

INP-ENSEEIH

PROJET DE FIN D'ETUDES

Détection de cellules embryonnaires en imagerie biomédicale 3D

Auteur:

Gaël MICHELIN

Superviseur:

Dr. Grégoire MALANDAIN

Mémoire de Projet de Fin d'Etudes

soutenu le 27/09/2013 pour l'obtention du diplôme de

Ingénieur de l'INP-ENSEEIH

Département Informatique et Mathématiques Appliquées

Septembre 2013

Résumé

Détection de cellules embryonnaires en imagerie biomédicale 3D

La compréhension de la morphogenèse (ensemble des lois déterminant la forme, la structure des organismes) des embryons est un important enjeu dans le domaine de la biologie du développement. Récemment, de nombreux travaux de recherche ont été appliqués à la question de la segmentation des cellules embryonnaires, et ce problème est encore aujourd'hui d'actualité. En raison de la quantité d'information très importante qu'il y a à traiter, il est indispensable de pouvoir proposer des méthodes entièrement automatisées, fiables et algorithmiquement efficaces pour effectuer cette tâche. Ce mémoire présente le travail qui a été effectué sur le sujet de la détection et la segmentation des cellules en imagerie 3D obtenue par microscopie confocale. Nous aborderons dans ce document les aspects théoriques, techniques, expérimentaux et méthodologiques qui ont entouré ce stage.

Remerciements

Mes premiers remerciements s'adressent à Grégoire Malandain, mon encadrant, qui m'a beaucoup fait profiter de son expertise en matière de traitement de l'image, de changement de pneus, de café ou encore de babyfoot.

Merci aussi à Léo Guignard, doctorant à l'Université Montpellier 2 et à qui je dois de nombreux conseils dans mon appréhension de diverses problèmes au cours de ce stage.

À l'origine de cette aventure, je n'oublie pas de remercier Jean-Denis Durou et Xavier Descombes qui m'ont donné cette opportunité de rejoindre l'équipe MORPHEME en m'accordant toute leur confiance il y a un an déjà.

Je remercie aussi les nombreuses personnes de l'équipe VORTEX de l'IRIT-Toulouse qui m'ont donné goût au traitement de l'image et m'ont encouragé à poursuivre dans cette voie. Je pense en particulier à Sylvie Chambon, ma tutrice de stage et encadrante l'an passé, mais aussi à Vincent Charvillat et à Géraldine Morin.

Je tiens aussi à remercier tout particulièrement Emmanuel Soubies et Alexis Zubiolo, pour leur aide et leur complicité durant ces six mois de stage.

De manière générale, je remercie aussi l'ensemble de l'équipe Morphème pour tout son soutien pendant ces 6 mois de stage. Je pense notamment à Eric Debreuve, à qui j'ai donné beaucoup de travail à cause de mes problèmes d'ordinateur et d'imprimante.

Table des matières

Résumé	i
Remerciements	iii
Table des matières	v
Liste des figures	viii
Liste des tableaux	x
1 Introduction	1
1.1 L'équipe	1
1.2 Contexte	2
1.2.1 Sujet du stage	2
1.2.2 Enjeux de la reconstruction de cellules	3
1.3 Organisation du mémoire	3
2 Procédés appliqués à la reconstruction de cellules embryonnaires	5
2.1 Etat de l'art	5
2.1.1 Détection des membranes	5
2.1.2 Détection et segmentation des cellules	6
2.2 Notre approche	6
2.2.1 Chaîne de traitements	6
2.2.2 Détection des membranes	7
2.2.3 Raccordement des discontinuités	7
2.2.4 Détection et segmentation des cellules	8
2.2.5 Analyse des résultats	8
2.2.5.1 Plan de tests	8
2.2.5.2 Bonnes détections, sous-détections, sur-détections	8
2.2.5.3 Résultats	10
2.3 Perspectives	12
3 Fonction de planéité	13
3.1 Introduction	13
3.2 Dérivées directionnelles : notions appliquées au traitement d'image	13

3.2.1	Dérivées d'ordre 1 : vecteur gradient	14
3.2.2	Dérivées d'ordre 2 : matrice hessienne	14
3.2.3	Dérivation sur des images	16
3.2.3.1	Dérivation par convolution	16
3.3	Fonctions de rehaussement des membranes	16
3.3.1	Éléments de théorie	16
3.3.2	Implémentation	18
3.4	Extraction des plans centraux	19
3.5	Binarisation des extrema	21
4	Tensor Voting	25
4.1	Introduction	25
4.2	Structure tenseur	26
4.2.1	Formalisme des tenseurs d'ordre 2	26
4.2.2	Décomposition en valeurs et vecteurs propres	26
4.3	Vote des tenseurs	27
4.3.1	Tenseur Stick	28
4.3.2	Tenseur Ball	28
4.3.3	Tenseur Plate	29
4.3.4	Cas général	29
4.4	Fonction de décroissance	29
4.4.1	Choix de la fonction de décroissance	29
4.4.2	Limitation	30
4.5	Processus de vote	30
4.6	Champs de vote	31
4.6.1	Discrétisation d'un demi-cercle	32
4.6.2	Discrétisation d'une demi-sphère	32
4.7	Application à la détection des parois cellulaires et résultats	33
5	Segmentation des cellules	37
5.1	Introduction	37
5.2	Ligne de partage des eaux	37
5.3	Choix des graines de régions	38
5.3.1	Minima régionaux	38
5.3.2	Maxima de distance	38
5.3.3	Érosion, taille des composantes connexes et labellisation	41
5.4	Résultats	42
6	Conclusion et perspectives	45
6.1	Apports du travail réalisé	45
6.2	Enrichissement personnel	45
	Bibliographie	47
	A Résultats complets des détections	A. 1

B	Tensor Voting	B. 1
B.1	Solution analytique : <i>Closed-Form solution to Tensor Voting</i>	B. 1
B.1.1	Votes Stick	B. 1
B.1.2	Votes Ball	B. 2
B.1.2.1	Cas 2D	B. 3
B.1.2.2	Cas 3D	B. 5
B.1.3	Théorème de la solution analytique pour le Tensor Voting	B. 5
C	Article	C. 1

Table des figures

2.1	Chaîne de traitement pour la segmentation des cellules.	7
2.2	Plan de tests	9
2.3	Illustration des différents scénarios de détection	10
3.1	Gradient dans une image	14
3.2	Profils d'intensité et leurs dérivées premières en un point de membrane	15
3.3	Profils d'intensité et leurs dérivées secondes en un point de membrane	17
3.4	Interpolation trilineaire	19
3.5	Résultats des fonctions de planéité	20
3.6	Résultats des extractions de plans centraux	21
3.7	Résultats des fonctions de planéité	23
4.1	Décomposition d'un tenseur	27
4.2	Vote d'un tenseur stick	28
4.3	Champs de votes stick	30
4.4	Champ de votes émis par un tenseur	31
4.5	Discretisation du demi-cercle	32
4.6	Discretisation non-régulière d'une sphère	33
4.7	Discretisation de la sphère	33
4.8	Résultats du Tensor Voting	35
5.1	Ligne de partage des eaux	38
5.2	Calcul des graines pour la LPE	39
5.3	h -minima	40
5.4	Minima régionaux sur l'image de Tensor Voting	40
5.5	Carte des distances et h -maxima de distance	41
5.6	Érosion des h -maxima	41
5.7	Watershed appliqué à l'image S	43
5.8	Diagrammes évaluant les résultats obtenus	44
B.1	Vote reçu.	B. 1

Liste des tableaux

2.1	Résultats des détections	11
A.1	Tableau complet des statistiques de détections	A.3

Chapitre 1

Introduction

1.1 L'équipe

Le déroulement de ce stage s'est passé au sein de l'équipe de recherche MORPHEME. Il s'agit d'un projet commun créé en 2011 entre l'Université de Nice - Sophia Antipolis, le CNRS et l'INRIA. Ce projet associe le Laboratoire Informatique, Signaux Systèmes de Sophia Antipolis (I3S) ainsi que l'Institut de Biologie de Valrose (iBV). Le responsable de l'équipe est le Dr Xavier Descombes, et les thèmes abordés y sont la Biologie Numérique et la Bioinformatique.

Les objectifs scientifiques de MORPHEME sont la caractérisation et la modélisation du développement et des propriétés morphologiques de structures biologiques allant de l'échelle cellulaire à l'échelle supra-cellulaire. A la frontière entre informatique, mathématiques appliquées et biologie, notre ambition est de comprendre les changements morphologiques qui apparaissent durant le développement en combinant imagerie in vivo, analyse d'image et modélisation numérique. La morphologie et la topologie des structures mésoscopiques ont en effet une influence majeure sur les aspects fonctionnels des organes. Notre ambition est de caractériser différentes populations ou différentes conditions de développement à partir de la forme des structures cellulaires ou supra-cellulaires, ce qui inclut les réseaux micro-vasculaires, ou les réseaux dendritiques et axonaux. A partir d'images 2D, 2D+t, 3D ou 3D+t (obtenues par microscopie confocale, bi-photon, vidéo-microscopie ou micro-tomographie), nous nous attachons à extraire des paramètres quantitatifs pour caractériser la morphométrie dans différents échantillons et son évolution au cours du temps. Nous analysons ensuite statistiquement les formes et les structures complexes pour identifier des marqueurs significatifs et des outils de classification. Pour finir, nous proposons des modèles explicatifs de l'évolution temporelle des échantillons observés. Nous espérons ainsi mieux comprendre le développement des tissus dans un contexte normal mais aussi caractériser, à un niveau supra-cellulaire, différentes pathologies telles que Alzheimer, le diabète ou le syndrome X-fragile.

L'équipe MORPHEME est structurée suivant les axes de recherche suivants :

- Formation des images : cela inclut i) la définition des populations étudiées (conditions expérimentales) et la préparation des échantillons pour la question biologique étudiée, ii) l’optimisation des protocoles d’acquisition et la définition des caractéristiques quantitatives mesurées, iii) la reconstruction et la restauration des données initiales pour améliorer leur qualité et leur interprétabilité.
- Extraction de marqueurs : cela consiste à détecter et extraire les structures biologiques d’intérêt à partir des images. Pour ce faire, des modèles biologiques définis a priori doivent permettre d’améliorer la détection et la caractérisation des structures. Les deux défis principaux concernent la variabilité des structures biologiques et la taille importante des données.
- Classification/Interprétation : à partir d’une base de données contenant différentes populations, nous pouvons estimer les paramètres d’un modèle pour chaque donnée ou image à partir de laquelle la structure biologique étudiée a été extraite. Nous pouvons alors définir des schémas de classification pour caractériser les différentes populations à partir de ces paramètres ou d’une métrique spécifique aux structures étudiées. Cette caractérisation fournira des informations biologiques définissant les différentes populations.
- Modélisation : deux aspects seront considérés. Le premier consiste à modéliser des phénomènes biologiques comme la croissance axonale ou la topologie de réseaux (neuronaux, vasculaires) dans différents contextes. Un de nos atouts réside dans la possible calibration et/ou validation des modèles par les informations extraites des images. Le second aspect concerne l’utilisation de modèles, biologiquement pertinents, pour l’extraction de l’information des images.

1.2 Contexte

1.2.1 Sujet du stage

L’objectif général de ce stage est de proposer, implémenter et évaluer une méthode originale de détection et segmentation de structures linéiques (réseaux vasculaires) ou planes (membranes cellulaires) dans des images biologiques tridimensionnelles. Certaines spécificités de l’imagerie biologique rendent difficile la tâche d’extraction des informations. En particulier, ces images présentent le plus souvent un bruit important, les structures sont parfois incomplètes et la résolution optique des images est inégale selon les différentes directions de l’espace. Au cours du stage, le sujet s’est recentré sur la détection et la segmentation des cellules d’un embryon dans des images d’embryons d’un animal marin nommé *Phallusia mammillata*, à différents stades de son développement. Ces images sont obtenues par microscopie confocale, et l’intensité des structures membranaires ne présente pas une propriété isotrope, ce qui justifie la nécessité d’outils spécifiques pour établir la segmentation des cellules.

1.2.2 Enjeux de la reconstruction de cellules

Notre compréhension des processus de développement des embryons est essentiellement fondée sur des observations effectuées *in vivo*. La morphogenèse, *i.e.* la formation d'un organisme par des mouvements cellulaires, les interactions entre cellules, le comportement collectif des cellules, les changements de forme des cellules, la division et la mort des cellules, constitue un processus dynamique de réarrangements intracellulaires rapides et de changements structurels plus lents au niveau de l'embryon. Un défi consiste à décrire l'évolution de l'activité des gènes lors du stade embryonnaire du développement [Munro et al., 2006] [Traas and Hamant, 2009]. Les récents progrès en microscopie à prises de vues à intervalles réguliers permettent aujourd'hui d'enregistrer pendant plusieurs heures des organismes vivants à une fréquence suffisamment élevée pour permettre de surveiller les déformations et les divisions cellulaires [Keller, 2013]. La capacité de capturer à la fois le comportement propre aux cellules et leur interaction au niveau global sur une période de temps longue est donc devenu un enjeu crucial pour la compréhension des mécanismes biologiques sous-jacents. L'obtention d'un schéma complet de la morphogenèse nécessite donc la mise en place d'un système d'imagerie et de traitement d'image à haute résolution spatio-temporelle très performant, ce qui pourrait apporter d'importantes contributions dans le domaine de la biologie du développement embryonnaire [Truong and Supatto, 2011]. La très grande taille des données à traiter nécessite cependant que les outils d'analyse d'images soient également peu coûteuses en temps de calcul.

1.3 Organisation du mémoire

Le premier chapitre constitue une présentation du cadre du stage ainsi qu'une brève introduction au sujet traité. En fin de chapitre, une partie récapitule l'organisation globale du mémoire.

Dans le chapitre 2, un état de l'art succinct est proposé afin de présenter les principaux procédés utilisés en traitement de l'image pour la détection des parois cellulaires et des cellules. Nous présentons ensuite une vue d'ensemble des procédés que nous appliquons pour réaliser la segmentation des cellules contenues dans une image.

Le chapitre 3 présente une méthode originale pour effectuer la détection de structures planaires. Cette méthode basée sur des informations dérivatives du premier et du second ordre, adaptée d'un travail portant sur la détection de structures linéiques, est ensuite appliquée à la détection de membranes dans le contexte de l'imagerie 3D. Une comparaison est faite avec une autre méthode également basée hessienne.

Le chapitre 4 décrit une méthode de groupement perceptuel de l'information dans le cas de l'imagerie 3D : on propose dans ce chapitre d'appliquer cette méthode à la réorganisation des membranes cellulaires après l'étape de détection décrite par le chapitre précédent.

Puis le chapitre 5 donne une solution pour exploiter le résultat précédent afin de segmenter les régions cellulaires. L'enchaînement complet que nous avons mis au point y est détaillé et une méthodologie d'évaluation quantitative est donnée.

Enfin, le dernier chapitre présente les perspectives du travail réalisé, et la dernière partie met en valeur l'enrichissement personnel lié au stage, d'un point de vue technique, mais aussi d'un point de vue des rapports humains.

Chapitre 2

Procédés appliqués à la reconstruction de cellules embryonnaires

2.1 Etat de l'art

Avant de présenter la chaîne de traitements que nous proposons pour réaliser la segmentation des cellules, voici un succin résumé des méthodes existantes dans la littérature.

