bray Curtis*<sup>25</sup> » utilisé en écologie. Cette méthode a pour avantage de conserver les surfaces, le résultat est compris entre 0 et 1, où 1 représente 100% de similarité.

$$BC_{ij} = \sum \frac{|n_{ik} - n_{jk}|}{(n_{ik} + n_{jk})}$$

Figure 37. Formule de Bray Curtis

Avec  $n_{ik}$  la valeur de la surface de notre cellule courante et  $n_{jk}$  la valeur de la surface voisine à tester.

Grâce à cette méthode nous pouvons maintenant trouver quel voisin est le plus semblable à notre cellule courante. Nous choisirons la cellule qui minimise la distance et maximise la similarité (exemple ci-dessous). Une fois le nœud choisi on lui donnera un poids (somme de l'angle et de la similarité) qui représentera son score dans la file. L'ensemble des scores individuels des nœuds de la file donnera le score de la file.

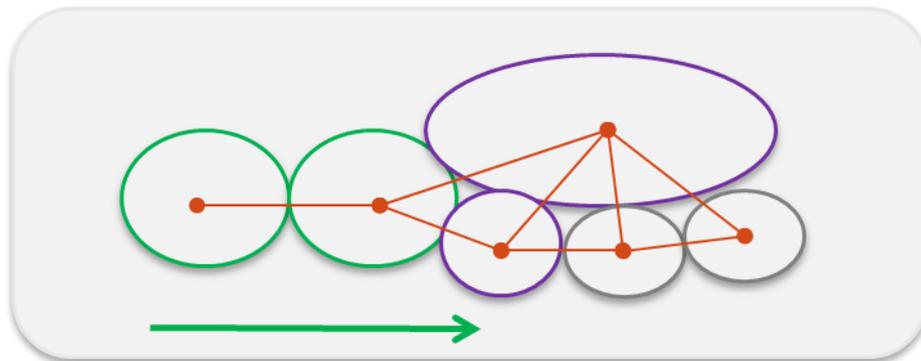


Figure 38. Alignement "Complexe"

Cette méthode permet de construire des files les plus justes possible en une itération, car il y a entre chaque cellule une cohérence au niveau de l'angle mais aussi au niveau de la ressemblance. Nous ne pouvons, néanmoins, pas certifier qu'il n'y ait pas un débordement sur une file voisine. Pour cela on va mettre en place une méthode de contrôle pour vérifier la véracité de nos files créées.

<sup>25</sup> Bray Curtis dissimilarity, [http://en.wikipedia.org/wiki/Bray\\_Curtis\\_dissimilarity](http://en.wikipedia.org/wiki/Bray_Curtis_dissimilarity)

### 2.3.2.3.2 RETRO CONTROLE

Comme nous venons de l'évoquer il se peut qu'une file déborde sur ce qui pourrait être une autre file qui ne serait pas encore créée (Cf. Figure ci-dessous). Pour cela j'ai pensé au processus biologique de rétro contrôle qui permet une action inverse en cas de problème.

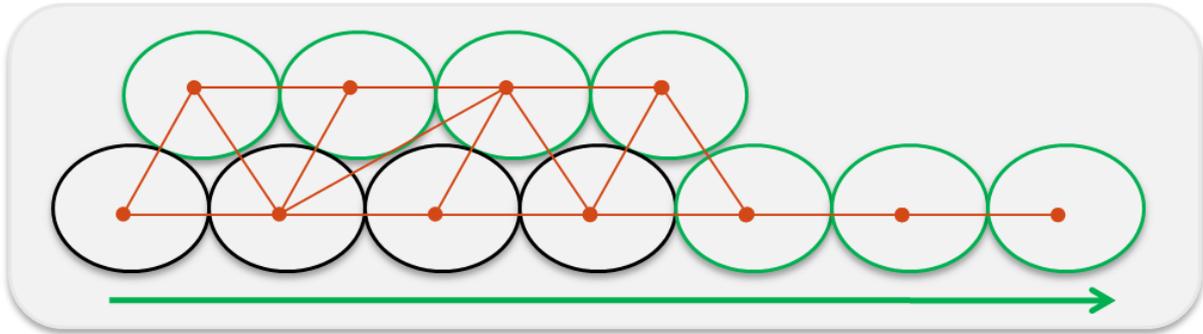


Figure 39. Débordement d'une file lors de sa création

Pour vérifier la cohérence de notre file, nous allons repartir en sens inverse à partir de la dernière cellule de la file. Pour cela on utilisera le vecteur principal de l'image dans le sens opposé. Si le parcours inverse retourne jusqu'à la cellule de départ alors la file bénéficiera d'une diminution de son score car nous cherchons à minimiser l'énergie des files correctes. Dans le cas où la reconstruction ne donnerait pas la même file, nous enlèverons la portion de file à partir du nœud à problème. Et on repartira du nouveau nœud de fin, jusqu'à arriver jusqu'au nœud de départ (Cf. Figure 40).

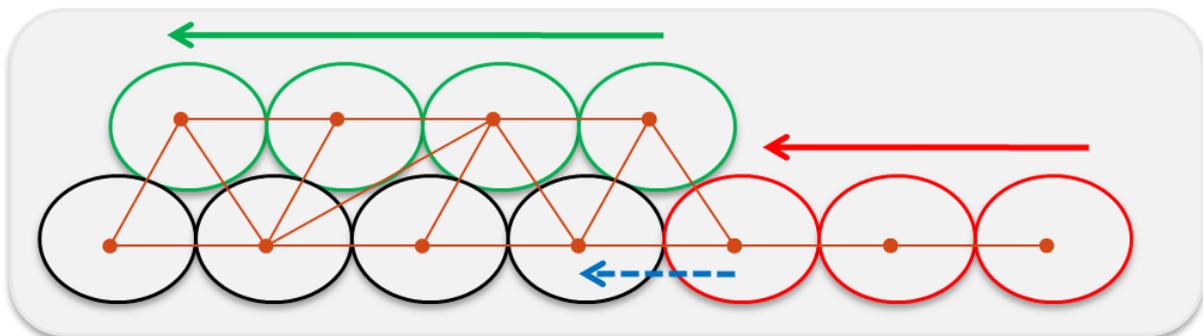


Figure 40. Parcours du rétro contrôle dans le cas d'un débordement de file

Quand une file a subi des modifications à la suite du rétrocontrôle, le score de la file sera diminué selon le pourcentage de nœud enlevé par rapport au nombre total de nœud de la file.

### 2.3.2.3.3 CHEVAUCHEMENT

Comme nous venons de le voir il se peut qu'une file déborde sur une autre file, mais dans les cas précédant les nœuds étaient « libres ». Nous allons maintenant voir comment sont gérés les cas où les nœuds sont déjà dans une file créée. Dans notre exemple ci-dessous la file A (en violet) est déjà construite et la file B (en vert) est en construction. Quand une file a comme voisin principal un nœud d'une autre file, on va comparer le score du nœud de la file A avec le score de la file B. Si le score de la file B minimise les scores globaux et le score local

(entre les deux nœuds voisins), on ne détruira pas la file A et on cherchera si la file B a d'autre candidat possible. Si la file B obtient un meilleur score, le file A sera tronquée au niveau du nœud. Les score de la file A seront alors remis à jour selon les nouveaux paramètres de la file et les nœuds libérés seront rajouter à la liste de traitement.

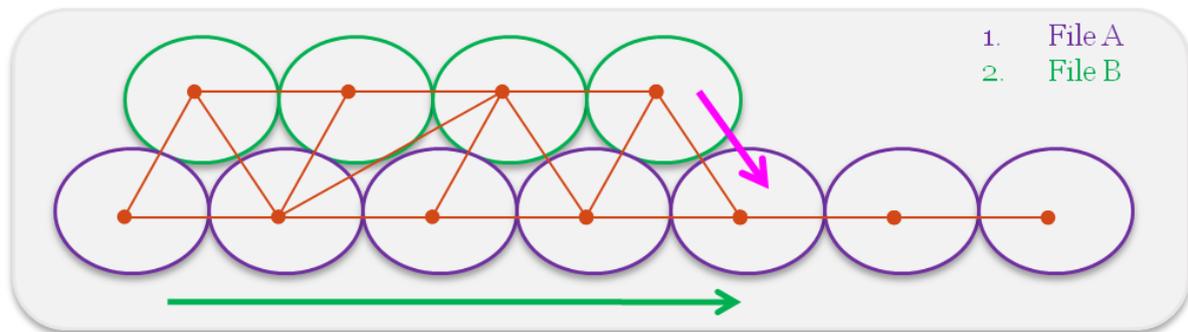


Figure 41. Gestion des chevauchements de files

Remarque : Les scores sont calculés grâce à l'angle et au coefficient de similarité de Bray Curtis.



Figure 42. Résultat des trois premières étapes de la détection des files cellulaires

Avec ces trois systèmes mis en place nous allons pouvoir créer des files et tronçons de file les plus « juste » possible. Comme on peut le voir sur l'image ci-dessus il serait intéressant de pouvoir raccorder les tronçons qui sont éclatés. Sur notre exemple la file en vert foncé et la file orange font très probablement partie d'une seule et même file. Lors du détourage fait par la LPE, la dernière cellule en vert et la première en orange ne sont pas considérées comme étant adjacente à un pixel près. Il faudra donc un traitement supplémentaire pour raccorder correctement nos tronçons.

#### 2.3.2.3.4 RACCORD

Nous allons maintenant voir comment reconstruire nos files à partir de tronçons. Pour cela nous allons découper le raccord en trois parties de complexité croissante :

- Le cas avec deux voisins
- Les cas avec trois voisins
- Les cas complexes

#### 2.3.2.3.4.1 LE CAS AVEC DEUX VOISINS

Si les deux files (D et C) sont toutes les deux strictement comprises entre les mêmes deux files (A et B), on peut alors affirmer que C et D sont issues d'une même file. Schéma ci-dessous.

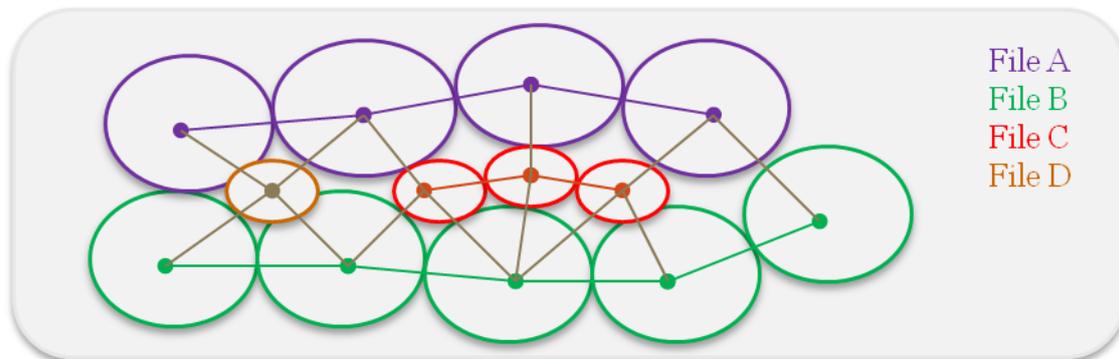


Figure 43. Schéma des files comprises entre deux files communes

#### 2.3.2.3.4.2 LE CAS AVEC TROIS VOISINS

Dans cette partie on peut distinguer deux sous parties. Dans le premier cas deux files différentes sont comprises entre les trois mêmes files, elles seront alors raccordées sans plus de test. Dans le second une file peut être encadrée par trois files (Cf. figure ci-dessous) dont deux qui vont être significative et une autre négligeable. Le but est de ramener une file encadrée par trois autres files à un cas simple à deux files voisines.

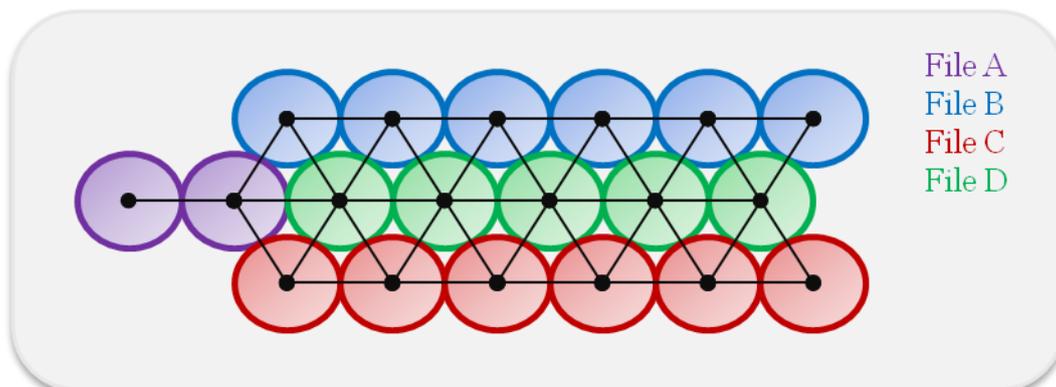


Figure 44. File comprise entre trois files voisines

Pour résoudre ce problème nous avons testé plusieurs méthodes (dont celle de Bray Curtis) pour trouver quelles sont les files les plus « similaires » à notre file d'étude (ici en vert). Plusieurs méthodes nous ont données de bons résultats mais nous devons faire attention à ne pas créer trop de cas ramenés à deux files pour ne pas créer de fausses files. Après discussion avec les biologistes nous avons choisi de mettre en place un arbre de décision basé sur des analyses biologiques. Nous allons comparer le nombre d'adjacences totales avec le nombre d'adjacences par file.