De nombreuses recherches en imagerie biomédicale 3D abordent les problèmes liés à l'analyse d'images d'embryons ou d'organismes végétaux. On peut classifier les divers algorithmes liés à la détection des cellules en deux groupes : les approches basées contours, où on exploite les propriétés intrinsèques aux membranes cellulaires, et celles basées région qui utilisent des informations sur la forme des régions constituant les cellules.

2.1.1 Détection des membranes

Le plus souvent, dans l'objectif de réaliser une segmentation des cellules d'une image, les approches basées contours sont utilisées en tant que pré-traitements permettant d'améliorer l'image d'entrée en éliminant du bruit et en augmentant le contraste entre les structures planes et les autres formes de structures. On peut citer différents filtres permettant de réaliser cette tâche. Ces filtres peuvent être basés sur les propriétés photogrammétriques ou/et structurelles des membranes. On trouve des méthodes de type détection de symétrie de phase [Kovesi, 1997], des techniques s'appuyant sur la morphologie mathématique [Luengo-Oroz et al., 2007], les filtres d'ondelettes [de Monvel et al., 2001] ou des filtres linéaires (lissage gaussien) [Marcuzzo et al., 2008].

Un autre procédé est d'utiliser un détecteur de surfaces en 3D par des filtres de diffusion [Mosaliganti et al., 2010].

Ce procédé exploite le fait qu'il est possible de déduire de la matrice Hessienne (matrice des dérivées secondes de l'intensité de l'image) associée à un point de l'image à la fois une probabilité de présence de surface en ce point et, le cas échéant, l'orientation de cette surface. En effet,

si on considère un point situé sur une surface 3D de l'image, et si on considère les valeurs et vecteurs propres de la matrice Hessienne associée au point, deux de ces valeurs propres seront proches de 0 et le plan formé par les vecteurs propres associés constituera le plan tangent à la surface, tandis que la dernière valeur propre possèdera une grande valeur absolue témoignant d'un changement fort de dérivée première directionnelle et donc d'un extremum d'intensité en ce point dans la direction orthogonale à la surface portée par le vecteur propre correspondant.

On obtient à l'aide de ces diverses méthodes une image où le rapport signal / bruit a été grandement amélioré, ce qui doit permettre de faciliter l'étape de détection. La méthode de prétraitement à utiliser est dépendante de la technique de détection qu'on souhaite réaliser.

2.1.2 Détection et segmentation des cellules

Plusieurs méthodes sont envisageables pour procéder à la segmentation des cellules. Un procédé classique et algorithmiquement peu coûteux, la ligne de partage des eaux (algorithme *watershed* en anglais) [Vincent and Soille, 1991], a été adapté à ce problème [Fernandez et al., 2010, Mosaliganti et al., 2012]. Cependant, le bruit important des images traitées peut causer des irrégularités indésirables entre les cellules, ainsi que des erreurs de sur-segmentation et parfois de sous-segmentation, ce qui nécessite de procéder à des étapes de pré- et post-segmentation afin de prévenir et réparer ces erreurs. L'utilisation de méthodes basées modèles déformables (contours actifs, l'idée est de minimiser une fonction d'énergie afin que le modèle se superpose à l'image du mieux possible) est une alternative qui permet l'obtention de cellules bien formées [Meijering et al., 2009], mais le coût d'une telle méthode peut s'avérer très grand lorsqu'une image contient un nombre important de cellules, et cette méthode entraîne des erreurs de sous-segmentation.

2.2 Notre approche

2.2.1 Chaîne de traitements

Nous avons adopté lors de ce stage une approche semblable à celle présentée dans [Mosaliganti et al., 2012]. Dans cet article, le traitement s'effectue en trois temps principaux : i) un prétraitement permettant de rehausser les structures membranaires via un filtre de diffusion [Mosaliganti et al., 2010], ii) une étape de propagation des membranes par Tensor Voting [Medioni et al., 2000] pour obtenir une nouvelle image de membranes de meilleure qualité que la précédente, iii) la segmentation des cellules par un algorithme de ligne de partage des eaux. Ce procédé a démontré sa capacité à obtenir une segmentation de bien meilleure qualité qualitativement et quantitativement en comparaison à une ligne de partage des eaux effectuée sur l'image originale. Cependant, le coût algorithmique du Tensor Voting est non-négligeable et cette méthode a le défaut de requérir un temps de calcul important.

Nous avons choisi d’adopter une stratégie similaire (figure 2.1) en proposant d’une part un détecteur de surfaces original basé sur l’exploitation des dérivées premières et secondes de l’image, et d’autre part en suggérant différentes stratégies de raccordement par Tensor Voting qui présentent l’avantage d’être plus rapides que la solution originale.

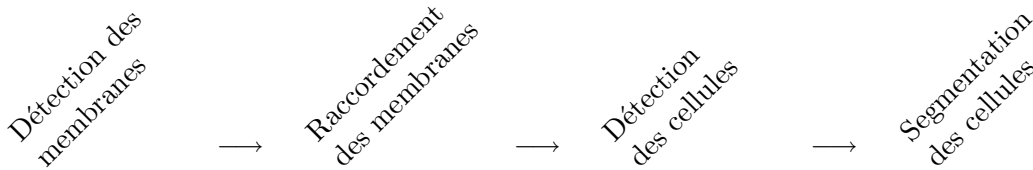


FIG. 2.1: Chaîne de traitement pour la segmentation des cellules.

2.2.2 Détection des membranes

Notre approche pour la détection des membranes s’appuie sur le travail réalisé par [Krissian et al., 2000], qui propose un détecteur de structures filaires multi-échelle. Dans le cadre de ce stage, nous avons donc adapté le modèle proposé dans ce travail au cas de la détection de structures planaires, mais nous nous sommes limités à un cadre mono-échelle en émettant la supposition que l’épaisseur des membranes est approximativement la même en tout point de l’image. Le lecteur pourra se référer au chapitre 3 pour davantage de précisions quant à cette méthode basée sur l’exploitation des dérivées premières et secondes de l’image.

Nous avons également implémenté la fonction de réponse utilisée dans [Mosalganti et al., 2012], ce qui nous a permis d’évaluer et de comparer les deux approches.

2.2.3 Raccordement des discontinuités

Cette étape est réalisée en appliquant une méthode d’organisation perceptuelle (psychologie de la forme) 3D appelée Tensor Voting. Cette méthode est d’autant plus coûteuse que la quantité d’éléments échangeant de l’information est grande dans l’image à raccorder. Dans [Mosalganti et al., 2012], l’information exploitée est l’ensemble des points de l’image où la fonction de réponse a retourné une valeur non-nulle, ce qui correspond à une image quasi-dense.

Nous avons cherché à alléger cette étape en réalisant une présélection beaucoup plus forte des points qui échangent de l’information lors du Tensor Voting. Le but étant de refermer les structures comportant des trous, nous avons procédé à l’extraction des plans centraux qui résultent de l’étape de détection des membranes. Cette phase d’extraction est détaillée dans la section 3.4. De cette façon, seuls les extrema locaux directionnels de la fonction de réponse sont utilisés pour le procédé du vote, ce qui réduit considérablement le temps de calcul de cette étape.

Nous avons enfin envisagé de procéder à la binarisation de l'image des extrema afin d'en mesurer l'impact sur le Tensor Voting et sur la suite de la chaîne de traitement. On se référera à la section 3.5 pour plus de précisions sur la binarisation des extrema.

2.2.4 Détection et segmentation des cellules

Nous avons fait le choix de privilégier la méthode de la transformation de la ligne de partage des eaux pour la segmentation des cellules. Cette approche, qui fait toujours l'objet d'études pour cette application, pose deux nouvelles questions. D'abord, il faut proposer à l'algorithme un ensemble de *graines* qui correspondent à une initialisation des cellules de l'image. La détermination des graines est d'une grande importance car c'est celle-ci qui détermine le nombre et la position initiale de ces graines. Le calcul de ces graines correspond à l'étape de détection des cellules. Enfin, il faut proposer à l'algorithme de ligne de partage des eaux une image *topographique* qui permette une bonne croissance des graines.

Notre travail ayant essentiellement porté sur la détection et le raccordement des membranes lors de ce stage, nous avons choisi de procéder de manière systématique pour la détection et la segmentation des cellules (chapitre 5), afin d'obtenir des détections et des segmentations comparables pour les différents procédés utilisés précédemment. Ainsi, nous sommes en mesure de conclure sur la pertinence de nos méthodes de détection des membranes et de raccordement des discontinuités.

2.2.5 Analyse des résultats

2.2.5.1 Plan de tests

Notre travail a porté majoritairement sur l'ensemble des aspects précédant la détection des cellules dans la chaîne de traitement (figure 2.1). Pour évaluer la qualité de ces différents traitements, nous avons mis en place une stratégie de tests (figure 2.2). Dans cette stratégie, nous générons un ensemble de segmentations en faisant varier le processus de calcul entre chaque test. Pour permettre une appréciation de la valeur ajoutée des étapes de pré-détection, nous avons recueilli les détections et segmentations obtenues par traitement direct de l'image d'origine, et nous avons reproduit l'expérience proposée dans [Mosaliganti et al., 2012].

2.2.5.2 Bonnes détections, sous-détections, sur-détections

Notre objectif est d'évaluer quantitativement la performance de chacune des chaînes de traitement proposées. Connaissant une Vérité Terrain (VT) de l'image d'origine, obtenue par une expertise, nous connaissons la cellule d'appartenance théorique de chacun des points de l'image d'origine. Une première approche pour quantifier nos résultats serait de procéder à un calcul des coefficients de Dice, c'est-à-dire où on mesure un ratio de recouvrement entre les cellules de la VT et celles de la segmentation. Cependant, nous cherchons davantage à estimer

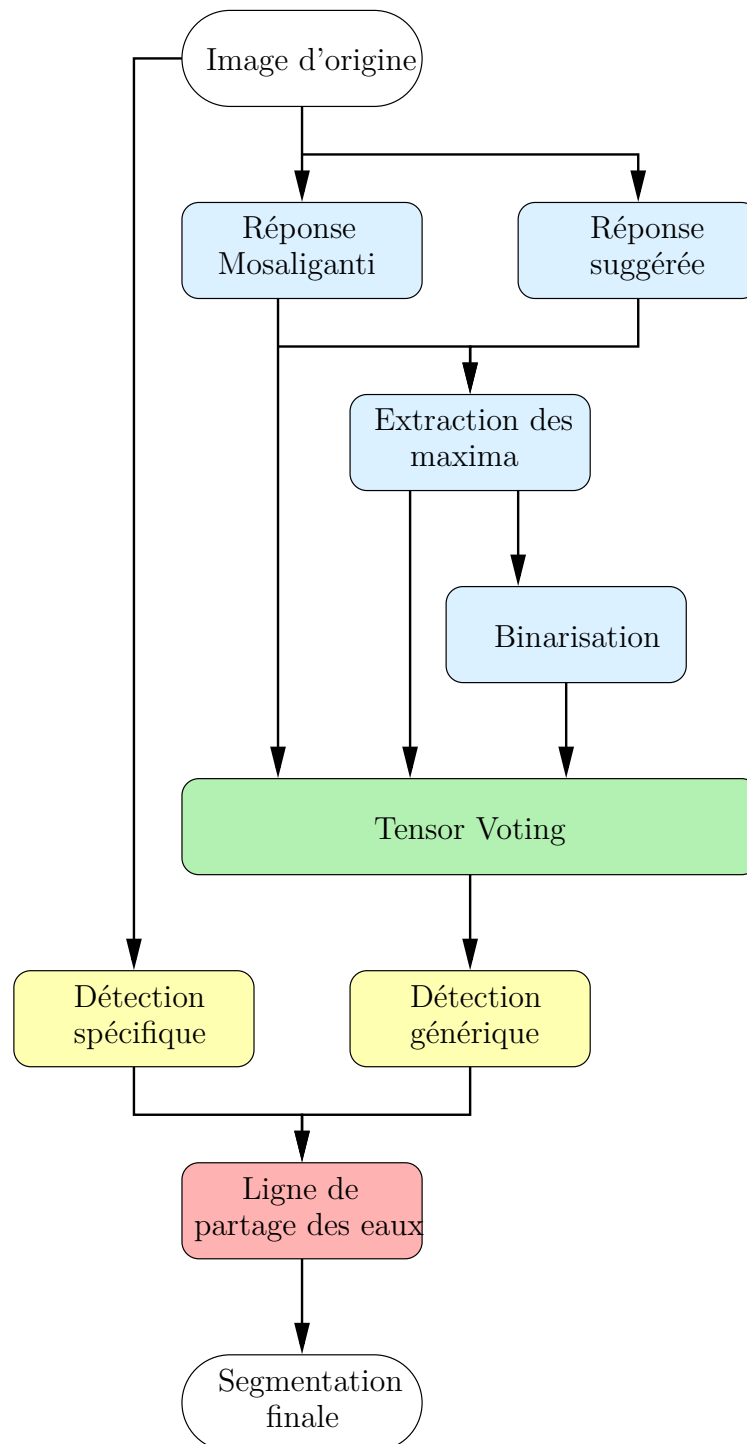


FIG. 2.2: Plan de tests mis en place pour l'évaluation des méthodes de détection et raccordement des membranes. Les calculs des dérivées sont effectués à différentes échelles. En bleu : détection des membranes. En vert : raccordement des membranes. En jaune : détection des cellules. En rouge : segmentation des cellules.

la qualité des détections plutôt que des segmentations finales. Nous avons donc défini comme suit les trois notions de *bonne détection* TD (true detection), de *sur-détection* OD (over-detection) et de *sous-détection* UD (under-detection) :

- une TD correspond à une cellule de la VT dont l'étape de détection associe une unique graine, et dont cette graine est entièrement incluse dans la cellule ;
- une OD d'ordre $n \geq 1$ correspond au cas où une cellule ou région de la VT est associé à $n + 1$ graines ;
- une UD d'ordre $n \geq 1$ correspond au cas où une graine s'intersecte avec $n + 1$ cellules de la VT. Une cellule ne contenant aucune graine engendre également une UD d'ordre 1.

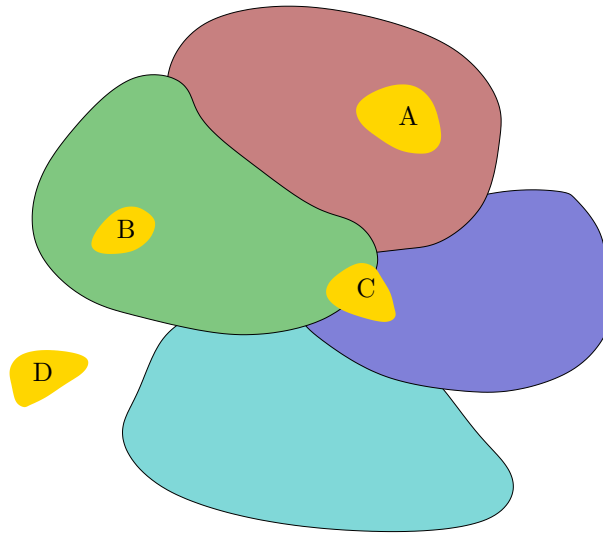


FIG. 2.3: Illustration des différents scénarios de détection. Les graines apparaissent en jaune, les cellules de la Vérité Terrain sont les autres régions colorées. La cellule rouge et (A) sont une bonne détection. La cellule verte est un cas de sur-détection puisqu'il est associé à (B) et (C). La graine (C) est un cas de sous-détection car elle intersecte deux cellules (verte et bleue). La cellule cyan est un cas de sous-segmentation car elle ne contient aucune graine. La graine (D) est un cas de sur-détection car elle n'est incluse dans aucune cellule de la Vérité Terrain. Dans cet exemple, on comptabilise ainsi 1 TD, 2 OD et 2 UD.

2.2.5.3 Résultats

Nous avons mesuré les bonnes, sous- et sur-détections précédemment définies à l'aide de tableaux de confusion. Un échantillon des résultats correspondants aux meilleures configurations trouvées est présenté dans le tableau 2.1. Nous évaluons dans ce tableau à la fois la qualité des détections obtenues, mais aussi le temps mis par l'algorithme de Tensor Voting afin de mesurer les bénéfices en temps de calcul offerts par le procédé que nous proposons. Ces tests ont été réalisés sur deux images réelles d'un même embryon capturées à quelques heures d'intervalle. Le tableau complet des tests réalisés est disponible en annexe A.

Il est important de préciser qu'il existe des algorithmes de post-segmentation permettant d'apporter des corrections à celle-ci [Fernandez et al., 2010]. Cependant, on constatera que s'il est possible de post-traiter de façon automatisée le cas des sur-segmentations, le problème des sous-segmentations ne peut être résolu qu'à l'aide d'une intervention humaine. Nous privilégions donc grandement des erreurs de type sur-segmentation aux erreurs de sous-segmentation.

Image (#cellules)	Réponse	Tensor Voting	σ	h_{\min}	TD	OD	UD	Temps TV (s)
Im1 (32)	Michelin	BINAIRE	5	30	30	3	0	1943
	Mosaliganti	BINAIRE	5	30	30	3	0	2104
	Mosaliganti	REPONSE	3	10	28	4	0	65210
	Michelin	EXTREMA	4	10	28	4	0	8253
	REFERENCE					31	9	0
Im2 (162)	Michelin	BINAIRE	4	15	149	15	3	2422
	Michelin	EXTREMA	5	10	144	11	8	7009
	Mosaliganti	REPONSE	4	15	139	18	8	64682
	Michelin	BINAIRE	3	15	136	29	2	2562
	REFERENCE					128	40	5

TAB. 2.1: Résultats des détections pour deux images d'un embryon à différents moments de son développement. #cellules : nombre de cellules de la VT. Réponse {Michelin/Mosaliganti} : type de fonction de réponse choisi. Tensor Voting {REPONSE/EXTREMA/BINAIRE} : procédé utilisé pour le Tensor Voting (figure 2.2). σ : échelle utilisée (en pixels) pour le calcul des dérivées d'intensité. h_{\min} : unique paramètre variable de la phase de détection (voir section 5.3.1). TD/OD/UD : nombre de bonnes/sur-/sous-détections. Temps TV : temps mis pour effectuer l'étape de Tensor Voting (en secondes).

À la lecture du tableau, certains enseignements sont apportés.

- Le **coût de calcul du Tensor Voting** varie fortement entre les trois méthodes présentées. Comme on pouvait le présupposer, la réalisation du Tensor Voting sur les extrema à la place de l'image d'origine offre un gain de temps considérable, le processus étant alors de l'ordre de 10 fois plus rapide. De plus, en effectuant une binarisation de l'image des extrema, on est encore en mesure de réduire d'un facteur voisin de 3 le temps nécessaire au Tensor Voting, soit un un facteur supérieur à 25 entre le Tensor Voting de [Mosaliganti et al., 2012] et celui que nous présentons dans le cas binarisé.
- Les résultats de détection donnent les résultats les plus satisfaisants pour une **échelle de calcul** des dérivées voisine de 4, indépendamment de la fonction de réponse choisie, ce qui justifie en partie l'approximation de départ que nous nous étions donné, à savoir que l'épaisseur des membranes varie peu dans l'espace ainsi que dans le temps.
- Si la première image ne semble pas privilégier l'utilisation de l'une ou l'autre des **fonctions de planéité**, la seconde image laisse transparaître une tendance légèrement favorable à l'utilisation de la fonction que nous avons proposé. Dans les deux cas, la valeur ajoutée de l'application d'une fonction de réponse associée au Tensor Voting paraît indéniable en comparaison à la détection de référence.
- L'étape de **Tensor Voting** donne dans ces exemples de meilleurs résultats avec une initialisation binaire, et de moins bons résultats avec l'initialisation de [Mosaliganti et al., 2012] qui prend en paramètre l'ensemble des points de la fonction de réponse. L'intérêt de procéder à cette étape d'extraction des extrema est donc double puisqu'il permet d'améliorer à la fois les performances de temps de calcul et de détection de l'image.

2.3 Perspectives

Nous avons réalisé une méthode de détection de structures planaires originale composée d'une fonction de planéité nouvelle combinée à une étape de Tensor Voting réalisée sur une sélection fortement restrictive des points issus de la fonction de réponse. Nous avons appliqué ce procédé à la détection de membranes cellulaires sur des images d'embryons de *Phallusia mammillata* capturées par microscopie confocale, avec l'obtention de résultats encourageants. Une première amélioration à apporter sera l'adaptation de notre fonction de planéité au cas multi-échelle, ce qui nous permettra de nous affranchir du choix de l'échelle de calcul. Nous avons également de nombreuses améliorations à proposer d'un point de vue algorithmique dans la réalisation du Tensor Voting (parallélisation des calculs, structure des données), afin de permettre une exploitation plus efficace de cet outil. Cette étape a par ailleurs montré de bons comportements lors de son application sur des images d'extrema binarisés de la fonction de réponse. Il serait donc intéressant de s'interroger sur le choix des paramètres optimisant cette binarisation.

Nous avons également proposé une suite de traitements permettant de procéder à la détection puis à la segmentation des cellules en exploitant le travail précédent. L'étape de détection des cellules est décisive puisque les graines obtenues au terme de cette étape conditionnent entièrement l'étape de segmentation des cellules. Au cours de ce stage, nous avons fixé un certain nombre de paramètres jugés peu influents pour la détection finale. Il sera important pour les autres paramètres d'évaluer leur criticité et de fournir des outils de sélection automatique de l'ensemble de ces paramètres.

Le post-traitement des segmentations n'a pas été au centre de notre étude. Il serait malgré tout intéressant de prendre davantage en considération la manière dont ce post-traitement est réalisé afin de proposer en amont une segmentation qui soit le plus adaptée possible à cette phase importante du processus.

Il ne faut enfin pas perdre de vue l'objectif final de ce travail qui est de parvenir à réaliser un tracking des cellules à travers le temps, et de proposer aux biologistes le meilleur outil possible afin d'étudier la morphogenèse de l'embryon. Un certain nombre de travaux ont déjà été réalisés dans ce sens, mais cela n'a pas été abordé au cours de ce stage et une prochaine étape serait donc de prendre en considération l'évolution dans le temps de l'embryon pour réaliser l'opération de segmentation et de suivi des cellules et de leur descendance.

Chapitre 3

Fonction de planéité

3.1 Introduction

Nous avons souligné dans le chapitre 2 l'importance de réaliser en guise de prétraitement de l'image un filtrage conservant les structures apparentées à des surfaces et éliminant les autres structures. Dans ce chapitre, nous proposons une revue de l'ensemble du travail qui a été effectué dans ce domaine à l'occasion de ce stage.

Il était possible de choisir entre les langages C et C++ pour réaliser l'implémentation des algorithmes et fonctions. J'ai fait le choix du langage C pour réaliser les fonctions présentées dans cette partie.

3.2 Dérivées directionnelles : notions appliquées au traitement d'image

Les fonctions de planéité que nous présentons dans ce chapitre sont toutes basées sur les informations différentielles de l'intensité de l'image. Avant de détailler plus précisément ces méthodes, il est important d'avoir quelques connaissances préliminaires sur le sens des dérivées dans le cadre de l'imagerie.

Nous définissons dans cette section les éléments suivants :

- $\Omega \subset \mathbb{R}^3$: ensemble ouvert de points de l'espace.
- $I : (x, y, z) \in \Omega \mapsto I(x, y, z) \in \mathbb{R}^+$: fonction d'intensité \mathcal{C}^∞ .
- $D_{\mathbf{v}}I$: fonction dérivée directionnelle de I suivant le vecteur \mathbf{v} .
- $\frac{\partial I}{\partial u}$: fonction dérivée partielle de I selon la variable u .
- $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial z} \right)^T$: fonction vecteur gradient de l'image.
- $H(I) = \left(\frac{\partial^2 I}{\partial u_i \partial u_j} \right)_{1 \leq i, j \leq 3}$: fonction matrice Hessienne $((u_1, u_2, u_3) \stackrel{\text{n}}{=} (x, y, z))$.
- $\{\nu_1, \nu_2, \nu_3\} \in \mathbb{R}^3$: valeurs propres associées à $H(I)$ telles que $|\nu_1| \geq |\nu_2| \geq |\nu_3|$.
- $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\} \in (\mathbb{R}^3)^3$: vecteurs propres $H(I)$ associés aux valeurs propres $\{\nu_1, \nu_2, \nu_3\}$.

3.2.1 Dérivées d'ordre 1 : vecteur gradient

Considérons la fonction $I : \Omega \rightarrow \mathbb{R}^+$, et soit $P \in \Omega$ un point de l'espace. Le vecteur gradient $\nabla I(P)$ pointe la direction de plus forte variation de la fonction scalaire I , et son amplitude donne la force de cette variation (figure 3.1).

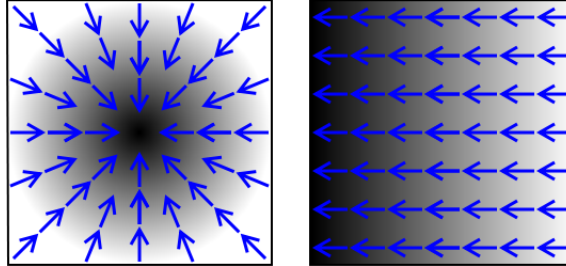


FIG. 3.1: Exemple de gradient : les lignes bleues représentent le gradient de couleur du plus clair vers le plus foncé. Illustration issue de Wikipedia.

Intéressons-nous au cas d'une fonction I contenant une structure de type surfacique de forte intensité, et au comportement du gradient de l'image au voisinage d'un point P situé sur cette surface.

Soit le repère $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ orthonormal du plan tel que :

- $\{\mathbf{v}_2, \mathbf{v}_3\}$ est une base du plan tangent à la surface au point P ,
- \mathbf{v}_1 est un vecteur normal à la surface en P .

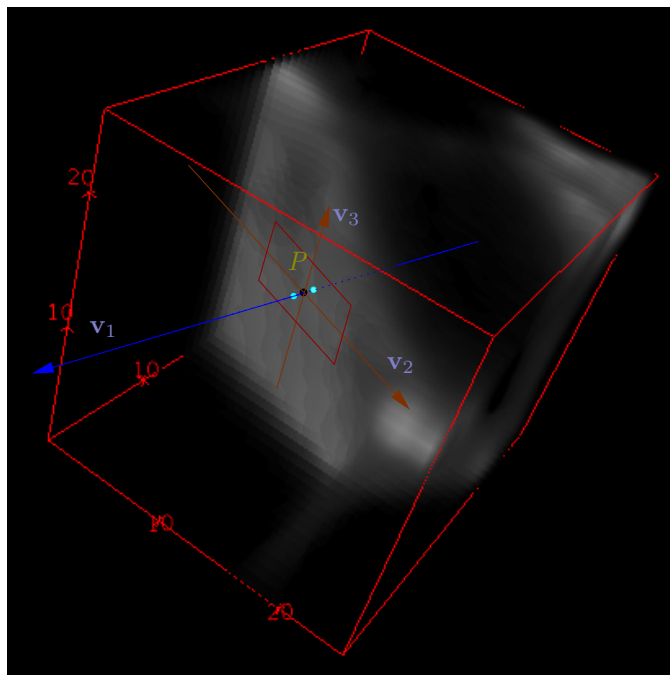
Le profil de l'intensité I apparaît comme étant peu variable selon le plan tangent à P dans le voisinage de ce point, tandis que dans la direction portée par \mathbf{v}_1 on doit observer un pic d'intensité au point P . Analytiquement, cela se traduit par des dérivées directionnelles de faible intensité dans le plan $\{\mathbf{v}_2, \mathbf{v}_3\}$, et par un profil typique de dérivée directionnelle dans la direction \mathbf{v}_1 . La figure 3.2 illustre ces différents cas.

3.2.2 Dérivées d'ordre 2 : matrice hessienne

Soit $P \in \Omega$ et $H \stackrel{\text{H}}{=} H(I)(P)$ la matrice des dérivées secondes de l'image, appelée matrice hessienne de I au point P :

$$H = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2}(P) & \frac{\partial^2 I}{\partial x \partial y}(P) & \frac{\partial^2 I}{\partial x \partial z}(P) \\ \frac{\partial^2 I}{\partial x \partial y}(P) & \frac{\partial^2 I}{\partial y^2}(P) & \frac{\partial^2 I}{\partial y \partial z}(P) \\ \frac{\partial^2 I}{\partial x \partial z}(P) & \frac{\partial^2 I}{\partial y \partial z}(P) & \frac{\partial^2 I}{\partial z^2}(P) \end{pmatrix}$$

La décomposition de H en valeurs propres $\{\nu_1, \nu_2, \nu_3\}$ et vecteurs propres $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$ donne une interprétation intéressante de la matrice hessienne, puisque les valeurs propres correspondent aux dérivations du second ordre dans les directions portées par leurs vecteurs propres respectifs,



(A) Membrane cellulaire dans l'image 3D.