Dans notre exemple (Cf. figures 44 et 45) la file A aura le plus petit score car elle possède le moins d'adjacence avec notre file d'étude (en vert). Si le score est supérieur à 0,15 alors nous ne pourrons pas traiter plus notre cas, dans le cas inverse nous pourrons dire que la file A est négligeable par rapport aux autres files (B et C). Biologiquement ce seuil correspond bien au phénomène d'intrusion de cellules ou des files

*Adjacence par file*  
*Adjacence totale*

cellulaires, nous l'avons déduit après l'analyse, avec les biologistes, d'un grand nombre d'images. Nous ne retrouvons maintenant avec un cas à deux files, néanmoins pour éviter tout faux positif, nous allons vérifier que la similarité des deux files restant est quasiment identique. Pour cela nous allons tester si le rapport des scores est supérieur à 0,7. Si c'est le cas notre file d'étude sera considérée comme encadré par deux files et on la reportera au cas simple pour voir s'il n'y a pas une autre file avec la même signature d'adjacence. Dans le cas contraire nous ne la traiterons pas.

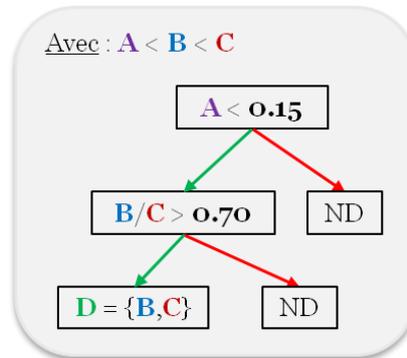


Figure 45. Arbre de décision pour les files avec trois files adjacentes

Le but de cette méthode n'est pas de traiter tous les cas à trois voisins, c'est pour cela que nous avons un arbre des décisions très restrictif, car nous avons choisi de laisser à l'utilisateur le choix de raccorder deux files que nous n'aurons pas traité, plutôt que de prendre le risque de produire de faux positif dans nos raccords.

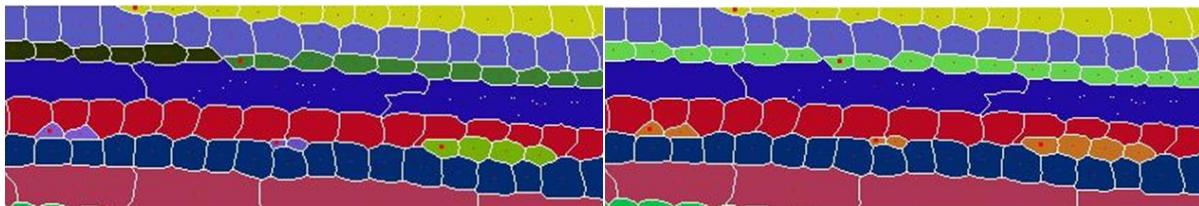


Figure 46. Résultat du raccord des files

#### 2.3.2.3.4.3 LES CAS COMPLEXES

Pour expliquer le dernier type de raccord nous allons devoir mettre en place une notion de qualité des files. J'ai choisi de mettre en place trois types de files ; *Good*, *Verified* et *Undefined*. A la construction toutes les files seront considérées comme *Undefined*. Avant de commencer le raccord, on effectue une vérification des *Good* files. Une file est considérée comme telle quand elle traverse de part en part l'image selon le vecteur directionnel. Cela nous permet de nous abstraire de la taille des files et ainsi de traiter les cas où les files sont des biais dans l'image et éviter les effets de bord. Les files ayant subi un raccord seront marquées comme *Verified*.

Le but de cette étape, va être de raccorder deux files ayant une *seule* file en commun. Cette file devra obligatoirement être marquée comme *Good*. Nous avons un exemple de ce type de raccord sur la figure 35. Nous allons raccorder la file B (en vert) avec la file C (en rouge). Pour cela nous allons vérifier en premier lieu qu'elles soient toutes les deux adjacentes à la même file *Good* et qu'elles soient du même côté. Dans un deuxième temps nous allons vérifier que les files adjacentes de nos files d'étude soient en contact avec notre file *Good*.

*Si B et C ⊂ A*  
*Et si A ⊂ {D, E}*  
*Alors B = C*

Si ces conditions sont remplies alors on peut considérer que les files C et B sont issues de la même file.

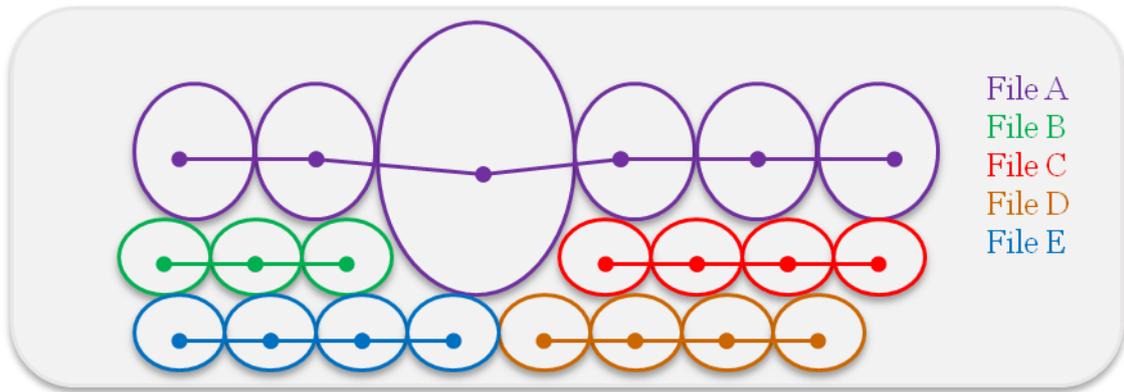


Figure 47. Raccord des cas complexes

Grâce aux méthodes de raccord nous obtenons des images avec la plus part des cas traités. Il est à noter qu'en parallèle du stage, un stage de L3 a été mené pour faire un logiciel d'édition et contrôle des files, le but étant que les biologistes puissent apporter intuitivement des modifications sur les files créées.

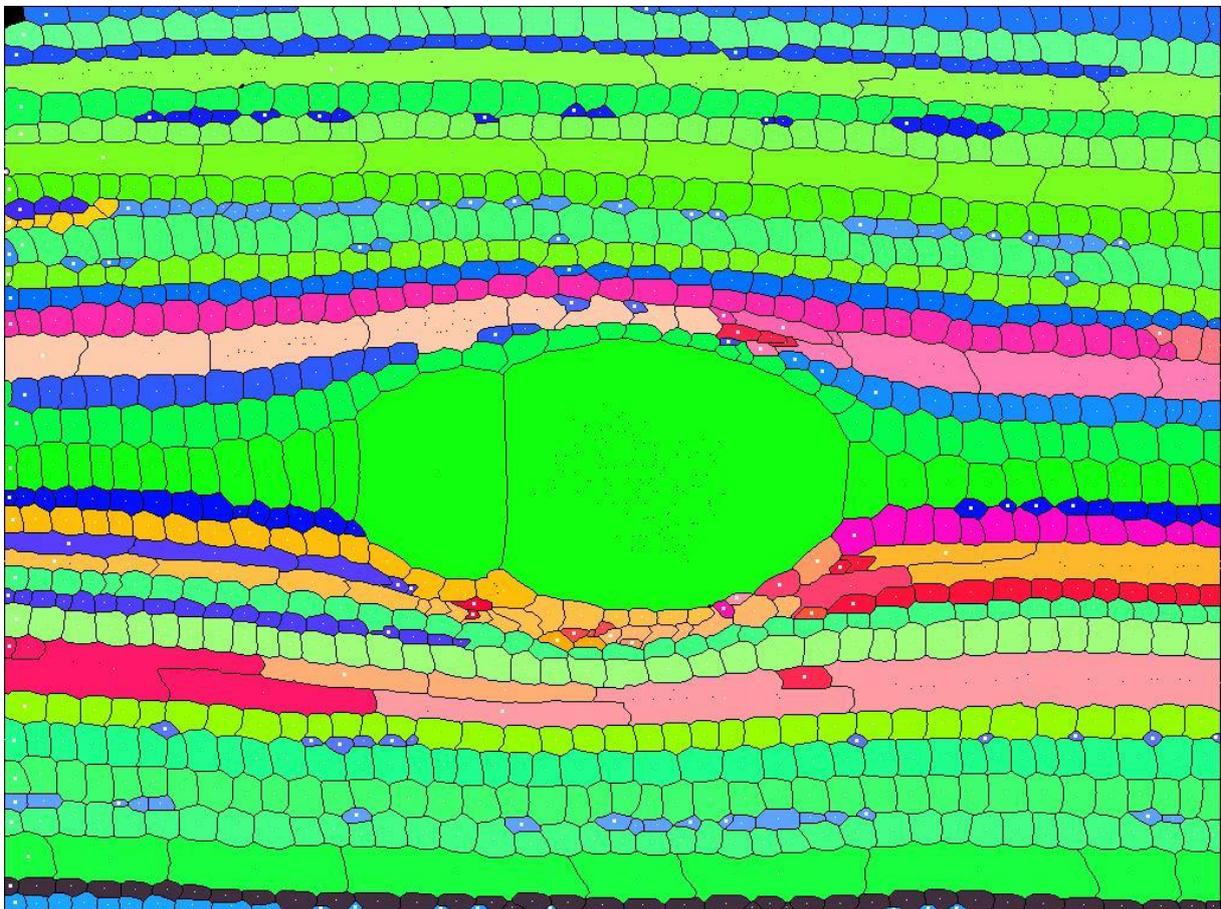


Figure 48. Résultat de la détection des files cellulaires sur l'image d'acajou

La coloration représente les différentes qualités des files comme nous l'avons expliqué dans le paragraphe ci-dessus. En dégradé de vert les files *Good*, en dégradé de bleu les files *Verified* et en dégradé de rouge les files *Undefined*.

#### 2.3.2.3.4.4 RESUME SCHEMATIQUE DE LA METHODE DE DETECTION DES FILES

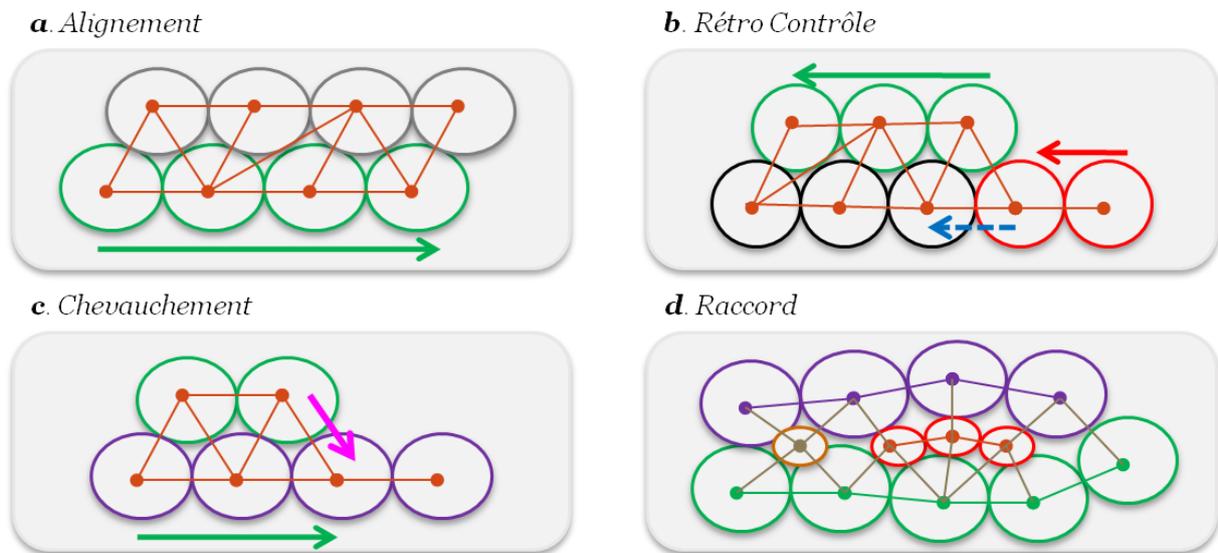


Figure 49. Schémas des différentes étapes de la détection des files cellulaires

### 2.3.3 TYPAGE DES CELLULES

Les files cellulaires étant déterminées, il reste à essayer de définir les types cellulaires. Comme nous travaillons chez les feuillus, nous aurons quatre typages possible : Rayon, Fibre, Vaisseau et Indéfini. Nous avons envisagé deux pistes pour la caractérisation, une méthode statistique par apprentissage avec jeu d'essai (data mining<sup>26</sup> avec weka [16]) ou une approche biologique avec construction d'un arbre de décision selon des critères issus d'une formalisation avec les biologistes.

Le data mining est souvent utilisé dans la littérature [8] pour notre problématique. Le principe de base est de déterminer les corrélations entre les  $n$  variables décrivant les régions, pour construire un arbre de décision. Cette méthode d'apprentissage est donc dépendante du jeu d'essai avec lequel nous allons construire l'arbre de décision. De ce fait, au vu de nos échantillons il est peu probable que cette méthode donne les résultats souhaités. En effet, si nous prenons des images avec une trop faible variabilité nous allons être trop restreint pour traiter l'ensemble des cas et dans le cas contraire avec une grande variabilité dans le jeu d'essai nous aurions un arbre trop permissif. Au vu de nos échantillons il est donc utopique de penser pouvoir tout traiter avec cette méthode. Une solution pourrait être de confronter les échantillons à un arbre de décision constitué grâce à des arbres de la même espèce (ou du même comportement de la variabilité biologique). Et ainsi créer une base de données d'arbre de décision, comme en génétique avec le logiciel BLAST<sup>27</sup>.