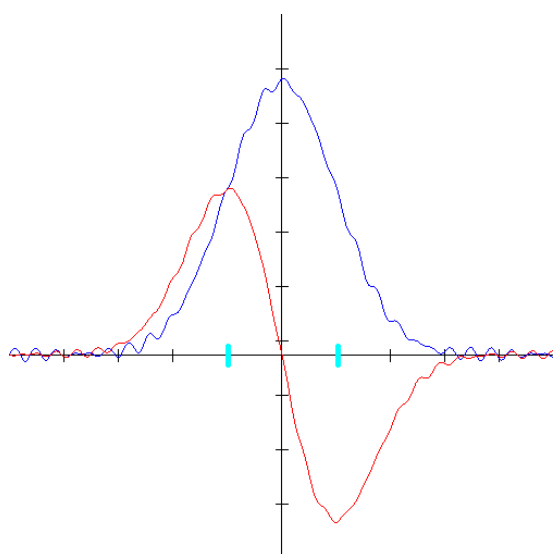
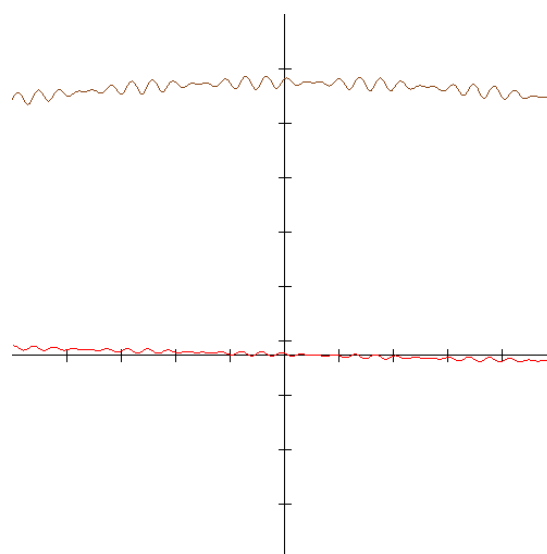
(B) Intensité $\delta \mapsto I(P + \delta \mathbf{v}_1)$ (courbe bleu) et dérivée directionnelle $\delta \mapsto D_{\mathbf{v}_1}(P + \delta \mathbf{v}_1)$ (courbe rouge).(C) Intensité $\delta \mapsto I(P + \delta \mathbf{v}_i)$ (courbe brune) et dérivée directionnelle $\delta \mapsto D_{\mathbf{v}_i}(P + \delta \mathbf{v}_i)$ avec $i = 2$ ou 3 (courbe rouge).

FIG. 3.2: Illustration des différents profils d'intensité et de dérivées directionnelles selon le plan tangent à une structure plane et selon la direction normale à cette structure. Dans le cas de la dérivée dans la direction normale à la membrane, on observe deux extrema de dérivées correspondant aux deux points qui apparaissent en cyan sur l'image 3.2a. Ces points correspondent aux positions en lesquelles la variation de l'intensité est maximale et leur éloignement de P est fonction de l'épaisseur de la membrane.

ou plus formellement :

$$D_{\mathbf{n}_i}^2 I(P) = \nu_i \quad (1 \leq i \leq 3) \quad (3.1)$$

Or les vecteurs propres d'une matrice hessienne déterminent les axes de courbure principaux de la fonction I , ce qui signifie que :

$$\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\} = \{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\} \quad (3.2)$$

L'analyse de la matrice hessienne en P nous informe donc sur l'orientation de la membrane et sur les dérivées secondes directionnelles selon la direction normale \mathbf{n}_1 et les directions tangentes \mathbf{n}_2 et \mathbf{n}_3 relatives à la membrane au point P . La figure 3.3 illustre que, si P est un point de membrane, on doit avoir :

- $|\nu_1| \gg |\nu_2| \simeq |\nu_3| \simeq 0$,
- $\nu_1 > 0$ si la membrane est de luminosité intense sur fond sombre, $\nu_1 < 0$ sinon,
- \mathbf{n}_1 donne la direction normale à la membrane au point P .

3.2.3 Dérivation sur des images

Dans le cadre du traitement d'image, l'hypothèse d'avoir une fonction I définie sur un ouvert Ω et \mathcal{C}^∞ n'est pas vérifiée : une image 3D numérique est formée d'un ensemble de voxels, et admet donc un domaine de définition discret. Il est cependant possible de calculer avec une certaine approximation les dérivées d'une fonction discrète.

3.2.3.1 Dérivation par convolution

On obtient ainsi selon chaque direction la dérivée de l'image originale lissée avec une échelle de lissage

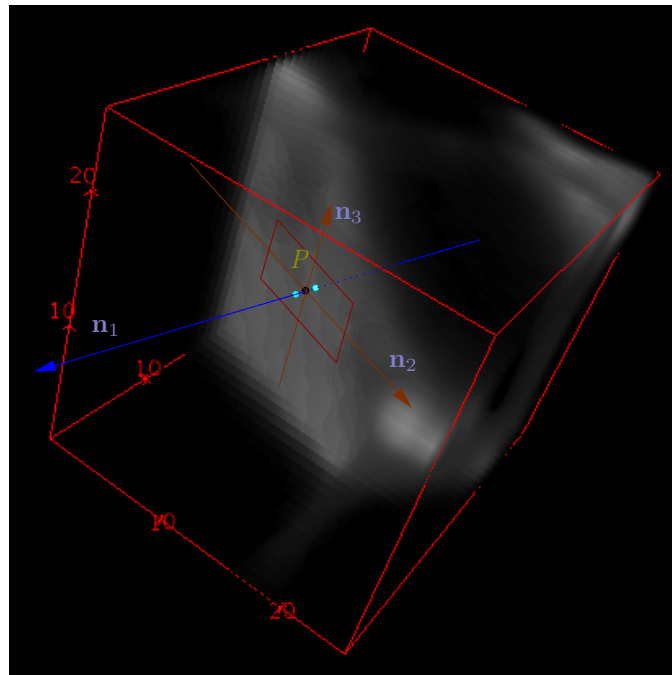
La solution que nous avons adopté pour le calcul des images gradient et hessienne est de procéder à une convolution de l'image par des filtres de dérivées gaussiennes. En effet, un tel procédé aboutit au calcul des dérivées partielles de l'image originale lissée par un paramètre d'échelle. Ce lissage présente l'intérêt d'éliminer sur le signal d'origine les petites irrégularités qui pourraient nuire à l'interprétation des dérivées.

L'implémentation de ce procédé classique en traitement de l'image était déjà réalisée.

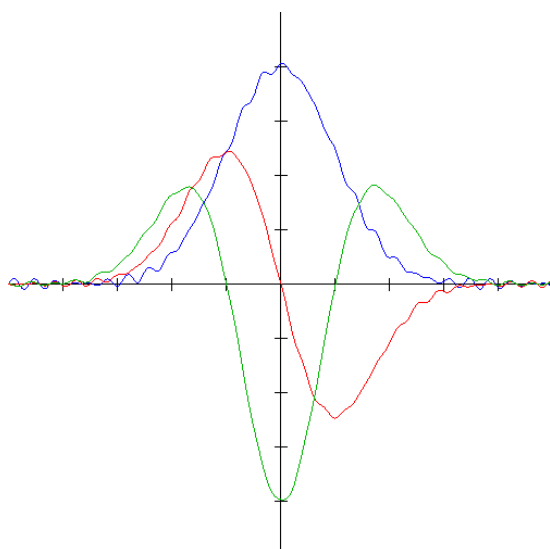
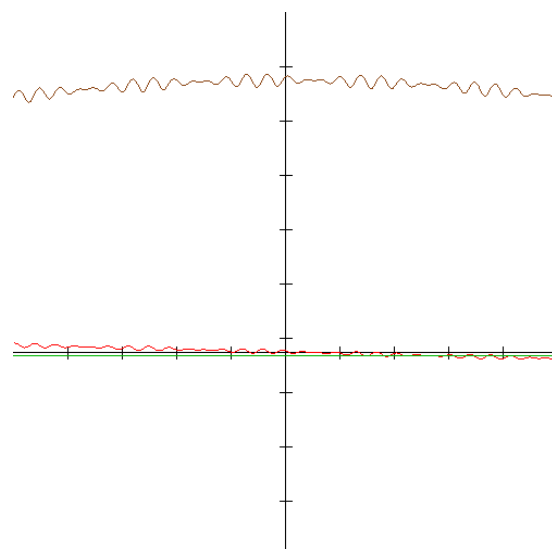
3.3 Fonctions de rehaussement des membranes

3.3.1 Éléments de théorie

En imagerie 3D, on peut caractériser les structures linéiques ou planaires à l'aide de la matrice hessienne. Soient les valeurs propres ν_i de la matrice hessienne en P qui vérifient $|\nu_1| \geq |\nu_2| \geq |\nu_3|$, on a :



(A) Membrane cellulaire dans l'image 3D.

(B) Intensité selon la direction \mathbf{n}_1 (courbe bleu) et ses dérivées directionnelles du 1er ordre (courbe rouge) et du second ordre (courbe verte).(C) Intensité selon une direction du plan tangent à la membrane en P (courbe brune) et dérivées directionnelles du 1er ordre (courbe rouge) et du second ordre (courbe verte).FIG. 3.3: Profils d'intensité et leurs dérivées directionnelles correspondantes dans les directions $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$ correspondant aux vecteurs propres de la matrice hessienne H au point P .

- Si $\nu_1 \simeq \nu_2 \simeq \nu_3 \simeq 0$, le point P est situé dans le fond avec d'éventuelles variations dues au bruit.
- Si $|\nu_1| \gg |\nu_2| \simeq |\nu_3| \simeq 0$, P est situé sur une structure de type planaire blanche sur fond noir ($\nu_1 < 0$) ou noire sur fond blanc ($\nu_1 > 0$).
- Si $|\nu_1| \simeq |\nu_2| \gg |\nu_3| \simeq 0$, avec ν_1 et ν_2 de même signe, P est situé sur une structure linéique blanche sur fond noir ($\nu_{\{1,2\}} < 0$) ou noire sur fond blanc ($\nu_{\{1,2\}} > 0$).
- Si $|\nu_1| \simeq |\nu_2| \simeq |\nu_3| \gg 0$ avec ν_1 , ν_2 et ν_3 de même signe, alors P se situe sur un pic d'intensité élevée ($\nu_i < 0$) ou faible ($\nu_i > 0$).

Dans notre cas, nous cherchons à détecter des structures planaires, qui apparaissent en blanc sur fond noir. Si un point P est situé sur une telle structure, la valeur propre ν_1 doit être négative et le vecteur propre associé \mathbf{n}_1 pointe la direction de la normale à la structure.

Nous définissons la fonction de planéité suivante sur l'image I :

$$R_\sigma(P) = \begin{cases} 0 & \text{si } \nu_1 \geq 0 \\ \frac{1}{2} (\nabla I(P - \sigma \mathbf{n}_1) \cdot \mathbf{n}_1 - \nabla I(P + \sigma \mathbf{n}_1) \cdot \mathbf{n}_1) & \text{sinon} \end{cases} \quad (3.3)$$

De cette façon, le paramètre σ doit être réglé pour que si P est situé au centre d'une structure planaire, les points $(P - \sigma \mathbf{n}_1)$ et $(P + \sigma \mathbf{n}_1)$ doivent correspondre aux deux pics de gradient qui apparaissent en cyan dans la figure 3.2. Dans ce cas, la fonction R_σ constitue un détecteur de plans dont la réponse est maximale lorsque le point P est au centre du plan, c'est-à-dire lorsque P correspond au maximum d'intensité dans la direction normale au plan. Il s'agit d'un détecteur dit basé-modèle, qui compare les informations connues en P et en son voisinage à un modèle dont on connaît théoriquement le comportement différentiel.

On voit cependant que cette fonction de réponse est sensible au paramètre d'échelle σ . Dans le cas appliqué à la détection de membranes cellulaires, l'hypothèse que ces membranes ont une épaisseur constante permet de considérer une réponse calculée pour une seule valeur de σ (détection mono-échelle). Cependant, cette hypothèse n'est pas juste puisqu'on observe qu'en raison du mode d'acquisition des images, les membranes ne sont pas détectées de la même manière selon leur orientation et apparaissent donc plus ou moins intenses et plus ou moins épaisses. Il sera nécessaire dans un prochain travail de proposer une version multi-échelle de cette fonction de planéité.

3.3.2 Implémentation

Après la phase de documentation et de recherche qui nous a amenés à proposer la fonction de planéité R_σ , il a été nécessaire d'implémenter cette fonction. Le langage à utiliser était le C, et de nombreuses fonctionnalités de base étaient déjà faites (lecture/écriture d'images, filtrage par convolution, décomposition d'une matrice en valeurs et vecteurs propres). Le codage de la fonction s'est fait de façon très basique, avec pour seule difficulté technique l'interpolation

trilinéaire (figure 3.4) des gradients de l'image pour pouvoir considérer le gradient en n'importe quelle coordonnée non-entière de l'image.

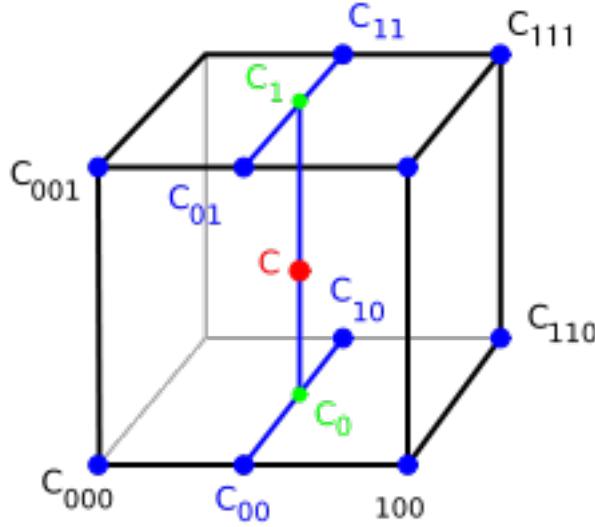


FIG. 3.4: Interpolation trilinéaire : la valeur associée au point c est déduite des valeurs c_0 et c_1 par interpolation linéaire. De même c_i est calculé par interpolation des valeurs en c_{ji} , elles-mêmes provenant des points c_{kji} (voxels de l'image) par interpolation linéaire. Illustration provenant de Wikipedia.

Afin de pouvoir comparer nos résultats avec ceux obtenus par la fonction de réponse de [Mosaliganti et al., 2010] nous avons également implémenté cette dernière qui ne présentait aucune difficulté supplémentaire à réaliser.

La figure 3.5 illustre les résultats obtenus par les deux procédés pour deux images sources différentes et à différentes échelles. Il n'est pas aisé d'estimer qualitativement laquelle des méthodes Krissian-like (équation 3.3) ou Frangi-like (fonction de réponse de [Mosaliganti et al., 2010]) donne de meilleurs résultats par rapport à l'autre. Nous avons donc comparé ces méthodes sur la base de la segmentation finale obtenue (tableau 2.1) qui accorde une légère tendance en faveur de la méthode Krissian-like.

3.4 Extraction des plans centraux

L'étape d'extraction des plans centraux est simple à mettre en œuvre. Nous définissons un maximum directionnel local comme étant un point de l'image dont la réponse est localement maximale dans la direction du vecteur \mathbf{n}_1 défini en section 3.2. On a :

$$E(P) = \begin{cases} 0 & \text{si } R(P) < R(P - \mathbf{n}_1) \text{ ou } R(P) < R(P + \mathbf{n}_1) \\ & \text{ou } (R(P) \leq R(P - \mathbf{n}_1) \text{ et } R(P) \leq R(P + \mathbf{n}_1)) \\ R(P) & \text{sinon} \end{cases} \quad (3.4)$$

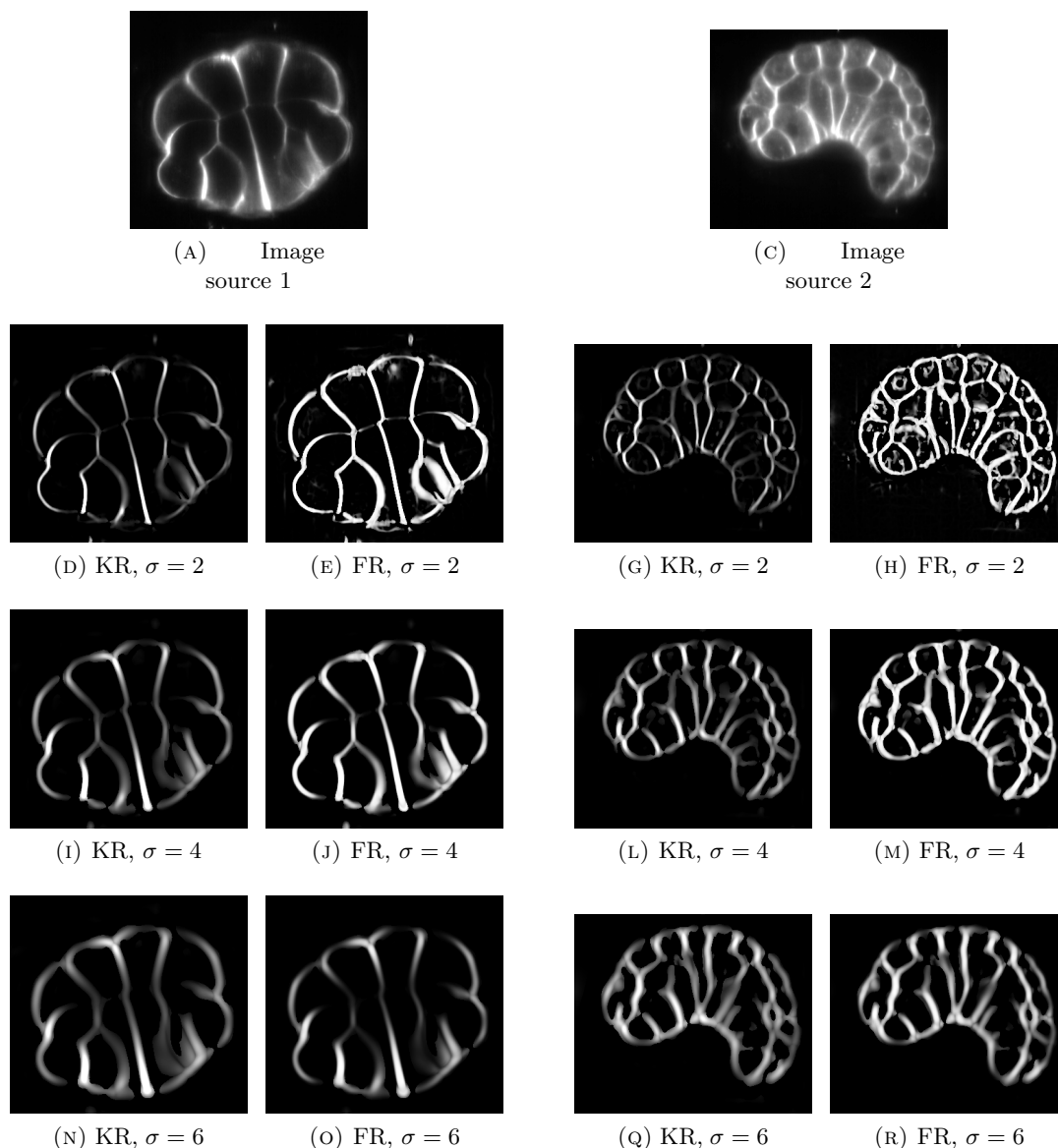


FIG. 3.5: Images d'origine et résultats des fonctions réponses Krissian-like (KR) et Frangi-like (FR) pour des échelles $\sigma \in \{2, 4, 6\}$.

où R désigne la fonction de réponse sur laquelle on procède à l'extraction des maxima. Le vecteur \mathbf{n}_1 (relatif au point P) a été précédemment calculé lors du calcul des réponses via le calcul des vecteurs propres des matrices hessiennes. De la même façon que pour le calcul de la fonction de réponse de type Krissian-like, il est nécessaire ici de réaliser une interpolation des valeurs de réponses pour que l'extraction des plans centraux soit bien réalisée.

On remarque dans la figure 3.6 que l'extraction réalisée sur la fonction de réponse de type Frangi-like apparaît plus dentelée que sur la fonction Krissian-like. Cette dernière fonction a été mise au point afin d'être optimale en les pixels de plans centraux, et apparaît comme étant bien adaptée au procédé d'extraction.

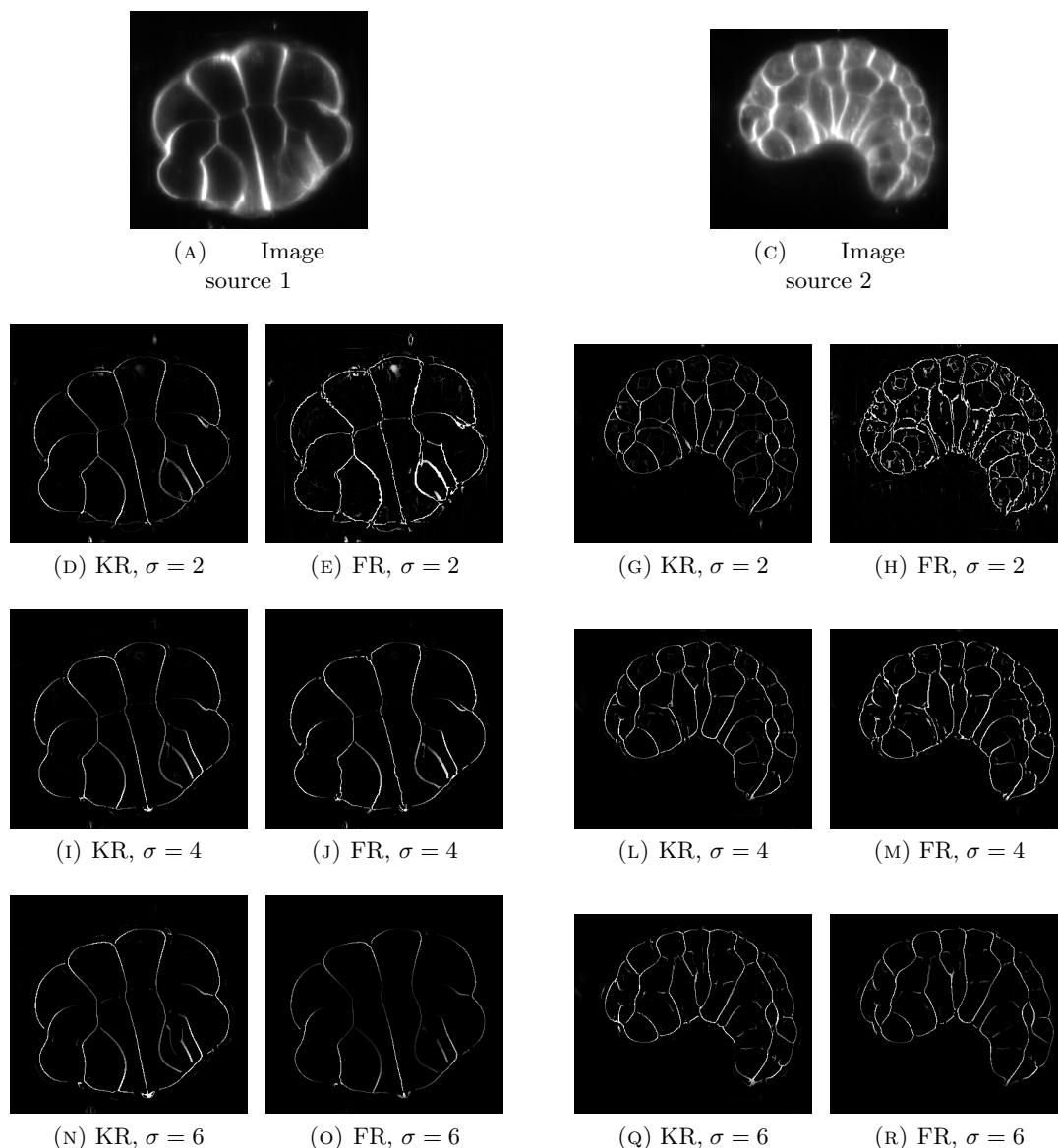


FIG. 3.6: Images d'origine et résultats des extractions des fonctions de réponse de type Krissian-like (KR) et Frangi-like (FR) pour des échelles $\sigma \in \{2, 4, 6\}$.

3.5 Binarisation des extrema

L'étape de binarisation des extrema n'est pas encore automatisée. Nous avons envisagé la combinaison de deux procédés pour réaliser cette étape : un seuillage par hystérésis et un filtrage par taille de composantes connexes. Ces deux procédés étaient déjà implémentés, la difficulté était donc de les exploiter au mieux pour réaliser une bonne segmentation des extrema.

La contrainte sur la taille des composantes connexes s'est avérée être inappropriée, car les structures correspondant à du bruit ou à du fond forment souvent un réseau de grande taille et possédant une réponse de planéité faible, et le paramètre de la taille des structures n'est pas pertinent ici.

Il reste donc deux paramètres sur lesquels on peut travailler : un seuil bas **sb** et un seuil haut **sh**. La binarisation de l'image est donc réalisée ainsi :

$$B(P) = \begin{cases} 0 & \text{si } E(P) < \mathbf{sb} \\ 1 & \text{si } E(P) \geq \mathbf{sh} \\ \text{istrue}\{\exists(P_1 \dots P_n) \text{ chemin continu } / P_1 = P \wedge E(P_n) \geq \mathbf{sh} \wedge E(P_i) \geq \mathbf{sb} \forall i\} & \text{sinon} \end{cases} \quad (3.5)$$

Le choix des seuils haut et bas s'est fait manuellement à l'occasion de ce stage et le résultat de ce seuillage est visible sur la figure 3.7, mais dans un futur travail il serait intéressant de procéder à un seuillage automatisé de l'image des extrema.

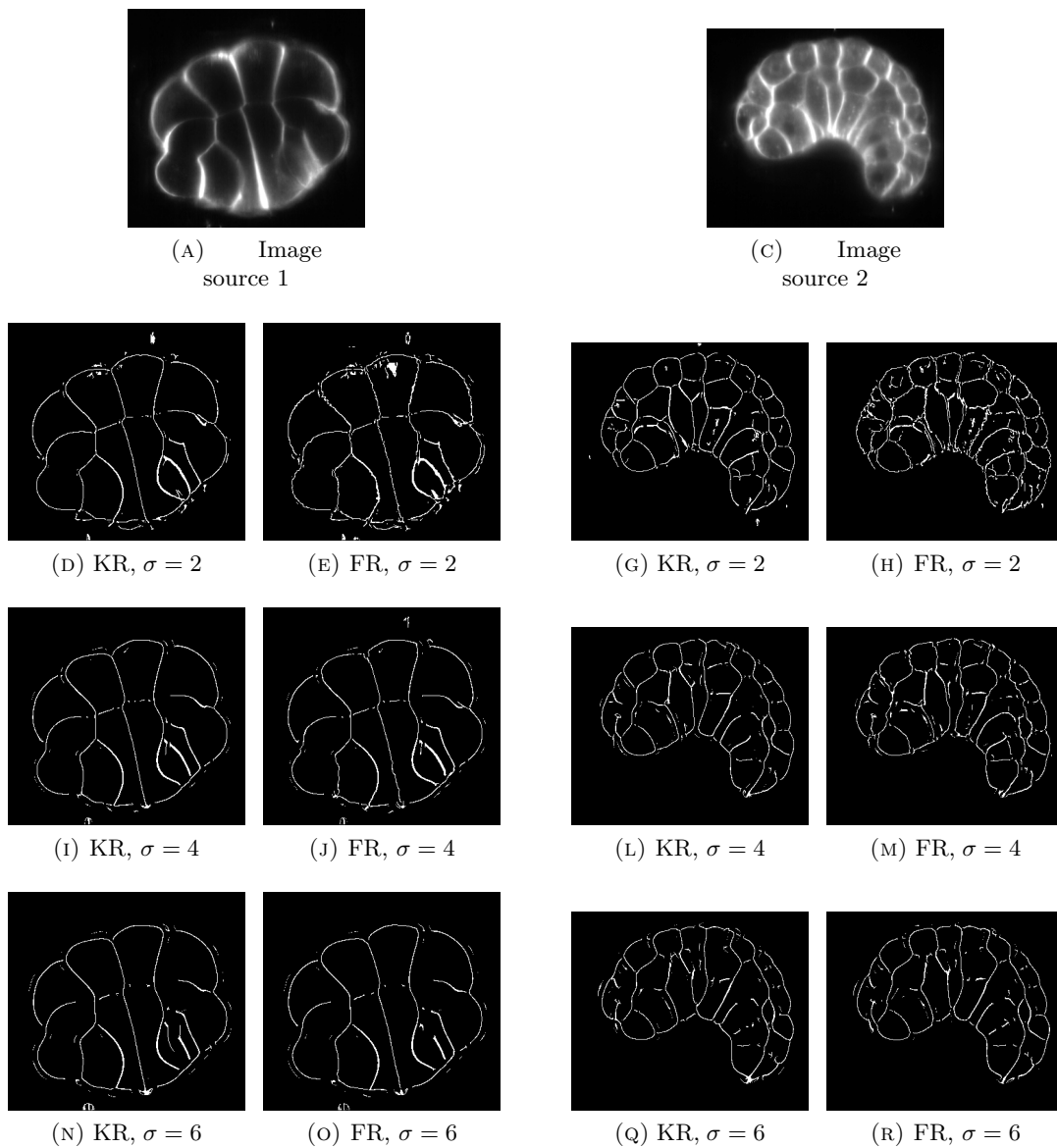


FIG. 3.7: Images d'origine et résultats de la binarisation des extrema calculés sur les réponses de type Krissian-like (KR) et Frangi-like (FR) pour des échelles $\sigma \in \{2, 4, 6\}$.

Chapitre 4

Tensor Voting

4.1 Introduction

Le procédé d'extraction des membranes cellulaires est mis à mal par le bruit des images ainsi que par les artefacts et les changements de contraste selon les directions, ce qui peut provoquer des trous dans les structures détectées. Une solution permettant de pallier à ce problème est de procéder à une *organisation perceptuelle* de l'information, qui permet de regrouper les structures d'une image selon des indices de proximité et de bonne continuation. L'organisation perceptuelle est une branche importante de la vision par ordinateur qui s'inspire des théories de la Gestalt (psychologie de la forme) [Koffka, 1935].

L'organisation perceptuelle, dans le cas des raccordements de discontinuités, consiste à regrouper des éléments isolés (primitives, ou tokens) suivant des modèles inspirés de critères perceptuels. Il existe trois classes de méthodes d'organisation perceptuelle : les méthodes locales, semi-locales et globales. Une méthode locale cherchera à regrouper des paires de tokens en se basant uniquement sur les informations liées à ces paires. Bien que rapide, ce type de méthode est cependant limité en terme de performance. Une méthode globale tient compte de l'ensemble des tokens pour effectuer les regroupements. La complexité algorithmique élevée de cette classe de méthode limite son applicabilité à notre problème. Une méthode semi-locale offre un compromis intéressant entre une approche locale et une approche globale, tout en conservant une complexité correcte. On trouvera dans [Tong et al., 2004] une bibliographie faisant état des principales méthodes d'organisation perceptuelle actuelles dans le contexte de l'imagerie 3D. On note toutefois qu'en comparaison avec le cas bidimensionnel, il existe aujourd'hui assez peu de telles méthodes fonctionnelles en 3D.