Notre autre piste consiste à créer un arbre de décision grâce à la formalisation des méthodes de détection faite naturellement par les biologistes quand ils observent une coupe de bois. Pour cela nous avons prévu une réunion prospective pour évaluer si cette méthode pourrait être envisageable.

C'est cette méthode dite de formalisation que nous avons retenue et qui va être maintenant détaillée.

<sup>26</sup> Data Mining : Exploration de données, aussi connue sous les noms fouille de données, a pour objet l'extraction d'un savoir ou d'une connaissance à partir de grandes quantités de données, par des méthodes automatiques ou semi-automatiques.

<sup>27</sup> BLAST (acronyme de basic local alignment search tool), <http://en.wikipedia.org/wiki/BLAST>

### 2.3.3.1 FORMALISATION

Dans le but de créer un arbre de décision avec des critères biologiques nous avons organisé une réunion avec les biologistes. Nous avons choisi un panel d'images assez varié pour ne pas créer un arbre trop spécifique à une espèce. Pendant cette réunion nous avons essayé de décrypter les processus déductifs qu'utilisent les biologistes pour typer les cellules. Nous avons déterminé en premier lieu que la taille (périmètre) des cellules permet de discriminer rapidement la plupart des vaisseaux, ensuite la circularité qui permet de différencier les rayons et enfin le rapport surface lumen/surface parois pour typer les derniers vaisseaux.

L'ordre dans lequel nous devons utiliser ces critères n'a pas été trouvé lors de la réunion, à cause de la variabilité des images, nous avons choisi de déterminer l'ordre empiriquement.

L'arbre de décision sera construit automatiquement selon les images. Il n'y aura aucune valeur fixe. Pour déterminer les seuils on va extraire l'ensemble des paramètres les un après les autres et on fera un *k-means clustering* d'ordre 2 pour diviser en deux groupes nos valeurs. Le *k-means clustering* est un outil de classification qui permet de répartir un ensemble de données en *k* classes homogènes.

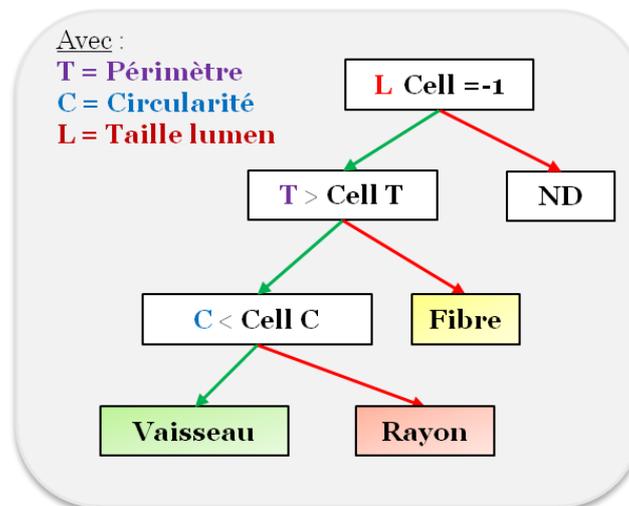


Figure 50. Arbre de décision pour le typage des cellules

Comme nous pouvons le voir dans le schéma ci-dessus, nous avons enlevé le paramètre « rapport surface lumen/parois » car il n'était pas déterminant pour l'ensemble de nos images. Le premier test permet d'enlever les « fausses » cellules, qui sont en fait des artefacts de la segmentation. Ensuite le périmètre va nous permettre de séparer les fibres des autres structures et enfin le périmètre va discriminer les vaisseaux (très circulaire), des rayons (de forme rectangulaire).

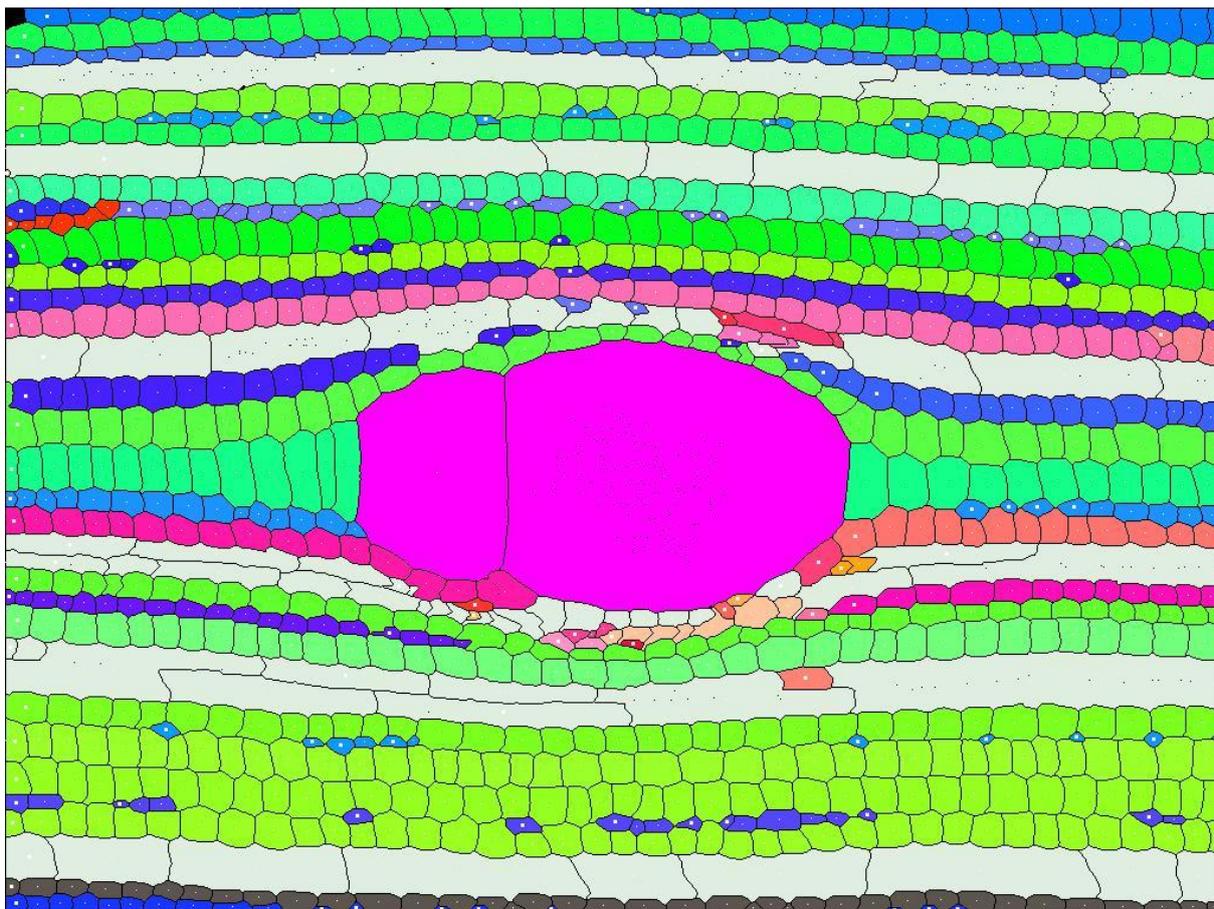


Figure 51. Résultat du typage des cellules

Le code couleur vu précédemment a été conservé au niveau de file. On a rajouté pour vérifier, avec les biologistes, deux autres couleurs ; en blanc les rayons et en violet les vaisseaux. Lors de la visualisation finale, ces deux couleurs ne seront pas retenues car les biologistes ont choisis de visualiser la qualité des files plutôt que les types cellulaires.

## 3 RESULTATS

L'intégration de notre chaîne de traitement dans ImageJ étant décrite, nous allons exposer les résultats ainsi obtenus, sous un aspect qualitatif puis quantitatif. De plus, il sera intéressant de discuter des limites de la nouvelle méthode créée et de prendre l'avis des experts, et donc, d'ouvrir de nouvelles perspectives amélioratrices du projet Ficelef. *Annexe 6.*

### 3.1 INTEGRATION SOUS IMAGEJ

La solution précédemment décrite a été implémentée et insérée sous forme de plugin dans ImageJ. L'API d'ImageJ propose facilement l'intégration de nouvelles fonctionnalités grâce à des interfaces prévues à cet effet. De même de nombreuses fonctionnalités sont utilisables quant à la manipulation d'image, d'opérateurs, de filtre, etc.

L'API d'ImageJ est construite autour de neuf packages, ce qui représente 283 classes Java et un nombre incalculable de méthodes associées.

La création d'un plugin se fait simplement en implémentant l'interface *ij.plugin.filter*. Cette interface implique l'implémentation d'une fonction *run* et d'une fonction *setup*. Ces fonctions sont utilisées au lancement du processus lié au plugin. Ils permettent entre autre de vérifier si l'image est compatible avec la méthode que nous avons créé.

Voici le corps d'une classe « hello world » permettant la création de plugin :

```
import ij.*;
import ij.plugin.filter.PlugInFilter;
import ij.process.ImageProcessor;

/**
 * Basic class as plugin for imageJ
 */
public class Hello_ implements PlugInFilter {

    @Override
    public void run(ImageProcessor ip) {
        IJ.showMessage("Hello World en Java pour ImageJ !");
    }

    @Override
    public int setup(String arg, ImagePlus imp) {
        // TODO Auto-generated method stub
        return 0;
    }
}
```

On remarquera les imports nécessaires à ce plugin concernant les classes *ImageProcessor* et *ImagePlus* brièvement décrites ci-dessous :

La classe *ImageProcessor* est une classe abstraite du package *ij.process* permettant la manipulation d'image de type byte, short, float, et RGB. . Le diagramme de classe du package *ij* est présent en annexe 10. Les objets *ImageProcessor* et *ImageStack* contiennent les données actuelles des pixels des images, et piles d'images (dans l'ordre). Ils sont utilisés pour traiter et convertir l'image.

La classe *ImagePlus* est une classe du package *ij* portant toute les informations et données d'une image. Le diagramme de classe du package *ij* est présent en annexe 10. Elle étend les fonctionnalités de la classe java `java.awt.image` comme le montre la figure 52, permettant d'ouvrir, stocker et afficher une image (une pile d'image). Elle permet aussi son affichage à l'écran.

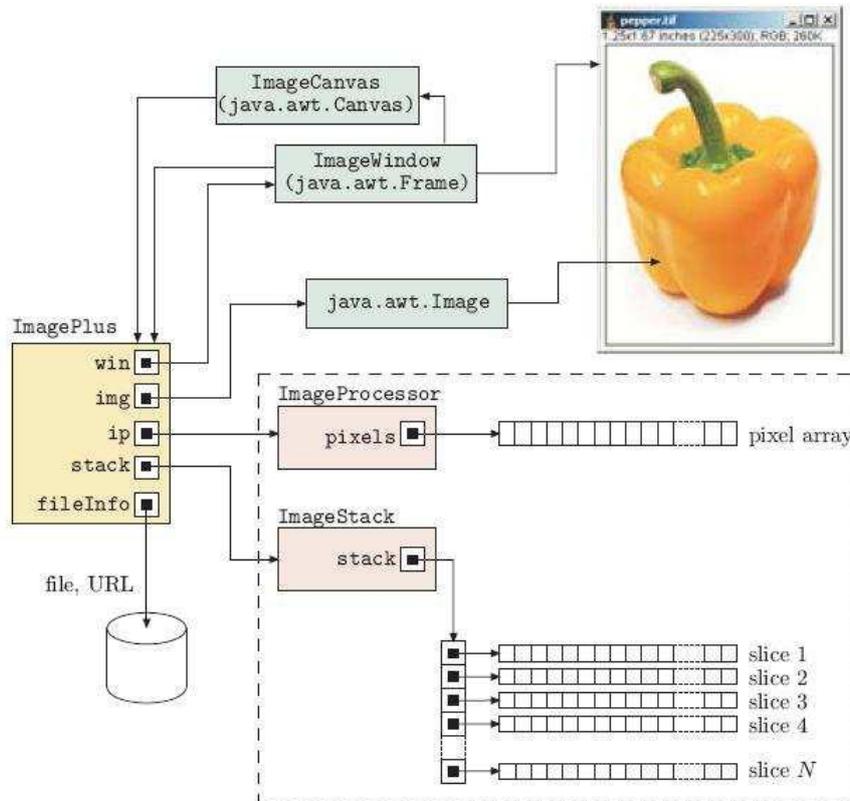
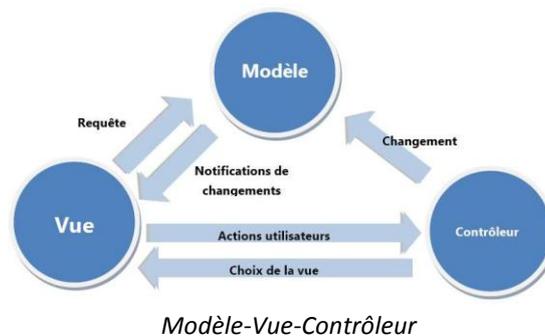


Figure 52. Représentation simplifiée d'une image (ou piles d'images) dans ImageJ

L'intégration de nos classes et méthodes sous forme de plugin nous impose une structure de type MVC<sup>28</sup>. Nous avons déjà respecté ce type de model en aval. Nous n'avons donc pas eu de problème pour franchir cette étape.

Le Modèle-Vue-Contrôleur (en abrégé MVC, de l'anglais Model-View-Controller) est une architecture et une méthode de conception qui organise l'interface homme-machine (IHM) d'une application logicielle. Ce paradigme divise l'IHM en un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l'interface.



<sup>28</sup>Le MVC (Modèle Vue Contrôleur), <http://fr.wikipedia.org/wiki/Mod%C3%A8le-Vue-Contr%C3%B4leur>

## 3.2 RESULTATS OBTENUS

Le stage a permis de réaliser un plugin s'insérant dans l'API d'ImageJ. Il est donc d'ores et déjà utilisable par un utilisateur en allant simplement sur le menu plugin du logiciel et en choisissant Ficelef (annexe 6). Ces résultats sont sous la forme d'une sortie graphique présentant les files colorées sur l'image traitée, et d'une sortie texte regroupant les résultats numériques : une sortie tableur compatible est fournie comportant les caractéristiques des files choisis par les biologistes.

### 3.2.1 RESULTATS

#### 3.2.1.1 BON RESULTAT

##### ❖ Qualitativement

La représentation des files permet d'une part de contrôler visuellement leur parcours, d'autre part de choisir les plus pertinentes à une étude statistique. Le code couleur que nous avons expliqué dans la présentation a été validé et conservé par les biologistes. En Vert les files taguées *Good*, en Bleu les files *Verified*, en Rouge les files *Undefined* et en Blanc les *Rayons*. Ceci permet de qualifier la fiabilité d'une file, en effet une file ayant subit beaucoup de raccords a plus de chance d'être fausse comparativement à une sans raccords. Les résultats obtenus dans notre panel test sont présentés ci-dessous :

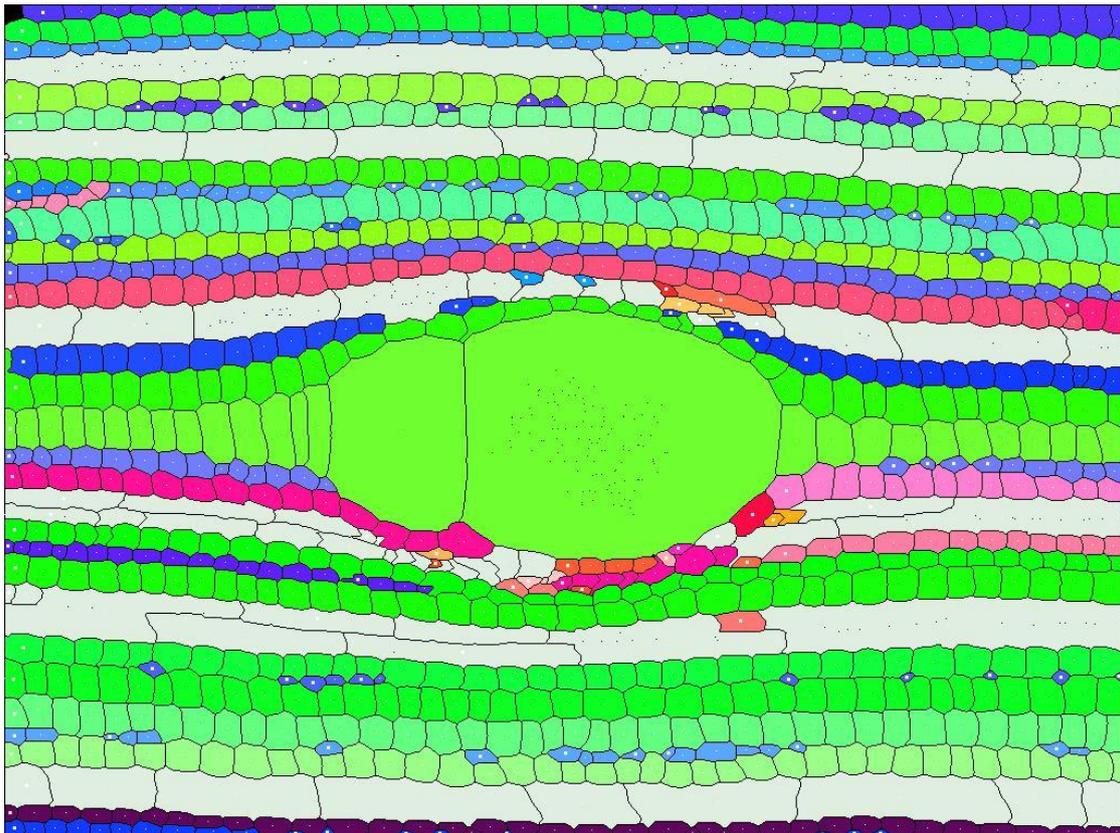


Figure 53. Résultat de Ficelef sur l'Acajou

**Panel d'images :**

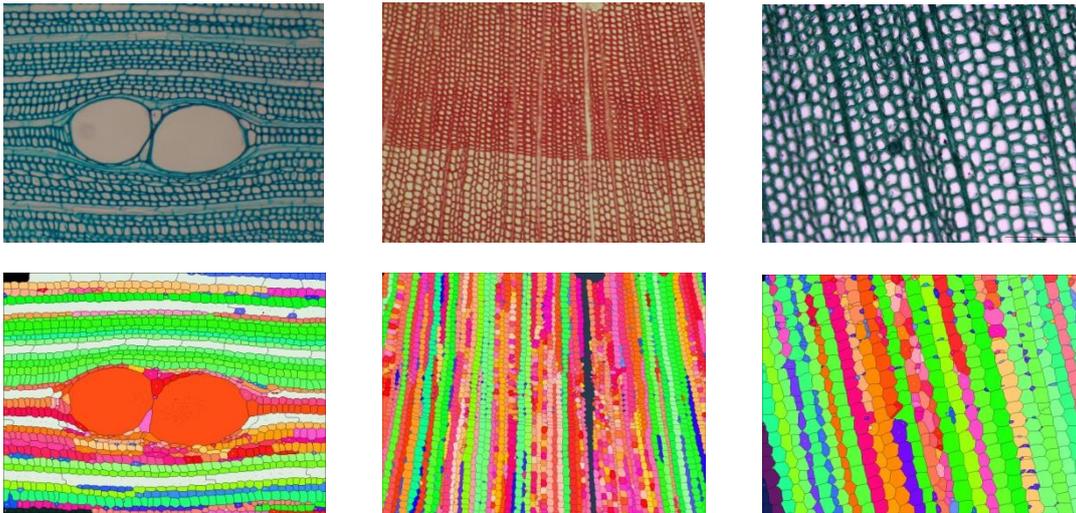


Figure 54. De gauche à droite : Acajou, Pin Noir et Sapin

La méthode donne de bon résultat avec des colorations différentes, des angles différents, de zoom (dans une certaine mesure) et surtout des structures différentes, en effet ici le sapin est une espèce de la famille des résineux. On peut aussi noter que la méthode est reproductible. En effet on retrouvera toujours les mêmes files de créées.

❖ **Quantitativement**

Au delà des données brutes des files, il a fallu quantifier la qualité d'une file. Nous nous sommes basés sur deux critères principaux : le score de la file et sa labellisation.

**Score et labellisation :**

La labellisation faite à partir de la longueur est un critère significatif car à priori les files cellulaires traversent toute l'image; se basant sur cette information, on peut qualifier les files petites comme moins fiables : il s'agira d'ailleurs le plus souvent de tronçons parasites, de défauts de raccord.

**Données brutes :**

Les données utiles permettant à l'expert de caractériser les files cellulaires ont été définies en accord avec leurs attentes et sont listées ci-dessous :

- surface du lumen
- ratio de surface lumen/paroi
- circularité du lumen

Avec les indices cités précédemment, l'expert pourra choisir les files jugées les plus fiables à inclure dans son étude statistique.

**Temps d'exécution :**

L'expert participant à la réalisation du projet a passé six heures à l'annotation de trois images, soit deux heures de traitement manuel par image. Le plugin Ficelef passe en moyenne une minute par image (variable en fonction de la résolution de l'image). Se basant sur le fait qu'une erreur commise par l'algorithme est corrigible en une minute par l'opérateur, on peut considérer que le plugin reste plus rapide qu'une annotation manuelle en dessous d'une centaine d'erreurs par image.

### Réussite de l'identification :

Il est difficile d'évaluer la quantité de files détectées par notre traitement en accord avec l'identification experte. Une estimation grossière du pourcentage de réussite du plugin à ce stade tourne autour des 40 à 60% sur les « belles images » et nettement moins sur les images dites complexes.

#### 3.2.1.2 LIMITES

Nous allons d'ailleurs maintenant voir qu'elles sont les limites de nos méthodes. Comme nous l'avons précisé plus haut nous sommes conscients qu'on ne pourra pas traiter tous les cas, notamment les cas complexes.

Les cas dit complexe sont par exemple la transition entre le bois d'été et le bois d'hiver chez certaines espèces. En effet sur ces images on observe sur le bois d'hiver une inversion du contraste des lumens (en noir) par rapport aux lumens des cellules du bois d'été (en blanc). Cf. figure ci-dessous.

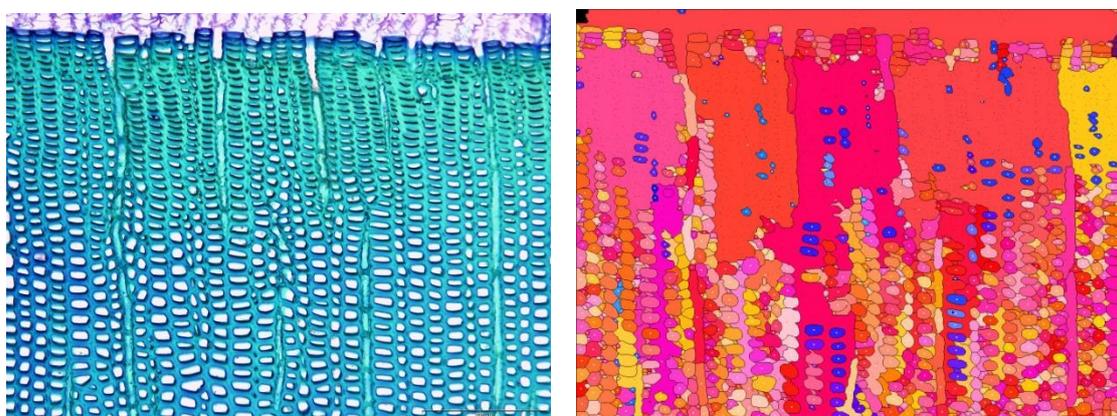


Figure 55. Coupe de Sapin avec transition été/hivers et résultat de ficelef

Notre méthode est largement basée sur les différences de contraste. On peut voir que l'algorithme de la LEP n'arrive pas à détourer correctement ce genre d'image. Les traitements post LEP ne peuvent pas gérer des cas si opposés.

Comme nous l'avons dit dans la partie prospective des méthodes, on ne traite pas les images non colorées. L'avantage de ces images se trouve coté humain. En effet prendre en photographie une coupe d'arbre sans la colorer réduit considérablement le temps de l'opération. Une coloration peut prendre jusqu'à 45min, alors qu'ici la prise serait directe. Nous avons néanmoins trouvé des pistes pour traiter ces images pendant nos études, mais la mise en place des méthodes aurait pris trop de temps, nous avons donc choisi de nous consacrer qu'aux images colorées. Le principe du traitement aurait été d'inverser localement les couleurs de l'image (le liseré noir autour des lumens) pour retrouver une cinétique identique à celle des images colorées.

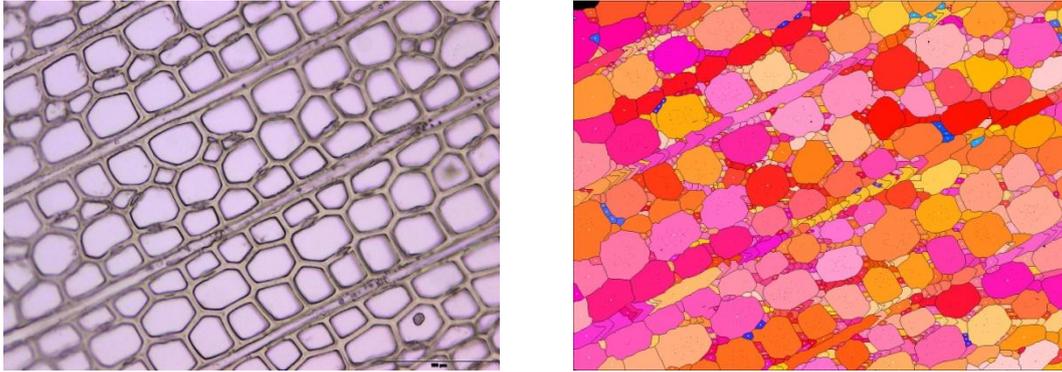


Figure 56. Coupe de Sapin non colorée

La méthode de parcours que nous avons mis en place permet de gérer les courbures des files, notamment aux abords des vaisseaux. Néanmoins si les variations autour des vaisseaux sont trop importantes, l'algorithme ne pourra les épouser. De plus, l'algorithme créera un grand nombre de tronçons difficilement raccordables, voire pas raccordables. Plusieurs solutions ont été envisagées par exemple, utiliser les rayons pour compter le nombre de files de part et d'autre des vaisseaux et ainsi pouvoir les raccorder, mais les discussions avec les biologistes ont montrés que cette hypothèse n'a pas encore été vérifiée. Et que justement le projet Ficelef pourrait servir à vérifier ce genre d'hypothèses. Une autre solution consisterait à créer des modèles de droites déformables pour se coller aux files.