Le groupement des contours par une approche stochastique est proposé dans [Williams and Jacobs, 1995]. Cette méthode consiste à minimiser l'énergie induite par la complétion des tokens par un système de pas aléatoires. Une approche plus proche de celle que nous adopterons est celle proposée par [Yen and Finkel, 1998], qui fait décroître la force de la connexion avec la distance des tokens et la courbure du chemin qui les relie. Enfin, nous mentionnons une méthode qui regroupe les

fragments de courbes par l'utilisation de filtres en fonction de leur proximité et de la compatibilité de leur orientation [Dolan and Riseman, 1992]. Finalement, la seule méthodologie qui à notre connaissance permet de réaliser un groupement perceptuel dans un volume de données 3D de manière efficace est le *Tensor Voting* (vote par tenseurs) [Medioni et al., 2000]. Les tokens sont ici des structures paramétrées par des informations telles que leur position, leur courbure ou leur direction. La communication entre les tokens s'effectue dans un même espace de données, un champ de tenseurs d'ordre 2, qui a la force de pouvoir exprimer et comparer ces différents paramètres. Diverses méthodes basées sur les votes de tenseurs, des méthodes itératives [Loss et al., 2011] ou non-itératives [Mosaliganti et al., 2012] [Risser et al., 2008], sont proposées afin de raccorder les discontinuités dans les structures linéiques ou surfaciques d'images 2D ou 3D bruitées.

Face à l'enjeu de proposer un temps de calcul raisonnable, nous avons adopté une méthode non-itérative voisine de celle proposée par [Mosaliganti et al., 2012] et dont nous présentons le procédé dans ce chapitre.

4.2 Structure tenseur

4.2.1 Formalisme des tenseurs d'ordre 2

La méthode du Tensor Voting s'appuie sur une représentation tensorielle de l'information d'une image. À chaque voxel P est associé un tenseur d'ordre 2 noté $\mathbf{T}(P)$, c'est-à-dire une matrice $n \times n$ réelle, symétrique et positive, où I est de dimension n . On définit pour le reste du chapitre les éléments suivants :

- I : image initiale de dimension n .
- $\mathbf{T}(P) \in \mathbb{R}^{n \times n}$: tenseur symétrique positif associé au point P de l'image.
- $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$: valeurs propres associées au tenseur $\mathbf{T} \stackrel{\text{def}}{=} \mathbf{T}(P)$.
- $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$: vecteurs propres unitaires associées aux valeurs propres λ_i de \mathbf{T} .

La famille de vecteurs $\mathcal{B} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ forme une base orthonormée de l'espace \mathbb{R}^n en raison de la symétrie de \mathbf{T} .

Dans le cas de l'imagerie 3D, $n = 3$.

4.2.2 Décomposition en valeurs et vecteurs propres

En dimension 3, si \mathbf{T} est le tenseur d'ordre 2 associé au point P de l'image, on peut décomposer le tenseur ainsi :

$$\begin{aligned} \mathbf{T} &= \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T + \lambda_3 \mathbf{e}_3 \mathbf{e}_3^T \\ &= (\lambda_1 - \lambda_2) \underbrace{\mathbf{e}_1 \mathbf{e}_1^T}_{\mathbf{T}_S} + (\lambda_2 - \lambda_3) \underbrace{(\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T)}_{\mathbf{T}_P} + \lambda_3 \underbrace{(\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T + \mathbf{e}_3 \mathbf{e}_3^T)}_{\mathbf{T}_B} \end{aligned} \quad (4.1)$$

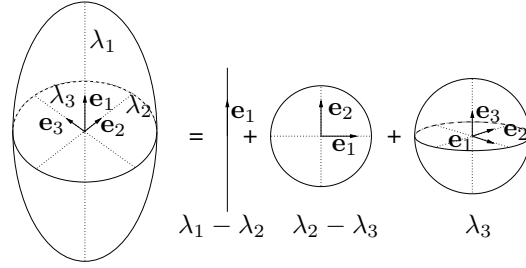


FIG. 4.1: Décomposition d'un tenseur du second ordre en composantes stick, plate et ball.

où dans la base \mathcal{B} ,

$$\mathbf{T}_S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{T}_P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{T}_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Les tenseurs \mathbf{T}_S , \mathbf{T}_P et \mathbf{T}_B sont des tenseurs appelés respectivement *tenseur stick*, *tenseur plate* et *tenseur ball* (figure 4.1).

- Un tenseur stick exprime une incertitude d'orientation du voxel dans le plan normal au vecteur propre \mathbf{e}_1 : il modélise un plan.
- Un tenseur plate exprime une incertitude d'orientation du voxel autour de l'axe \mathbf{e}_3 associé à la valeur propre nulle du tenseur : il modélise une structure linéique.
- Un tenseur ball n'exprime aucune préférence d'orientation du voxel : il modélise une intersection.

La décomposition du tenseur \mathbf{T} en somme de composantes *stick*, *plate* et *ball* permet d'exprimer combien le voxel associé au tenseur est en adéquation avec la présence d'une surface ($\lambda_1 - \lambda_2$), d'une courbe ($\lambda_2 - \lambda_3$) et d'une intersection (λ_3), et nous donne les informations directionnelles associées $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ où \mathbf{e}_1 désigne la normale à la composante surfacique et \mathbf{e}_3 indique l'axe de la composante linéique de \mathbf{T} .

Les trois catégories de tenseurs stick, plate et ball communiquent avec les voxels voisins grâce à une opération de vote qui permet de propager les structures de manière adéquate dans des directions privilégiées. À l'issue du vote de chacun des tenseurs, une nouvelle image de tenseurs \mathbf{J} est formée et la décomposition de chacun de ses éléments en composantes stick, plate et ball donne trois images respectivement d'adéquation avec une surface, une courbe ou une jonction.

4.3 Vote des tenseurs

Nous donnons dans cette section l'expression du vote des tenseurs de type stick, plate et ball. Le vote d'un tenseur dans le cas général peut être exprimé en combinant ces expressions à l'aide de la décomposition de l'équation 4.1.

4.3.1 Tenseur Stick

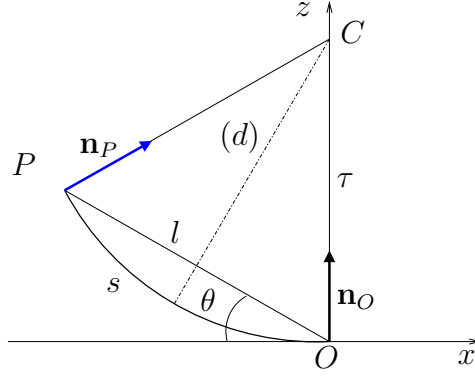


FIG. 4.2: Vote émis par un tenseur stick situé en O au point P .

Soit un tenseur stick unitaire $\mathbf{T}_S = \mathbf{n}_O \mathbf{n}_O^T$ et soit \mathbf{V}_S le vote reçu en un point P du plan (O, x, z) en provenance de \mathbf{T}_S . Le vecteur \mathbf{n}_O représente la normale à la surface qui passe par le point O . Si cette surface passe par le point P , alors le vecteur normal à la surface le plus vraisemblable en P est la normale à l'arc de cercle reliant O et P tangent au plan (O, x, y) (figure 4.2) :

$$\mathbf{V}_S(\overrightarrow{OP}, \mathbf{T}_S) = \mathbf{V}_S(\overrightarrow{OP}, \mathbf{n}_O) = \eta(\overrightarrow{OP}, \mathbf{n}_O) \mathbf{n}_P \mathbf{n}_P^T \quad (4.2)$$

où $\eta(\overrightarrow{OP}, \mathbf{n}_O)$ est la *fonction de décroissance* du vote, qui rend compte de l'influence décroissante du site O sur le site P avec la distance l les séparant et l'angle θ formé entre \overrightarrow{OP} et le plan (O, x, y) . Cette fonction de décroissance peut être définie de différentes façons. Nous discutons le choix de η dans la sections 4.4.

Le vecteur \mathbf{n}_P est le symétrique de \mathbf{n}_O dans le plan (O, x, z) par rapport à l'axe (d) et s'écrit :

$$\mathbf{n}_P = \left(\mathbf{I} - 2 \frac{\overrightarrow{OP} \overrightarrow{OP}^T}{l^2} \right) \mathbf{n}_O \quad (4.3)$$

4.3.2 Tenseur Ball

Le tenseur ball exprime l'absence de préférence d'orientation des données. On le fait voter en sommant l'ensemble des votes de tenseurs stick pris dans toutes les orientations de l'espace, ce qui traduit cette absence d'orientation privilégiée. On a :

$$\mathbf{V}_B(\overrightarrow{OP}) = \int_{\phi \in \mathcal{D}_B} \mathbf{V}_S(\overrightarrow{OP}, \mathbf{n}_\phi) d\phi \quad (4.4)$$

où $\{\mathbf{n}_\phi\}_{\phi \in \mathcal{D}_B}$ constitue l'ensemble des vecteurs unitaires de l'espace.

Ici, $\phi = (\phi_1, \phi_2)$ désigne ici le doublet des coordonnées sphériques du vecteur unitaire $\mathbf{n}_\phi = (\cos(\phi_1) \cos(\phi_2), \sin(\phi_1) \cos(\phi_2), \sin(\phi_2))^T$.

4.3.3 Tenseur Plate

Partant de la même approche que pour le tenseur ball, on fait voter le tenseur plate en sommant l'ensemble des votes de tenseurs sticks inclus dans le plan du tenseur. Dans le cas d'un tenseur plate normal au vecteur \mathbf{m} , le vote possède cette expression :

$$\mathbf{V}_P(\overrightarrow{OP}, \mathbf{T}_P) = \mathbf{V}_P(\overrightarrow{OP}, \mathbf{m}) = \int_{\phi \in \mathcal{D}_P(\mathbf{m})} \mathbf{V}_S(\overrightarrow{OP}, \mathbf{n}_\phi) d\phi \quad (4.5)$$

où $\{\mathbf{n}_\phi\}_{\phi \in \mathcal{D}_P(\mathbf{m})}$ constitue l'ensemble des vecteurs unitaires du plan normal à \mathbf{m} .

La somme des votes définie ainsi permet de privilégier la propagation du tenseur plate dans la direction portée par \mathbf{m} .

4.3.4 Cas général

Rappelons l'équation 4.1 :

$$\mathbf{T} = (\lambda_1 - \lambda_2)\mathbf{T}_S + (\lambda_2 - \lambda_3)\mathbf{T}_P + \lambda_3\mathbf{T}_B$$

Le vote $\mathbf{V}(\overrightarrow{OP}, \mathbf{T})$ du point O vers le point P est égal à la somme des votes de chacune des composantes stick, plate et ball du tenseur \mathbf{T} :

$$\mathbf{V}(\overrightarrow{OP}, \mathbf{T}) = (\lambda_1 - \lambda_2)\mathbf{V}_S(\overrightarrow{OP}, \mathbf{T}_S) + (\lambda_2 - \lambda_3)\mathbf{V}_B(\overrightarrow{OP}, \mathbf{T}_B) + \lambda_3\mathbf{V}_B(\overrightarrow{OP}) \quad (4.6)$$

4.4 Fonction de décroissance

4.4.1 Choix de la fonction de décroissance

Nous reprenons les notations de la figure 4.2 et de l'ensemble de la section 4.3.1.

La fonction de décroissance η est paramétrée par une unique variable σ qui est l'échelle de vote. Plus la valeur de σ sera élevée, plus un tenseur votant aura une échelle d'influence importante. Dans l'approche d'origine, [Medioni et al., 2000] propose d'exprimer η en fonction de la longueur s et de la courbure $\kappa = 1/\tau$ de l'arc qui relie le point émettant et le point recevant le vote :

$$\eta(\overrightarrow{OP}, \mathbf{n}_O) = \begin{cases} \exp\left(-\frac{s^2 + c\kappa^2}{\sigma^2}\right) & \text{si } |\theta| \leq \pi/4 \\ 0 & \text{sinon} \end{cases} \quad (4.7)$$

où

$$s = \begin{cases} \frac{\theta l}{\sin \theta} & \text{si } \theta \neq 0 \\ l & \text{sinon} \end{cases}, \quad \kappa = \frac{2 \sin \theta}{l}, \quad c = -16 \log(0.1)(\sigma - 1)\pi^{-2}$$

Le paramètre c fonction de σ sert au rapport d'influence de la proximité sur la courbure. La valeur de c augmente avec celle de σ pour pénaliser davantage les fortes courbures.

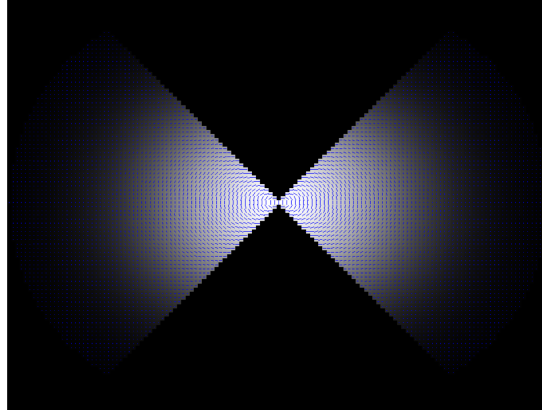


FIG. 4.3: Champ de vote généré par un tenseur stick vu en coupe. Les niveaux de gris donnent l'amplitude du vote et l'orientation des votes reçus est donnée par les bâtonnets bleus.

4.4.2 Limitation

Une difficulté du Tensor Voting tel qu'il est défini dans [Medioni et al., 2000] est que seule l'expression du vote d'un tenseur stick est connue analytiquement, et le calcul des champs de vote de type Ball et Plate se fait en réalisant des sommes discrètes de champs de tenseurs Stick pré-calculés. Le résultat des votes de tenseurs Ball et Plate est donc sensible à la discrétisation choisie pour les champs Stick.

D'autres formes de fonctions de décroissance ont été étudiées et proposées. L'une d'entre elles a attiré notre attention car elle propose une formulation acceptant une solution analytique pour un tenseur votant quelconque. Cependant, notre étude semble révéler des inexactitudes dans ce travail présenté en annexe B.

4.5 Processus de vote

La première étape du Tensor Voting est l'initialisation de l'ensemble des tenseurs \mathbf{T} associés à l'image d'origine I . Ces tenseurs peuvent être nuls, orientés (sticks ou plates) ou non-orientés (balls), en fonction de l'information dont on dispose au départ.