Pour vérifier la robustesse de notre algorithme nous avons choisi de faire pivoter notre image test d'acajou. Et nous n'avons pas (à notre grande surprise) retrouver les mêmes résultats, notamment au niveau du détournage des bassins. Après avoir fait une étude sur chaque étape pour voir d'où cela pouvait provenir, nous avons vu que l'algorithme de la LEP induisait ces différences. Cela vient peut-être de l'implémentation de la LEP que nous avons choisi et notamment de l'initialisation des graines de départ (même si elles sont censées être constantes). Comme nous l'avons vu précédemment le filtre flou peut produire des différences, néanmoins ces différences sont « négligeables » dans un premier temps, car elles ne sont pas systématiques et sont résiduelles.

### 3.3 DISCUSSION

Nous allons discuter ici des résultats que nous avons obtenus. Nous avons réussi à paramétrer beaucoup de seuils de filtres automatiquement, ce qui fait que pour l'utilisateur le plugin reste assez simple d'utilisation. Le paramètre restant n'a pas été mis en dur, même si empiriquement nous avons toujours le même seuil. Mais il se peut que quand l'image dépasse une certaine résolution ce paramètre change. Pour cela l'utilisateur pourra changer cette valeur. Un fichier d'aide sera donné pour comprendre comment régler ce paramètre informatique avec une mise en correspondance biologique pour une meilleure compréhension.

Certaines erreurs ou cas non traités, ont été faites par le plugin au regard des images expertes. Nous ne reviendrons pas sur toutes ces erreurs mais en exposerons les plus intéressantes.



Figure 57. Zone à problème. En noir le tracé expert

Notamment dans certains cas de figures assez récurrents que sont le choix de la bonne file en cas de bifurcation, de discontinuité, de raccord manquant, etc.

Nous avons choisi de limiter les faux positifs, donc la plupart des « erreurs » faites par le plugin sont laissées à l'appréciation des experts. Les rares cas où de faux positifs sont créés proviennent de la LPE et non de la création des files. Le problème de l'amélioration de l'image avant un traitement reste ouvert. Il n'y a pas de solution unique à ce problème. Celle proposée est efficace dans un grand nombre de cas et accepte une grande hétérogénéité des images. C'est pour cela que nous l'avons conservée malgré l'apparition de certaines « erreurs » ponctuelles. Ces « erreurs » sont d'ailleurs souvent dues à la grande variabilité biologique.

Dans le cas de droite (cf. figure 57, image de droite), on observe une fourche (embranchement entre deux files). Notre algorithme choisit son chemin principalement en se basant sur des directions à respecter et la similarité entre cellules et cherche à minimiser les différences. L'expert considère que la branche à choisir est celle comportant les plus fines. Ce point reste donc à intégrer au suivi des files.

La rapidité de la méthode reste un grand atout, tout en ayant de meilleurs résultats (chez les feuillus), ou identique chez les résineux. En effet en nous servant de filtre de bas niveau à la place de filtre de convergence comme le MeanShift, nous avons considérablement raccourci les temps de calcul (plus de la moitié), tout en restant très efficace.

Nous avons aussi pu mettre en avant que le typage des cellules n'était pas obligatoire pour la création des files cellulaires. Nous avons ainsi limité les biais pouvant être induits lors de cette étape et ce que ce soit d'un point de vu informatique où les corrélations entre cellules restent dures à différencier, ou d'un point de vu biologique où deux experts ne typeront pas de la même façon les cellules.

Le travail que nous avons mené sur le passage d'une image couleur (32bit) à une image en niveau de gris (8bit) n'est pas entièrement résolu pour l'instant. Mais cela semble une étape indispensable dans le traitement de l'image. En effet un passage linéaire n'est pas suffisant pour maximiser les différences.

Lors des études et discussions avec les biologistes nous avons ouvert de nouvelles hypothèses biologiques comme le nombre de files entre les rayons. Notre plugin pourra justement servir à vérifier ce type d'information.

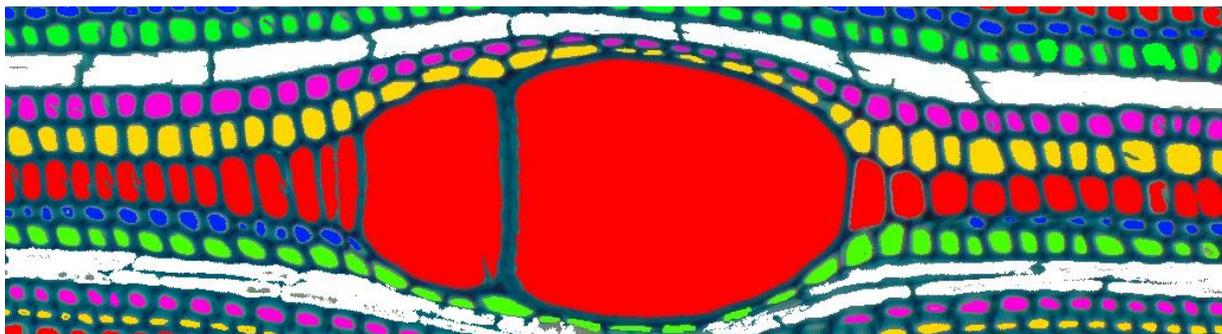


Figure 58. Coloration par un expert sur une image d'acajou

La labellisation des cellules et des files facilite grandement le travail des experts, car il permet d'écartier facilement les cellules à « problèmes » pour les études statistiques. Le code couleur permet, lui, une meilleure visualisation des résultats et permet de retoucher plus rapidement les files car plus facilement identifiables.

Malgré les cas non traités des files, la majorité des files extraites est tout de même d'ores et déjà exploitable. Et les indices de confiance établis permettent sans aucun doute de retrouver facilement ces files.

## 3.4 PERSPECTIVES

Nous pouvons, avec le recul que nous avons sur la détection de files, distinguer des perspectives à court terme comme à long terme.

### 3.4.1 PERSPECTIVES A COURT TERME

Le plugin doit maintenant être déployé en temps que plugin ImageJ. Il ne sera pas mis en tant que tel, mais inclus dans le projet Toaster (*Tree and planT Organs And Structures analyzER*), projet de plus grande envergure sur le traitement d'images numériques biologiques en cours de développement au laboratoire. Pour cela j'ai respecté au mieux les contraintes liées à l'intégration dans une nouvelle structure.

Dans un second temps, il y aura la réalisation d'un article avec les biologistes, dans une revue informatique et/ou dans une revue biologique.

### 3.4.2 PERSPECTIVES A LONG TERME

Dans notre méthode il reste deux paramètres que l'utilisateur doit renseigner. Idéalement il faudrait arriver à une application type « *push bouton* », c'est-à-dire avec des paramètres automatiques.

Le traitement de la transition du bois d'été et du bois d'hiver reste l'un des enjeux majeurs pour le traitement de l'image. Une des pistes envisagées est d'identifier la jonction bois d'hivers/bois d'été (par clustering par exemple) et de traiter séparément les deux parties de l'image avant la détection des files. La partie bois d'hiver se comporte comme les images non colorées que nous n'avons pas traitées.

Nous avons travaillé sur des images colorées. Il faudra maintenant traiter les images non colorées coupées ou poncées. En effet cela permettrait au biologiste un gain de temps pour la préparation des échantillons et notamment avec les images issues du ponçage la mise en place d'un bras robot pour poncer et photographier à la volée toutes les coupes d'un tronc.

Nous avons trouvé des pistes pour traiter les images non colorées mais nous n'avons pas eu le temps de les vérifier. Cependant les tests effectués semblent très prometteurs.

La mise en place d'un bras robot pourrait servir à photographier avec un contrôle très poussé toute la superficie du tronc en chevauchant chaque prise de vue pour ensuite le reconstituer. Pour cela il faudra gérer le pavage des images. C'est-à-dire une superposition des parties identiques dans les images pour reconstruire l'intégralité de la zone coupée (tronc ou branche).

Ces méthodes sont très gourmandes en image. Pour reconstituer un tronc de faible circonférence il y aura un très grand nombre de photographies. Une des pistes envisagée pour traiter autant d'image en un temps raisonnable est de paralléliser le code, en passant notamment par la programmation des processeurs de la carte graphique. Cela offrirait d'énormes possibilités encore inexploitées. Nous avons pendant le projet regardé la faisabilité de cette technique. Et le traitement d'image se prête très bien à cet exercice. Nous ne l'avons pas mis dans le projet par manque de temps, de matériels complémentaires adéquats.

## 4 CONCLUSION

Le projet Ficelef concernant la reconnaissance automatique de files cellulaires chez les Feuillus aura su capter toute mon attention durant ce stage de Master 2 bioinformatique et fait naître un fort intérêt pour le travail de traitement de l'image. Il a été le produit original d'applications et de développements informatiques en traitement d'images sur un sujet aujourd'hui indispensable à la compréhension du développement et de la plasticité des plantes, et ce au sein du laboratoire aujourd'hui reconnu dans ce domaine qu'est l'unité mixte de recherche « botanique et bioinformatique de l'Architecture des Plantes ».

Le sujet a été traité suivant une démarche de recherche encadré par un ingénieur de recherche en mathématiques appliquées à l'informatique au sein de mon laboratoire d'accueil (AMAP) ainsi que par un chercheur du CNRS spécialiste en traitement d'images appartenant à l'équipe d'imagerie du Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM).

Cette collaboration a permis la réalisation d'un plugin écrit en Java s'insérant dans un logiciel libre de traitement d'images intitulé ImageJ. La solution fournie a d'abord abordé la problématique par un travail de segmentation automatique des cellules de bois présentes sur les images microscopiques à traiter. S'en suit une approche de modélisation géométrique et topologique pour la reconnaissance des files en parcourant le graphe d'adjacence servant de structure de données aux régions segmentées. Et pour finir une étape de classification des types cellulaires.

Pour ma part, ces six mois m'ont permis de progresser dans bien des domaines aussi bien techniquement qu'humainement. La diversité des méthodes employées dans la réalisation de ce plugin, au delà des progrès en développement Java, m'ont permis de d'accroître mes compétences en traitement d'images, algorithmiques, mathématiques, statistiques, mais aussi en architecture végétale. De plus j'ai eu l'occasion de côtoyer bon nombre de chercheurs aux domaines d'applications très variés et aux compétences complémentaires, de participer à des conférences et modules sur le traitement d'image et la bioinformatique au sens large, me permettant d'insérer au mieux mon domaine de compétences encore jeune qu'est la bioinformatique.

# BIBLIOGRAPHIE

- [1] Monica Marcuzzoa. Automated Arabidopsis plant root cell segmentation based on SVM classification and region merging. *Computers in Biology and Medicine* 39 (2009) 785-793.
- [2] Benoit Roue, Patrick Bas, Nicolas Le Bihan. Décomposition et codage hypercomplexes des images couleur. 25 juin 2003.
- [3] Vincent L, Soille P. Watershed in digital spaces, an efficient algorithm based on immersion simulation. *Trans. PAMI* vol 13, n° 6, jun 91.
- [4] Meyer F & all. *Skeletons in digital spaces. Image analysis and mathematical morphology, theoretical advances*. Serra. Academic press.1988.
- [5] Ronald J. & Leanne B. A Graph-Based segmentation of wood Micrographs. CSIRO Division of Mathematics and Statistics.
- [6] Nicola Ritter, James Cooper. Segmentation and border identification of cells in images of peripheral blood smear slides. Australian Computer Society, Inc. 2007.
- [7] M.K. Mattias, M. Fujita, Fourier Transform Methods In Image Analysis Of Compression Wood At The Cellular Level, *IAWA Journal*, 25 (3), 311–324, 2004
- [8] M.K. Moël, L. Donaldson. Comparison of segmentation method for digital image analysis of confocal microscope images to measure tracheid cell dimensions, *IAWA Journal*, Vol. 22 (3), 2001: 267-288.
- [9] S. Bandyopadhyay and U. Maulik. Genetic clustering for automatic evolution of clusters and application to image classification. *Pattern Recognition*, Volume 35, Issue 6, June 2002, Pages 1197-1208.
- [10] D. Tseng and W. Shieh. Plume extraction using entropic thresholding and region growing. *Pattern recognition*, A. 1993, vol. 26, n° 5, pp. 805-817.
- [11] W. Ben ; Y. Zhang. Region based segmentation of QuickBird multispectral imagery through band ratios and fuzzy comparison. *ISPRS journal of photogrammetry and remote sensing*. A. 2009, vol. 64, n° 1, pp. 55-64
- [12] P. Kennel, G. Subsol, M. Guérout and P. Borianne, Automatic Identification of cell Files in Light Microscopic Images of Conifer Wood, *IEEE IPTA'10*, 8-12 July 2010, pg 150-156
- [13] P. Kennel. Rapport de stage M2 en BioInformatique, Projet Ficeler
- [14] <http://www.regentstruments.com/>
- [15] <http://rsbweb.nih.gov/ij/>
- [16] <http://www.cs.waikato.ac.nz/ml/weka/>
- [17] <http://jung.sourceforge.net/doc/index.html>
- [18] <http://www.cellprofiler.org/>