La deuxième étape est le procédé de votes. Chaque tenseur \mathbf{T} non-nul émet un champ de votes (section 4.6) dans son voisinage et chaque point de ce voisinage reçoit son vote. Soit U l'image de tenseurs qui résulte de cette étape de votes :

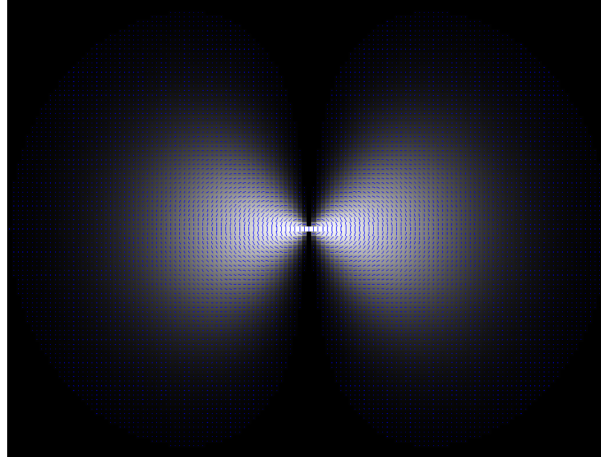


FIG. 4.4: Exemple de champ de votes émis par un tenseur. En nuances de gris, l'intensité du vote émis (blanc/noir : intensité maximale/minimale). En bleu, l'orientation privilégiée \mathbf{e}_1 associé au vote reçu en chaque point.

$$\mathbf{U}(P) = \sum_{O \in N(P)} \mathbf{V}(\overrightarrow{OP}, \mathbf{T}(O)) \quad (4.8)$$

où $N(P)$ désigne un voisinage du point P et $\mathbf{T}(O)$ est le tenseur votant au point O .

4.6 Champs de vote

L'équation 4.6 montre que pour faire voter un tenseur \mathbf{T} , il suffit de connaître le vote de ses composantes stick, plate et ball, qui dépend d'une orientation pour les deux premières et qui est pondéré par un coefficient de prépondérance dans chacun des cas.

Les champs de votes plate et ball s'obtiennent en intégrant le champ de vote stick unitaire selon les équations 4.5 et 4.4. Cette intégration ne possède pas de résultat analytique (annexe B). Leur calcul passe donc par une discrétisation des directions de l'espace, afin de procéder à une somme finie de champs de tenseurs sticks pour obtenir des approximations de champs plate et ball.

Tandis que le champ de tenseurs ball est non-orienté et ne nécessite d'être calculé qu'une seule fois, il y a autant de champs de tenseurs plates qu'il y a de directions dans l'espace, et il faut discrétiser les directions des plans normaux pour chaque calcul de champ plate.

Un calcul systématique des champs de tenseurs plate et stick correspondants aux directions des tenseurs votants serait un choix d'implémentation particulièrement coûteux. Une solution beaucoup plus économe en temps de calcul est de réaliser une table de correspondance : on effectue une discrétisation des directions de l'espace et on fait correspondre à chacune de ces directions un champ de tenseurs stick et un champ de tenseurs plate. Ainsi lors de l'étape de vote, la composante stick d'un tenseur vote à la manière du champ correspondant à l'orientation la plus proche de celle du tenseur, et de même pour la composante plate.

Le calcul des champs de tenseurs plate et ball pose la question de la discrétisation des directions respectivement d'un plan et d'un espace, c'est-à-dire la discrétisation d'un demi-cercle (figure 4.5) et d'une demi-sphère (car l'opposé d'un vecteur donne la même direction que celui-ci). La discrétisation des directions pour la table de correspondance est la même que celle utilisée pour le calcul du champ de tenseurs ball.

4.6.1 Discrétisation d'un demi-cercle

D'après l'équation 4.5, si \mathbf{T}_P est le tenseur plate normal au vecteur \mathbf{m} , la discrétisation de l'intégrale donne :

$$\mathbf{V}_P(\overrightarrow{OP}, \mathbf{T}_P) = \mathbf{V}_P(\overrightarrow{OP}, \mathbf{m}) = \sum_{\phi \in \tilde{\mathcal{D}}_P(\mathbf{m})} \mathbf{V}_S(\overrightarrow{OP}, \mathbf{n}_\phi) d\phi \quad (4.9)$$

où $\tilde{\mathcal{D}}_P$ est la discrétisation du \mathcal{D}_P de l'équation 4.5, c'est-à-dire la discrétisation du cercle du plan normal à \mathbf{m} .

Si N est le nombre de points par lequel on souhaite discrétiser le demi-cercle, on a :

$$\tilde{\mathcal{D}}_P = \{\cos(i\pi/N)\mathbf{n}_1 + \sin(i\pi/N)\mathbf{n}_2\}_{i=0..N-1} \quad (4.10)$$

où $\{\mathbf{n}_i\}_{1,2}$ est tel que $\{\mathbf{m}, \mathbf{n}_1, \mathbf{n}_2\}$ forme une base orthonormée de l'espace.

Réalisée de cette façon, la discrétisation des directions normales à \mathbf{m} est régulière et on est libre de choisir une discrétisation avec autant de directions que l'on veut.

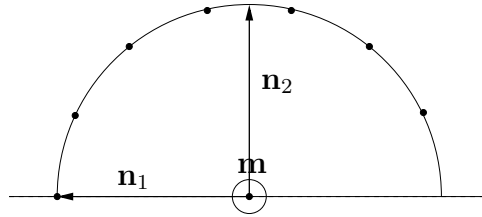


FIG. 4.5: Discrétisation du demi-cercle ($N = 7$).

4.6.2 Discrétisation d'une demi-sphère

La discrétisation de la demi-sphère est plus délicate à appréhender que celle du demi-cercle, en raison de l'importance d'avoir une répartition régulière des directions de l'espace. Une approche naïve est de procéder par une discrétisation des angles des coordonnées sphériques (figure 4.6), car la discrétisation est d'autant plus dense qu'elle se rapproche des pôles de la sphère.

Une solution pour obtenir une meilleure discrétisation des directions de l'espace est de procéder par itération :

- On construit un octaèdre régulier (figure 4.7a) inscrit dans la sphère unité à discrétiser.
- Pour chaque itération :

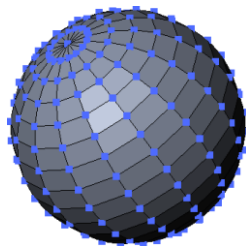
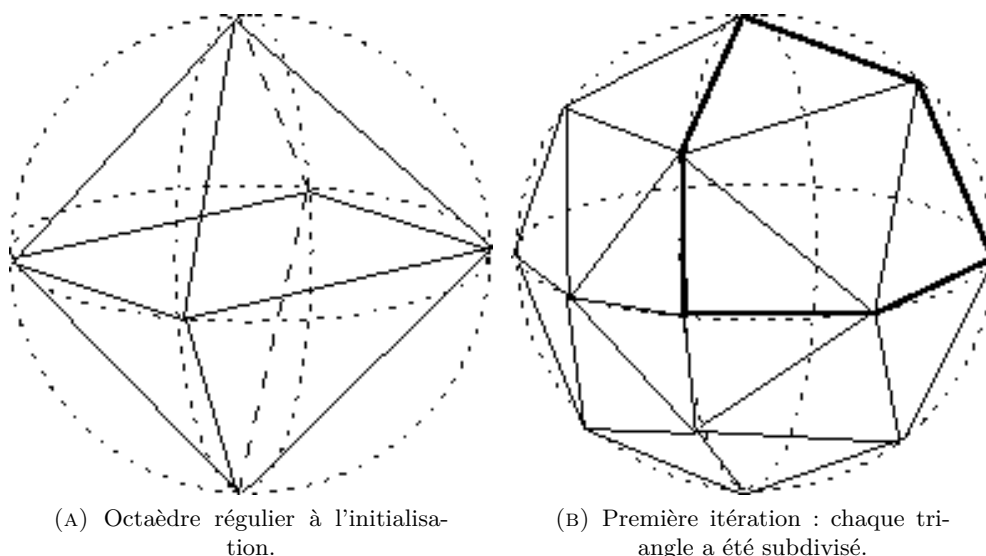


FIG. 4.6: Discrétisation non-régulière d'une sphère : la densité des points discrétisés est bien plus grande en les pôles de la sphère qu'au niveau de l'équateur.

- chaque face du polyèdre est subdivisée en 4 sous-faces ;
- tous les points créés lors de l'itération sont projetés sur la sphère unité (figure 4.7b).



(A) Octaèdre régulier à l'initialisation.

(B) Première itération : chaque triangle a été subdivisé.

FIG. 4.7: Discrétisation de la sphère.

Ce procédé ne permet pas d'obtenir un maillage régulier de la sphère mais est toutefois simple à implémenter et permet une répartition suffisamment équilibrée pour notre application.

On récupère la discrétisation de la demi-sphère en conservant les points vérifiant :

- une abscisse strictement positive **ou**
- une abscisse nulle et une ordonnée strictement positive **ou**
- une abscisse et une ordonnée nulles et une cote positive.

4.7 Application à la détection des parois cellulaires et résultats

Dans le chapitre 2, nous avons vu qu'il existait déjà un travail exploitant le Tensor Voting pour faire de la détection de membranes cellulaires. Notre objectif était de contribuer à ce travail en proposant une façon de procéder à cette étape de manière plus rapide. Pour cela, nous avons cherché à limiter le nombre de tenseurs votant (*i.e.* non-nuls après initialisation de l'image de

tenseurs) car le temps d'exécution du Tensor Voting est linéairement dépendant du nombre de tenseurs votants.

Nous avons donc initialisé l'image de tenseurs de deux façons :

1. avec l'image des extrema issue de l'équation 3.4;
2. avec la binarisation des extrema (équation 3.5).

Soit $M \in \{E, B\}$, et soit $\mathbf{T}(P)$ le tenseur initial associé au point P de l'image :

$$\mathbf{T}(P) = \begin{cases} \mathbf{0}_{3 \times 3} & \text{si } M(P) = 0 \\ M(P)\mathbf{e}_1\mathbf{e}_1^T & \text{sinon} \end{cases} \quad (4.11)$$

où \mathbf{e}_1 est le vecteur associé à la valeur propre la plus grande en valeur absolue de la matrice hessienne calculée en P .

Cette initialisation signifie que tous les tenseurs non-nuls sont des tenseurs stick, donc seuls les champs de vote de type stick sont utilisés. Après les votes, les tenseurs obtenus sont décomposés en valeurs propres afin d'en déduire l'image des $S = (\lambda_1 - \lambda_2)_{P \in \mathcal{D}(I)}$.

Nous avons également appliqué l'initialisation proposée par [Mosaliganti et al., 2012], à savoir en prenant $M = R$, afin de pouvoir comparer à la fois l'aspect temps de calcul et l'aspect résultat de la détection. On discute ces résultats dans la section 2.2.5.3. La figure 4.8 présente les résultats du Tensor Voting pour différentes initialisations.

Nous avons procédé à une implémentation classique du Tensor Voting car dans une première phase de développement, nous souhaitons nous assurer qu'une valeur ajoutée était apportée par ce processus. On note toutefois qu'il existe des variantes de Tensor Voting visant à optimiser son coût algorithmique [Jia et al., 2012] [Franken et al., 2006] [Min and Medioni, 2006]. Nous nous sommes en particulier intéressés à [Jia et al., 2012] qui est discuté dans l'annexe B, mais la méthode proposée paraît erronée.

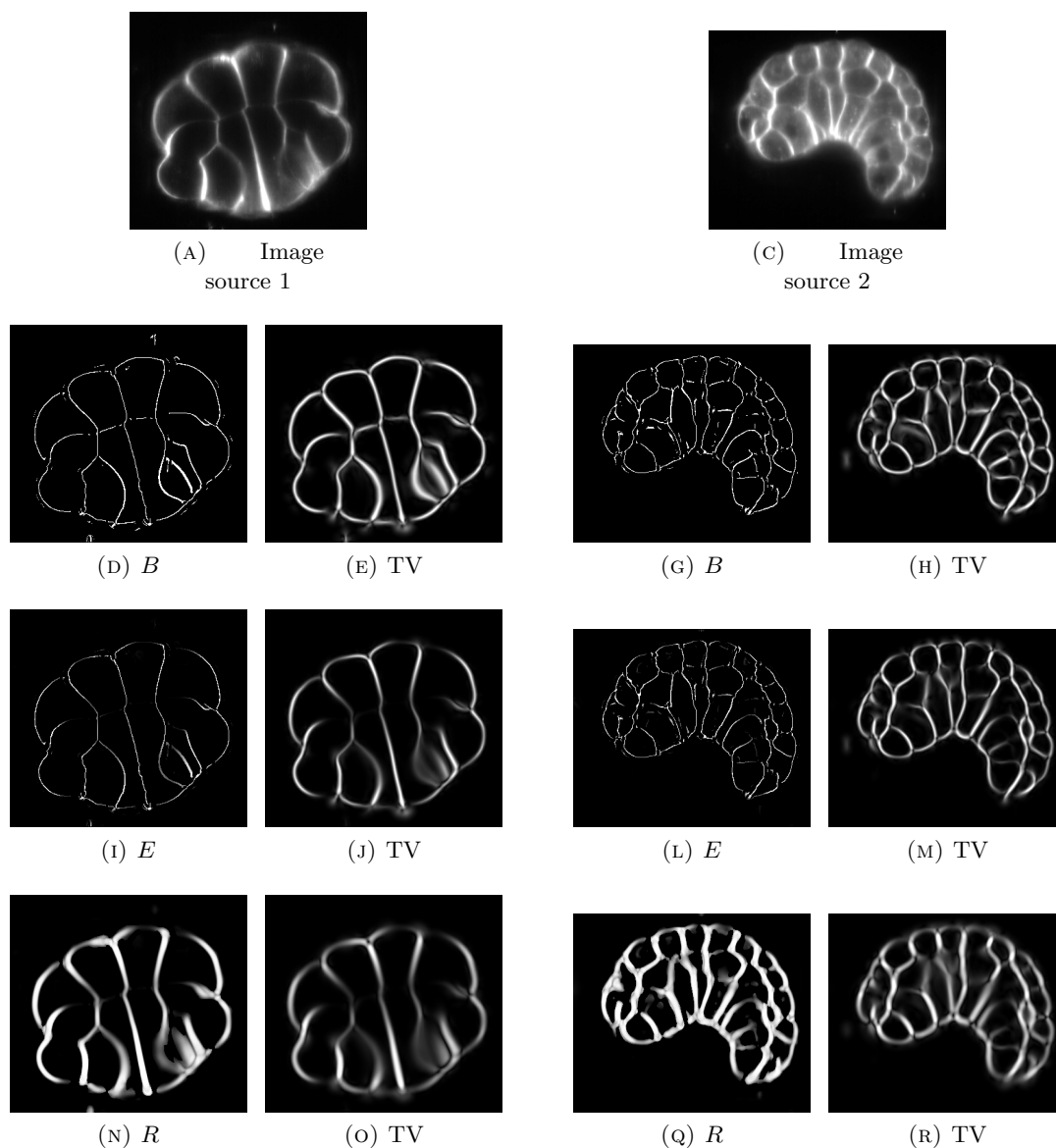


FIG. 4.8: Résultats du Tensor Voting pour différentes entrées. $\{R, E, B\}$: initialisation de l'image de tenseurs par la fonction donnée pour une échelle $\sigma_R = 4$; TV : Tensor Voting réalisé avec $\sigma_{TV} = 10$ (l'image affichée est l'image des surfaces S).

Chapitre 5

Segmentation des cellules

5.1 Introduction

En dernière partie de stage, le travail s'est porté sur l'exploitation des précédents résultats pour la segmentation des cellules. Dans ce chapitre, la majeure partie des fonctions présentées était déjà implémentée et seules quelques fonctions relatives au test ou à des post-traitements élémentaires ont été implémentées. L'analyse des résultats a été programmée en Matlab car l'environnement qu'offre cet outil permet d'extraire et de visualiser aisément de nombreuses informations issues d'un tableau de confusion (voir section [2.2.5.2](#)).