# TABLE DES FIGURES

Figure 1. Localisation de l'AMAP dans le monde .....	3
Figure 2. Eléments structurants chez le résineux.....	7
Figure 3. Coupe Transversale d'Acajou .....	8
Figure 4. Cellule d'acajou .....	9
Figure 5. Détection des files cellulaires sur une coupe d'Acajou .....	11
Figure 6. Interface d'ImageJ .....	13
Figure 7. Interface du Redmine.....	14
Figure 8. Compte rendu "Image Non Colorée" .....	14
Figure 9. De gauche à droite: P. caribensis (résineux coloration carmine rouge), Chêne vert (feuillus), B. utiles (feuillus), Sapin (résineux, coloration bleu de toluidine, x400), P. Acajou (feuillus) et F. Angustifolia (feuillus) .	15
Figure 10. Reconnaissance de files sur l'Acajou .....	16
Figure 11. De gauche à droite : Teinte, Saturation, Brillance. Sur Acajou .....	19
Figure 12. Application d'un médian avec $v=0$ (1), $v=2$ (2), $v=6$ (3) et $v=10$ (4). Sur Acajou .....	20
Figure 13. Soustraction de deux images .....	20
Figure 14. Résultat d'un flou, taille du masque = 3.....	21
Figure 15. Histogramme en niveau de gris de l'image d'acajou.....	22
Figure 16. LEP entre deux bassins .....	23
Figure 17. De gauche à droite : Profil d'une coupe non coloré, Profil d'une coupe coloré. Chez le sapin.....	24
Figure 18. Chaîne de traitements établie pour le plugin ficelef.....	25
Figure 19. Cellule à problème .....	26
Figure 20. Fragment de l'image d'Acajou, utilisé pour nos tests .....	27
Figure 21. Résultat du filtre Image des détails.....	28
Figure 22. Résultat du filtre Laplacien par les flous .....	29
Figure 23. Schéma et résultat de la soustraction .....	30
Figure 24. Résultat du filtre médian.....	30
Figure 25. Relief topographique de notre partie de l'image d'acajou .....	31
Figure 26. Illustration d'une surface topographique .....	32
Figure 27. Illustration de l'immersion d'une surface topographique .....	32

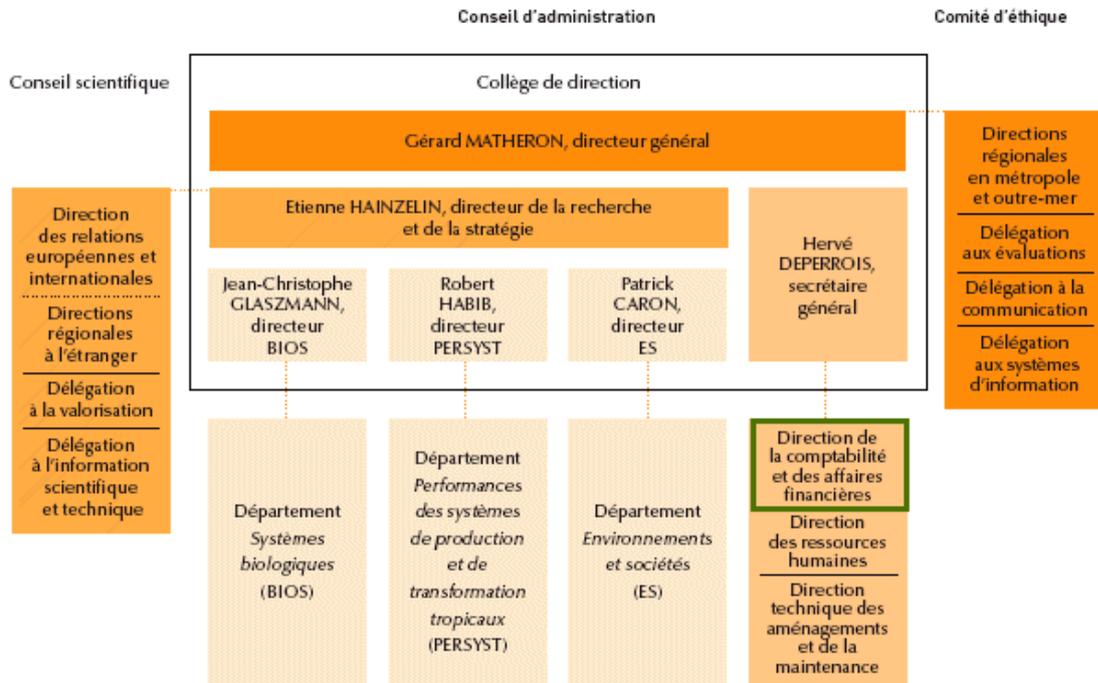
Figure 28. Résultat de l'algorithme LEP sur une portion de l'image d'acajou .....	33
Figure 29. Superposition des lignes de partage des eaux avec l'image native .....	33
Figure 30. Résultat de caractérisation des arrêtes et sommets.....	34
Figure 31. Profil d'une arête de sur-segmentation .....	34
Figure 32. Morphologie paroi végétale.....	35
Figure 33. Traitement de l'image détail des filtres .....	35
Figure 34. Superposition du graphe d'adjacence et des bassins "nettoyé" .....	37
Figure 35. Distribution des angles formés par les arêtes avec l'axe des abscisses sur l'image d'acajou .....	38
Figure 36. Alignement "simple" .....	40
Figure 37. Formule de Bray Curtis.....	41
Figure 38. Alignement "Complexe" .....	41
Figure 39. Débordement d'une file lors de sa création.....	42
Figure 40. Parcours du rétro contrôle dans le cas d'un débordement de file .....	42
Figure 41. Gestion des chevauchements de files.....	43
Figure 42. Résultat des trois premières étapes de la détection des files cellulaires.....	43
Figure 43. Schéma des files comprises entre deux files communes .....	44
Figure 44. File comprise entre trois files voisines .....	44
Figure 45. Arbre de décision pour les files avec trois files adjacentes.....	45
Figure 46. Résultat du raccord des files .....	45
Figure 47. Raccord des cas complexes .....	46
Figure 48. Résultat de la détection des files cellulaires sur l'image d'acajou .....	46
Figure 49. Schémas des différentes étapes de la détection des files cellulaires .....	47
Figure 50. Arbre de décision pour le typage des cellules.....	48
Figure 51. Résultat du typage des cellules .....	49
Figure 52. Représentation simplifiée d'une image (ou piles d'images) dans ImageJ.....	51
Figure 53. Résultat de Ficelef sur l'Acajou .....	52
Figure 54. De gauche à droite : Acajou, Pin Noir et Sapin.....	53
Figure 55. Coupe de Sapin avec transition été/hivers et résultat de ficelef .....	54
Figure 56. Coupe de Sapin non colorée .....	55

Figure 57. Zone à problème. En noir le tracé expert.....	55
Figure 58. Coloration par un expert sur une image d'acajou.....	56

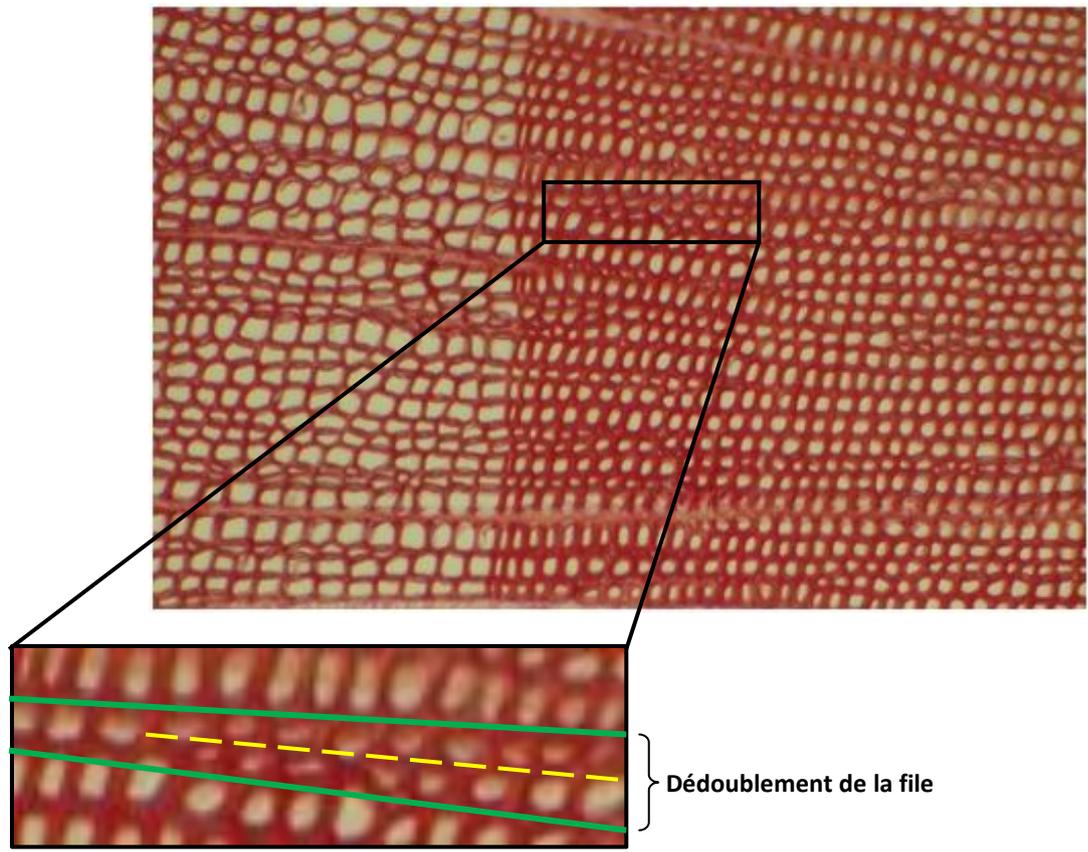
# TABLES DES ANNEXES

Annexe 1. Schéma de l'organisation de l'AMAP .....	64
Annexe 2. Dédoublment d'une file .....	64
Annexe 3. Panel d'image utilisé pour les tests .....	65
Annexe 4. Pseudo Code de l'algorithme de la Ligne de Partage des eaux.....	66
Annexe 5. Algorithme des colonies de Fourmis .....	67
Annexe 6. Utilisation du plugin Ficelef dans ImageJ .....	68
Annexe 7. Protocole de coloration.....	69
Annexe 8. Algorithme de détection des bassins versants (code Java).....	71
Annexe 9. Diagramme de classe des classes FBasin et FEdge.....	72
Annexe 10. Diagramme de classe des packages IJ et IJ.Processor .....	73

Annexe 1. Schéma de l'organisation de l'AMAP

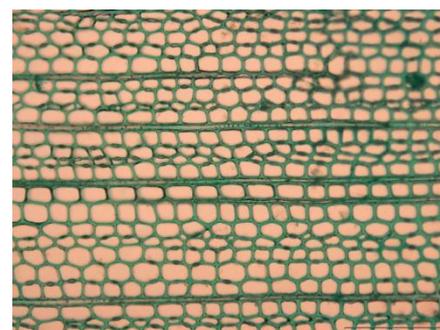
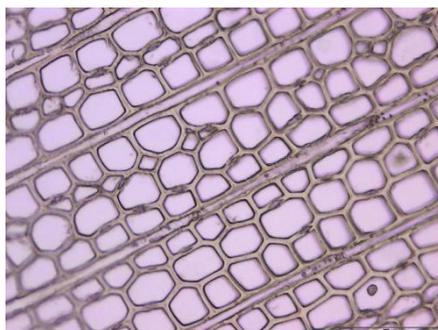
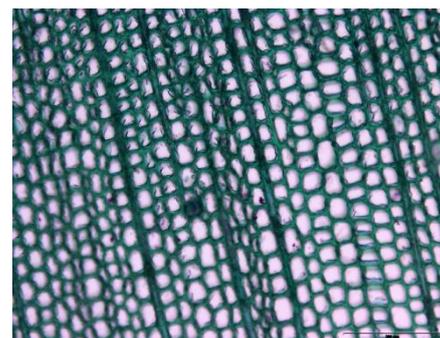
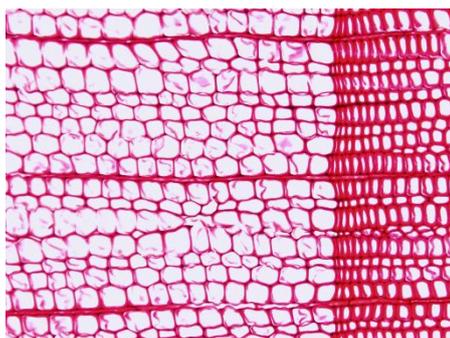
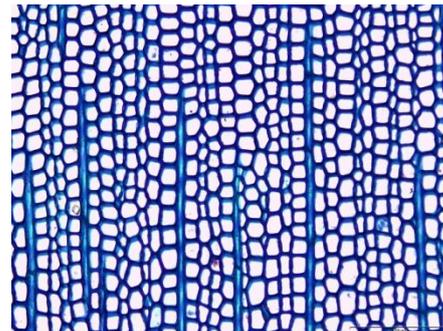
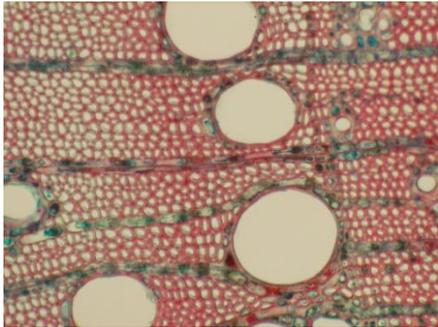
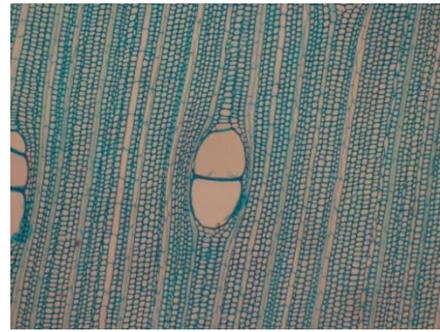
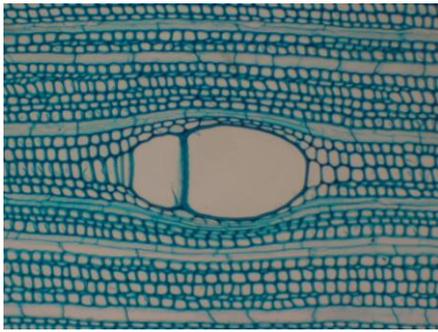


Annexe 2. Dédoublage d'une file



Coupe transversale de Pin noir

Annexe 3. Panel d'image utilisé pour les tests



De gauche à droite, **Feuillus** : Acajou (x400), Acajou (x100), Frêne (x400), Chêne Vert (x100). **Résineux** : Pin Noir (x200), Sapin (x400), Pin Monticola (x400), Sapin (x400), Sapin (x400, Non coloré), Sapin (x400).

Annexe 4. Pseudo Code de l'algorithme de la Ligne de Partage des eaux

```
Pour NIVEAU de 0 à HMAX

  /** Action 1 : Etendre les régions existantes ***/

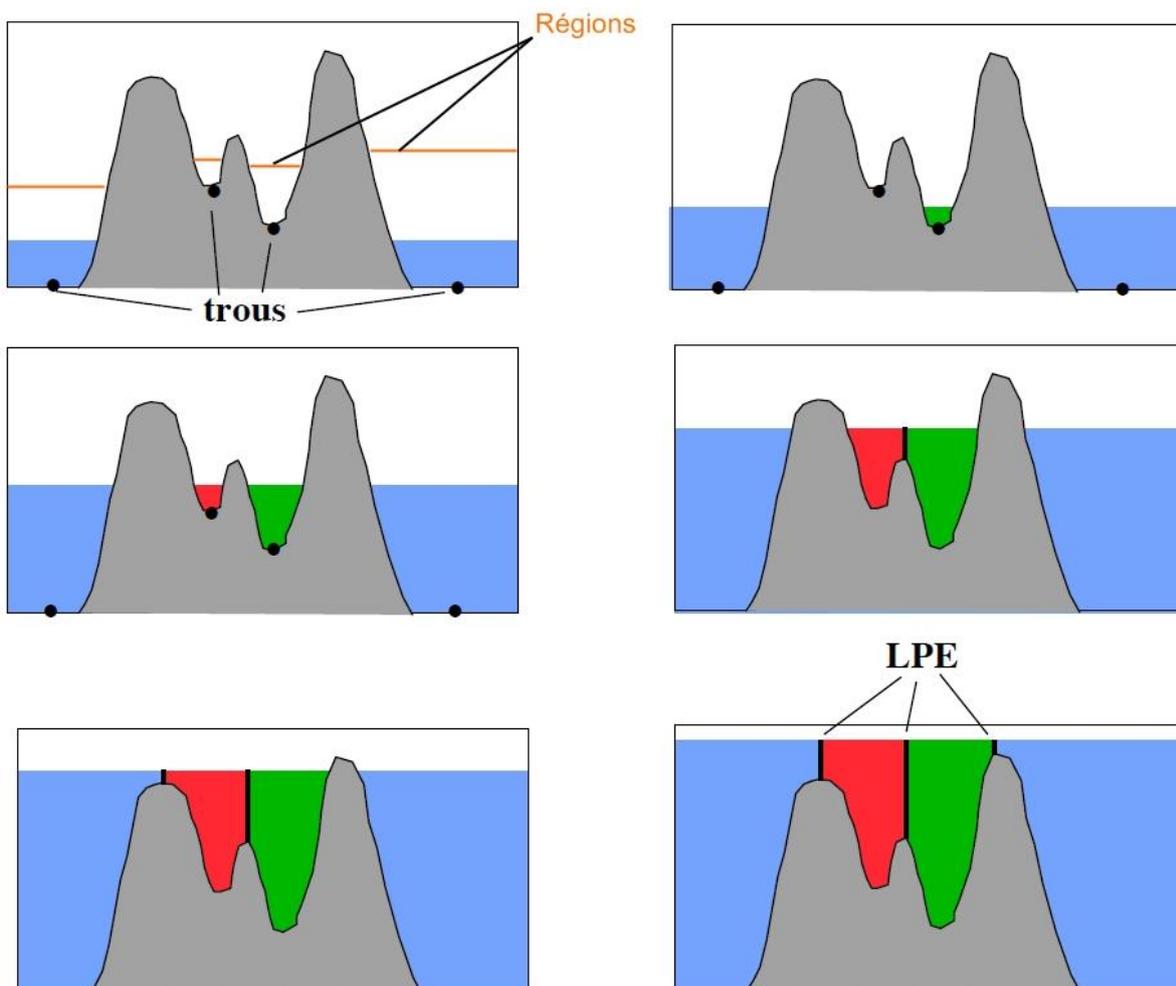
  Pour chaque Région « [R] » dans la liste « [Régions] »
    Faire croître « [R] » jusqu'à l'altitude NIVEAU
  Fin Pour

  /** Action 2 : créer les nouvelles régions ***/

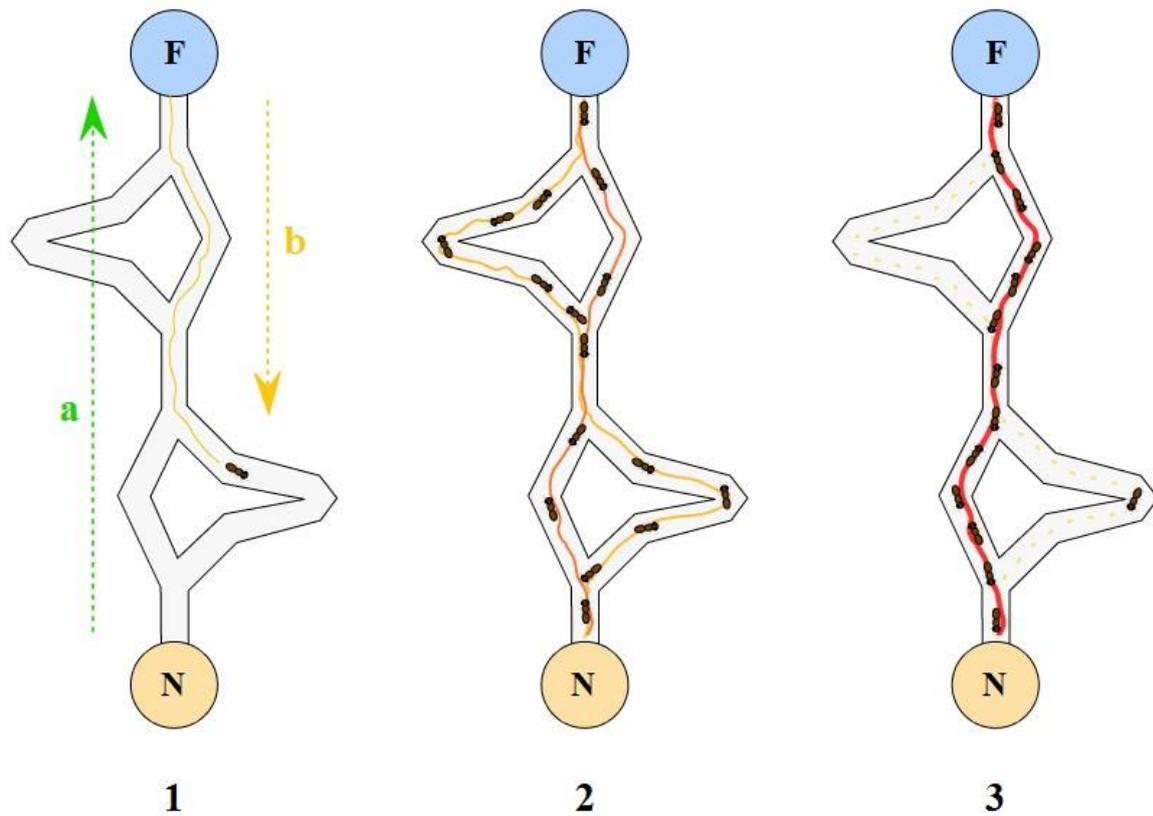
  Pour chaque pixel « P » à l'altitude NIVEAU
    Si « P » n'est pas associé à une région Alors
      Créer une nouvelle région « [R] » dans la liste « [Régions] »
      Ajouter le pixel « P » dans la région « [R] »
      Faire croître « [R] » jusqu'à l'altitude NIVEAU
    Fin Si
  Fin Pour

Fin Pour
```

*Schéma récapitulatif de l'algorithme de la ligne de partage des eaux :*



## Annexe 5. Algorithme des colonies de Fourmis



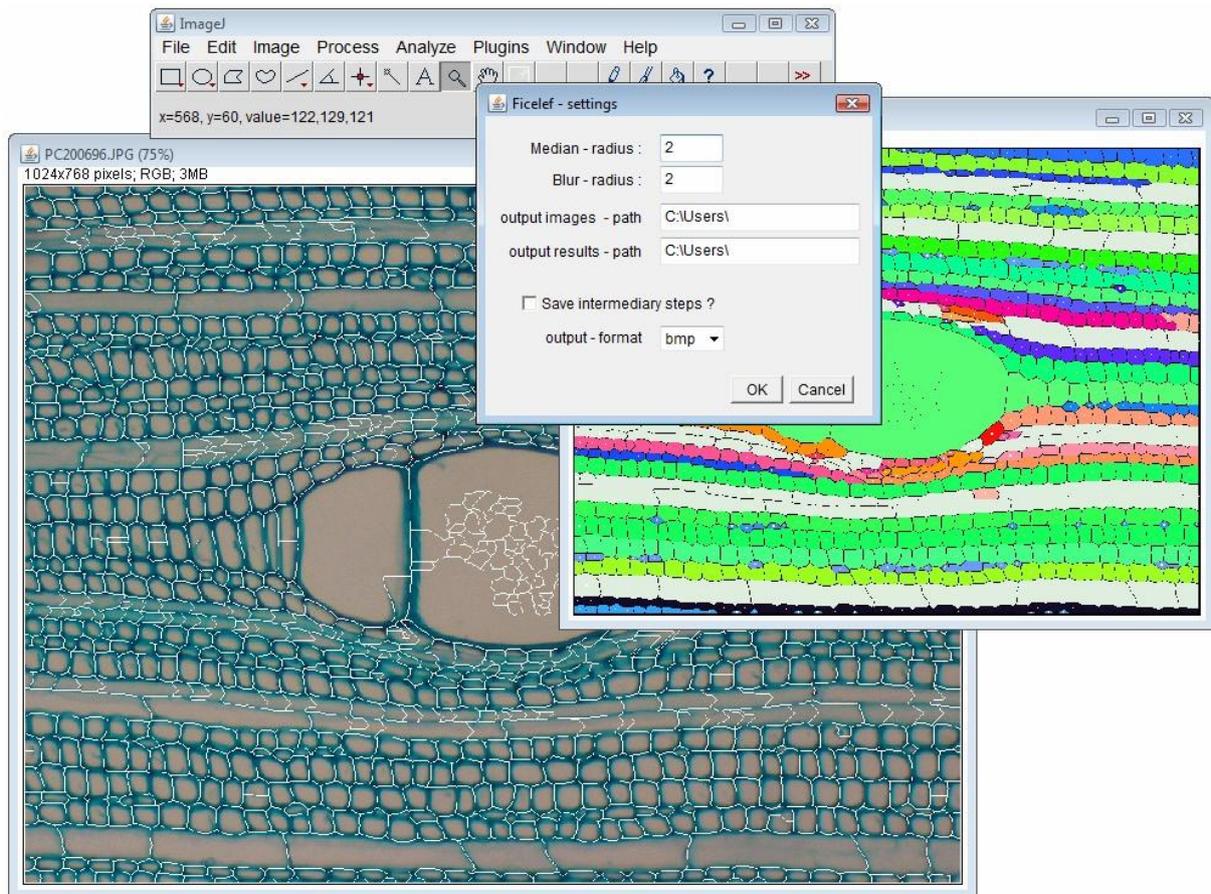
### Détail du parcours :

1. la première fourmi trouve la source de nourriture (F), via un chemin quelconque (a), puis revient au nid (N) en laissant derrière elle une piste de phéromone (b).
2. les fourmis empruntent indifféremment les quatre chemins possibles, mais le renforcement de la piste rend plus attractif le chemin le plus court.
3. les fourmis empruntent le chemin le plus court, les portions longues des autres chemins perdent leur piste de phéromones.

On observe en pratique qu'un grand nombre d'algorithmes se réclament d'une inspiration « colonies fourmis », sans toujours partager le cadre général de l'optimisation par colonies de fourmis canonique (ACO : « *Ant Colony Optimization* »). En pratique, l'utilisation d'un échange d'informations entre fourmis via l'environnement (principe dénommé « stigmergie ») suffit à rentrer dans la catégorie des algorithmes de colonies de fourmis. Ce principe a mené certains auteurs à créer le terme d'« optimisation stigmergique ».

On trouve ainsi des méthodes s'inspirant de comportements de recherche de nourriture, de tri de larves, de division du travail ou de transport coopératif.

## Annexe 6. Utilisation du plugin Ficelef dans ImageJ



## Protocole de coloration à la safranine.

### Objectif

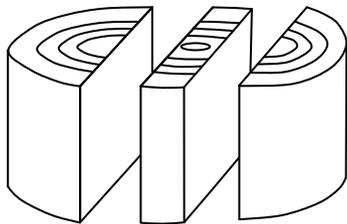
Coloration des tissus végétaux pour observation microscopique : comptage du nombre de cernes de croissance.

### Préparation de l'échantillon

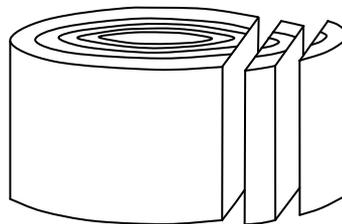
#### Préparation physique

Réaliser une coupe propre du tronçon de rameau à décrire : sécateur ou scie selon le diamètre du rameau.

Si la surface du rameau excède 1 cm<sup>2</sup>, tailler le tronçon en lame longitudinale, au ciseau à bois (plusieurs possibilités : lame radiale, lame tangentielle)



Coupe longitudinale radiale



Coupe longitudinale tangentielle

Insérer l'échantillon sur le porte-objet du microtome. S'assurer qu'il est bien fixé.

- ❖ Ajuster la verticalité de l'échantillon dans l'axe perpendiculaire au sens de coupe, et dans l'axe de coupe.
- ❖ Ajuster la hauteur du porte-objet pour être très proche de la lame. Affiner l'approche par rotation de la molette crantée ou en utilisant le mouvement incrémenté de la lame.
- ❖ Ajuster les angles d'attaque de la lame par rapport à l'échantillon :
  - si la lame est trop horizontale, elle glissera sur l'échantillon, si elle est trop verticale, elle accrochera l'échantillon et fera des sauts.
  - éviter une position provoquant l'enroulement de la coupe (cause des problèmes lors du montage sur lame).

« Nettoyer » la surface de l'échantillon par plusieurs coupes.

Ajuster l'épaisseur de coupe : utiliser la tige (située à l'avant du microtome) qui règle le nombre de crans pris par le mouvement de retour du porte-lame.



Attention ! Ne jamais toucher la lame avec un objet dur (risque d'abîmer le fil).

Réaliser des coupes les plus fines possibles (récupérer les coupes au pinceau et les stocker dans l'eau distillée.

Afin de faciliter la coupe, humidifier régulièrement l'échantillon et la lame.

Nettoyer la lame avec le nettoyant pour métaux.

Nettoyer le microtome et la pailleuse.

## Préparation chimique



Attention ! Ne pas traiter toutes les coupes réalisées, mais en conserver quelques unes (une dizaine) dans l'éthanol 70% pour traitement ultérieur, si nécessaire (gain de temps, disponibilité variable d'échantillons frais).

Déposer les coupes dans un cristalliseur contenant de l'eau de javel.

Suivre la séquence de préparation suivante :

Etape	Cristalliseur	Temps (minutes)	Commentaires
1	Eau de javel (12° chloro.)	15	« nettoyage » des cellules
Rinçage à l'eau distillée			
2	Eau distillée	10	
Rinçage à l'eau distillée			
3	Acide acétique 10%	5	Blocage du pH, fixation
Rinçage à l'eau distillée			
4	Eau distillée	10	
Rinçage à l'alcool 70°			
5	Ethanol 70%	10	Déshydratation de l'échantillon
6	Ethanol 100%	10	Déshydratation de l'échantillon
Récupérer l'alcool dans un récipient sec			
7	Safranine (1% dans Ethanol 50%)	10 minimums	Stockage sous la hôte Juste pour couvrir les coupes
Rincer avec l'alcool récupéré puis avec de l'alcool « propre », jusqu'à obtention d'une solution de lavage incolore (toujours récupérer les bains de rinçage dans le récipient à alcool)			

### Quelques précautions à prendre :

- lors des différents rinçages, procéder lentement afin de ne pas évacuer les coupes.
- la safranine pure peut être réutilisée d'une série de coupes à une autre, en prenant garde de ne pas transvaser des coupes dans le cristalliseur de la série suivante.
- jeter les solutions aqueuses dans les bonbonnes spécifiques.

Pour traiter simultanément plusieurs séries de coupes (6 au maximum), déposer les coupes dans les supports munis d'un filet. Les différents bains seront réalisés dans des boîtes rectangulaires transparentes.

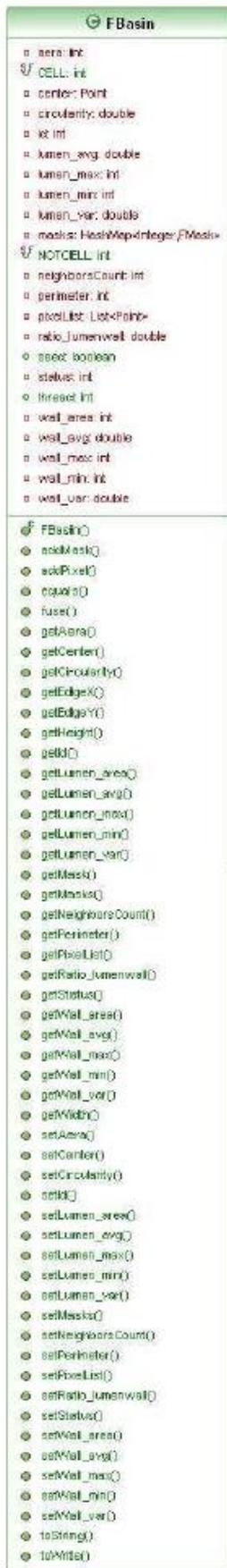
Nettoyer le matériel.

Annexe 8. Algorithme de détection des bassins versants (code Java)

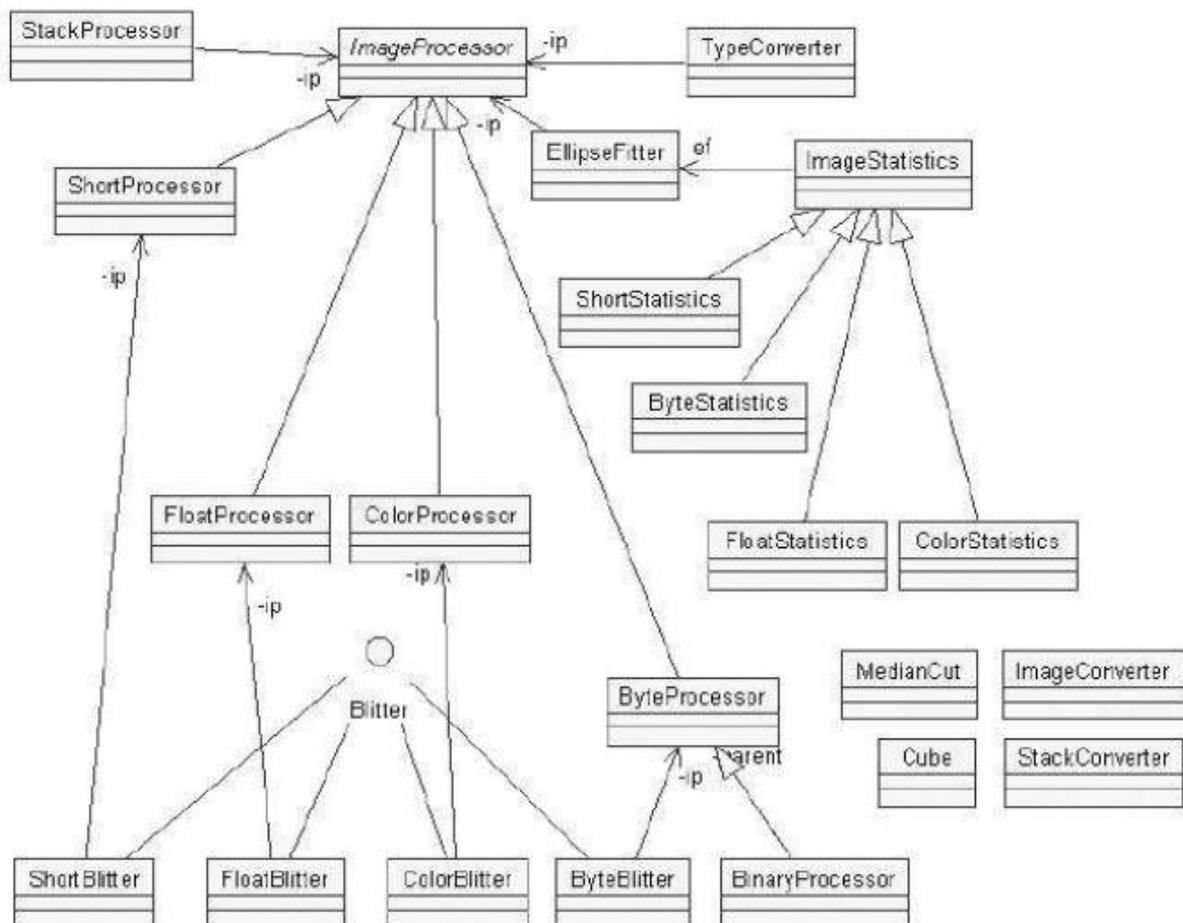
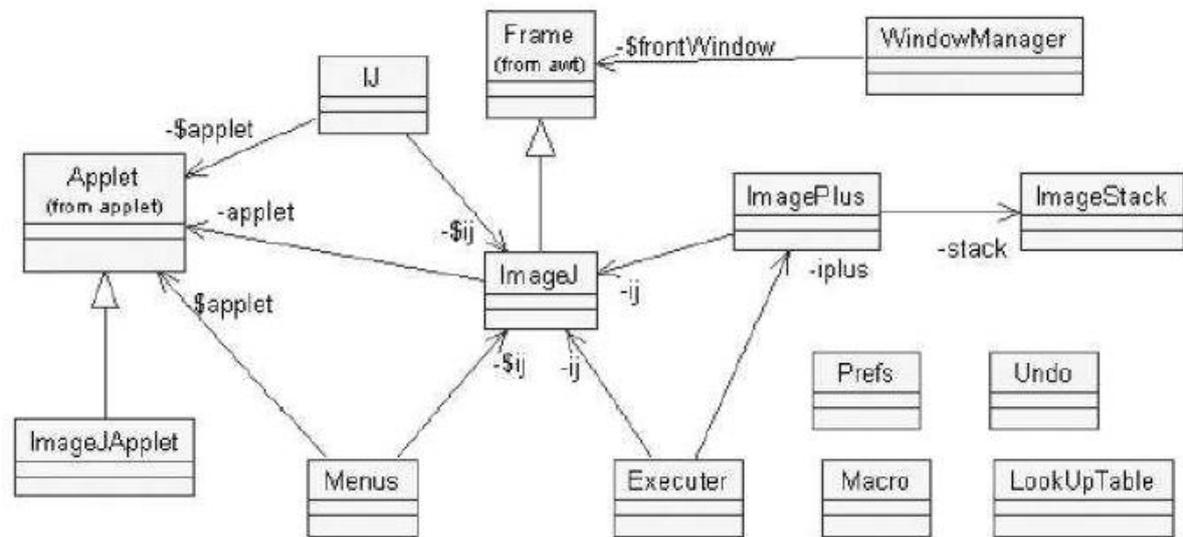
```
/**
 * METHOD WHO FULL THE BASIN STRUCTURE FROM WATERSHED IMAGEPROCESSOR
 */
private void extractBasins() {
    // initialize boolean read[W][H] array to false (W=image width, H=image
    // height) to know read pixel.
    initReadArray();
    int number = 0;
    for (int y = 0; y < H; y++) {
        for (int x = 0; x < W; x++) {
            FBasin currentBasin;
            List<Point> listpix;
            // if pixel is not read and black
            if (!read[x][y] && watershedIProcessor.getPixel(x, y) == 0) {
                listpix = new ArrayList<Point>();
                // build current basin
                trackBasin(x, y, listpix);
                currentBasin = new FBasin(number, listpix);
                this.listOfBasin.add(currentBasin);
                number++;
            }
        }
    }
    indexBasin = new int[listOfBasin.size()];
    for (FBasin basin : listOfBasin)
        indexBasin[basin.getId()] = basin.getId();

    selectRightBasin();
    extractPixelBasin(listOfBasin);
    System.out.println(listOfBasin.size() + " basins detected (xxx)");
}
```

Annexe 9. Diagramme de classe des classes FBasin et FEdge



Annexe 10. Diagramme de classe des packages `ij` et `ij.Processor`



# RESUME

Le projet **Ficelef** concerne la reconnaissance automatique de **Files Cellulaires** chez les **Feuillus**. Il a été le produit original d'application et développement informatique en traitement d'images, statistiques, mathématiques sur un sujet aujourd'hui indispensable à la compréhension de l'homme concernant le développement et l'architecture des plantes. Ces besoins ont été émis par l'unité mixte de recherche « *botAnique et bioinforMatique de l'Architecture des Plantes* » spécialiste dans ce domaine. Les résultats obtenus auront permis une approche nouvelle du problème déjà réfléchi dans le passé. Une chaîne de traitement aura été mise en place, fonctionnelle sous le logiciel libre de traitement d'images ImageJ. Le plugin fourni pourra cependant suivre autant d'améliorations techniques que d'innovations pouvant élargir sa portée au niveau des espèces et/ou structures traitées.

Mots-clés : *traitement d'image, file cellulaire, plugin, ImageJ, classification, détection automatique.*

# SUMMARY

The Ficelef project concerns the automatic recognition of cell lines among angiosperm. It was the original product of application and computer development in image processing, statistics, mathematics on a subject now necessary to the human understanding of plants' development and structure. These needs were brought to light by the mixed unit of research AMAP: an expert in the field. Acquired results had allowed a new approach on this matter already reflected in the past. A processing chain has been set up, functional under free image processing software: ImagesJ. The provided plugin will, however, be able to follow as much technical improvements as innovations that can enlarge his range on others tree species and/or treated structures.

Key word : *image processing, cell lines, plugin, ImageJ, classification, Autodiscovery.*