5.2 Ligne de partage des eaux

L'approche que nous souhaitons adopter pour la segmentation des cellules est la transformation de la ligne de partage des eaux (watershed en anglais). Il s'agit de considérer une image en niveaux de gris comme étant un relief topographique, et de calculer les « lignes de partage des eaux » de ce relief, qui correspondent aux crêtes de l'image. Les régions segmentées correspondent alors aux « bassins versants ». Nous utilisons un algorithme par immersion [Vincent and Soille, 1991] pour réaliser le watershed (figure 5.1).

La principale difficulté du watershed est qu'il y a autant de régions segmentées que de minima locaux dans l'image, ce qui provoque le plus souvent un problème de sur-segmentation de l'image.

Une solution est de pré-calculer les « minima » qui seront utilisés pour le procédé d'immersion afin de contrôler leur nombre et leur localisation. Ces « minima » sont appelés *graines*, et de ces graines proviennent les différentes régions segmentées. Le calcul de ces graines est donc l'étape clé de la ligne de partage des eaux.

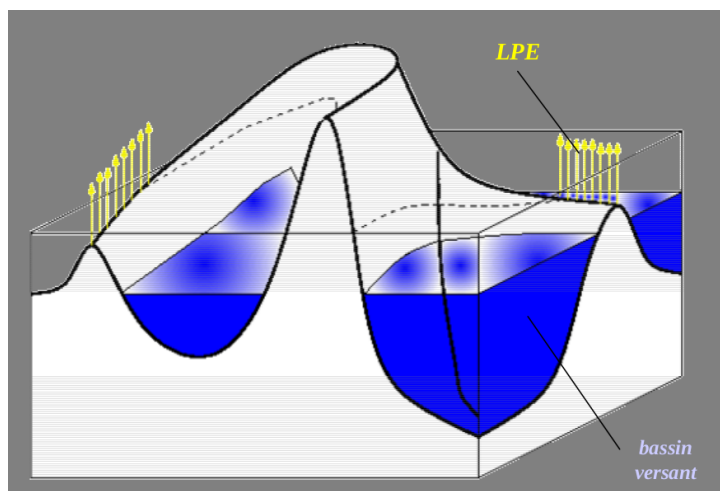


FIG. 5.1: Ligne de partage des eaux : l'ensemble des crêtes forme la LPE, l'ensemble des bassins versants forme les régions segmentées. Illustration d'un cours de Morphologie Mathématique de l'ENSTA (http://wwdf.fr/Cours/docs/ESE42/Ese4_chap5.pdf).

5.3 Choix des graines de régions

Avant de procéder à la ligne de partage des eaux, une série de calculs morphologiques sont effectués pour obtenir les graines qui déterminent les positions des futures cellules détectées (figure 5.2). Ces cellules sont calculées sur l'image S qui résulte du Tensor Voting convertie en image de niveaux de gris 8 bits (0-255).

5.3.1 Minima régionaux

De manière à prendre en considération la dynamique des bassins versants, c'est-à-dire leur profondeur, on effectue une binarisation des h -minima de l'image. Le choix du paramètre h_{min} est important car cela fixe une profondeur minimale en dessous de laquelle un minimum local ne constituera pas une région à part entière (figure 5.3).

Nous choisissons un paramètre h_{min} relativement élevé. Ainsi, les minima obtenus sont largement étendus et les zones non marquées correspondent aux positions des membranes (figure 5.4). On note l'image binaire obtenue H_{min} , où :

$$H_{min}(P) = \begin{cases} 1 & \text{si } P \text{ est dans un } h\text{-minimum de } S \\ 0 & \text{sinon} \end{cases} \quad (5.1)$$

5.3.2 Maxima de distance

La faiblesse du procédé précédent est que l'image S a une réponse très faible au niveau des jonctions de membranes, en raison de la perte de la propriété de structure planaire en ces positions. La répercussion de ce phénomène est que le plus souvent, les h -minima de cellules

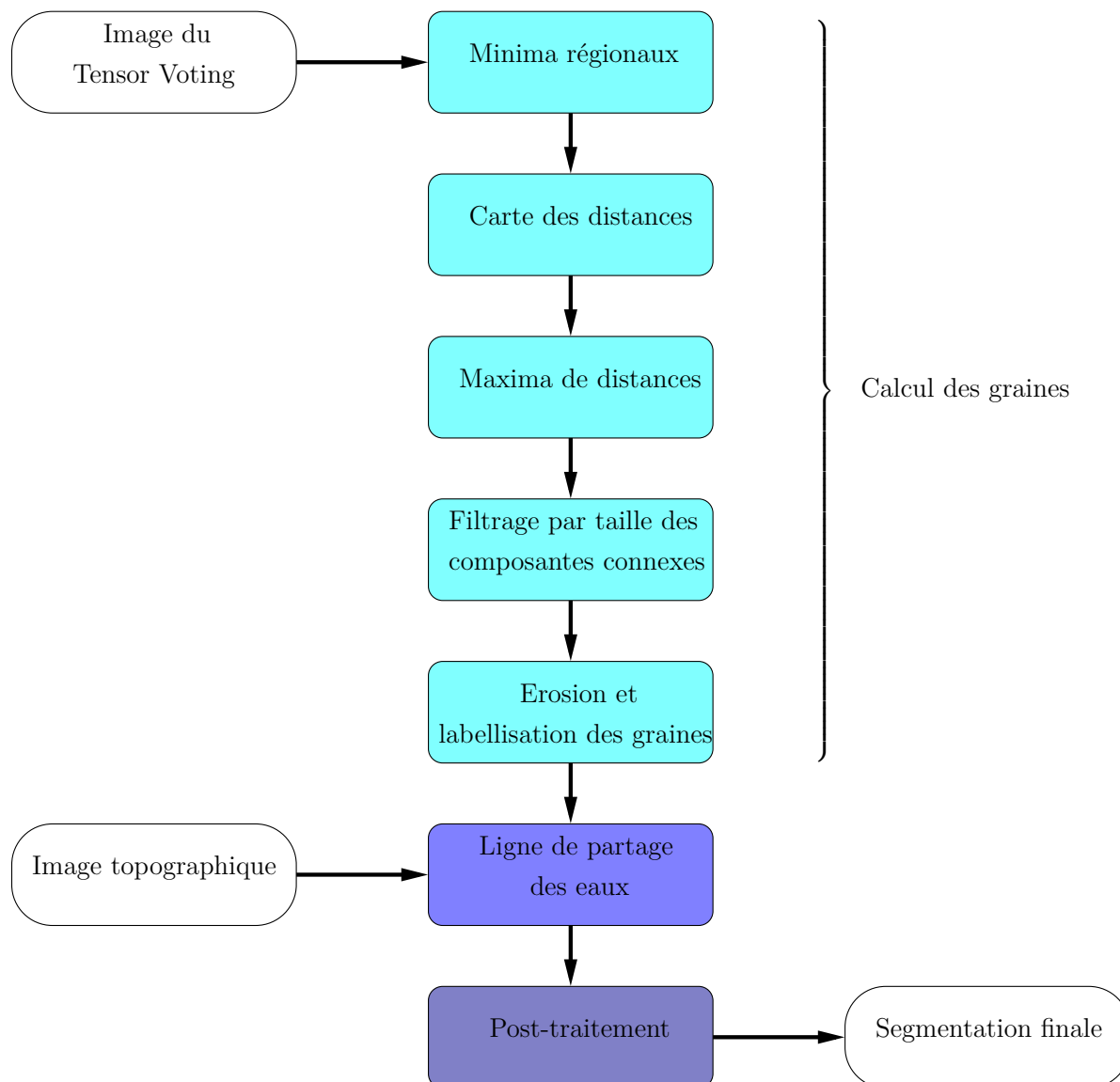


FIG. 5.2: Vue d'ensemble : calcul des graines pour la ligne de partage des eaux, LPE et post-traitement.

voisines seront reliés par un mince passage au niveau de ces jonctions, obligeant ainsi à effectuer un traitement sur l'image des h -minima.

Nous avons d'abord envisagé un filtrage par érosion des graines, qui permet idéalement de supprimer les connexions au niveau des jonctions et donc de séparer les h -minima qui doivent l'être. Cette technique n'était finalement pas bien adaptée à notre problème car il fallait procéder à une érosion trop importante qui risquait d'éliminer complètement certaines graines et de provoquer une sous-segmentation dans le résultat final.

Finalement, nous avons choisi de calculer l'image des distances D de chacun des pixels à une zone de membrane et de récupérer l'image H_{max} des h -maxima sur cette image des distances

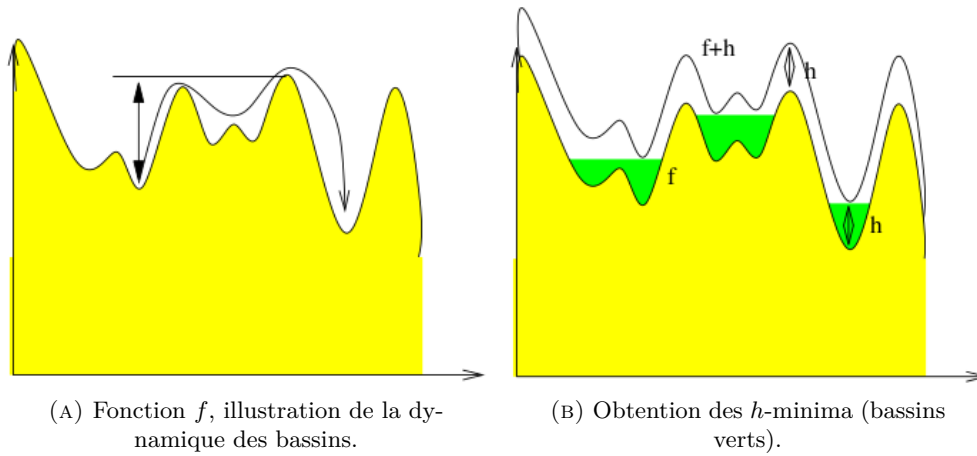


FIG. 5.3: h -minima. Un choix de h plus petit conduirait à l'obtention de davantage de minima (jusqu'à 5 dans cet exemple).

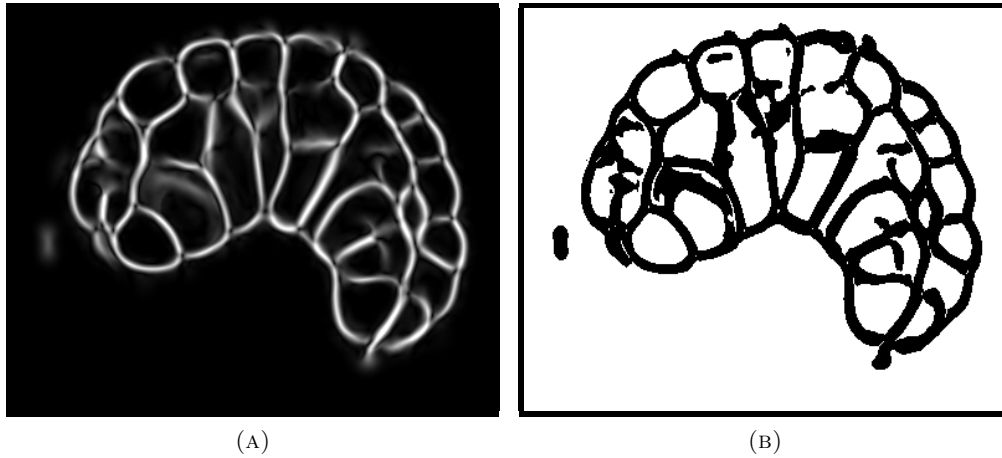


FIG. 5.4: Minima régionaux (5.4b) sur une image d'embryon (5.4a).

(basé sur exactement le même principe que les h -minima). On a :

$$D(P) = \min\{\text{dist}(P, Q)\}_{Q \in H_{min}^{-1}\{0\}} \quad (5.2)$$

$$H_{max}(P) = \begin{cases} 1 & \text{si } P \text{ est dans un } h\text{-maximum de } D \\ 0 & \text{sinon} \end{cases} \quad (5.3)$$

où la fonction dist exprime la distance de chanfrein, qui présente le double avantage d'être très rapide à calculer et de proposer une excellente alternative à la distance euclidienne [Thiel and Montanvert, 1992]. Le choix du paramètre h_{max} pour les maxima régionaux est moins critique que le choix précédent du h_{min} . On choisit h_{max} suffisamment petit pour pouvoir séparer les graines qui étaient connectées au niveau des jonctions. On obtient les résultats de la figure 5.5.

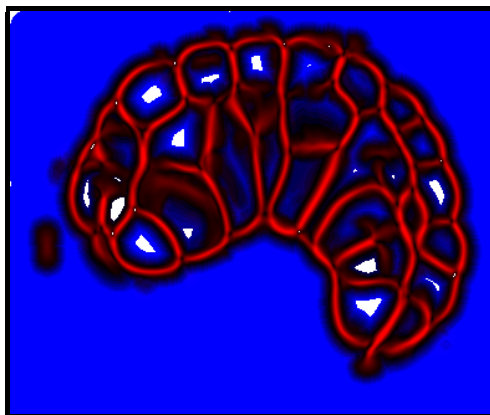
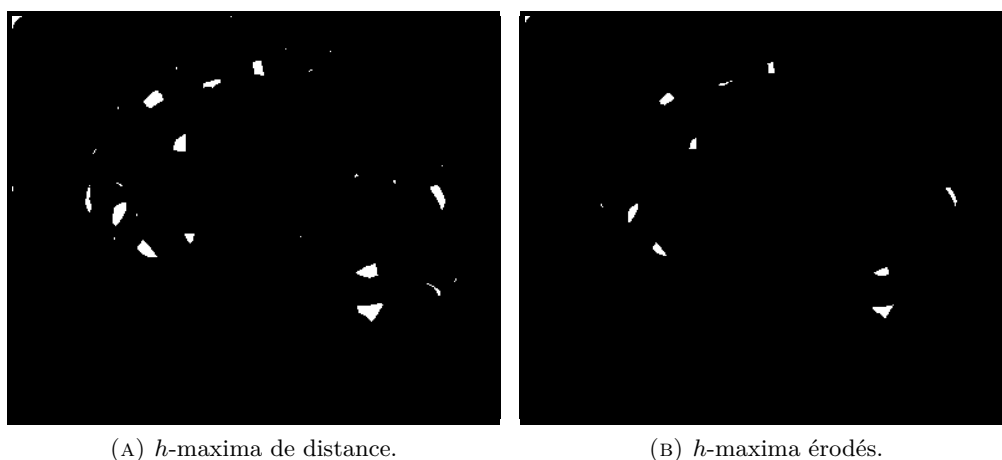


FIG. 5.5: Carte des distances (en bleu) et h -maxima de distance (en blanc) superposés à l'image S du Tensor Voting. On remarque la présence de multiples h -maxima au niveau des jonctions de cellules.

5.3.3 Érosion, taille des composantes connexes et labellisation

En raison du codage de la fonction des h -maxima, on obtient une image H_{max} qui présente de nombreuses composantes connexes surfaciques qu'on souhaite pouvoir éliminer de la liste des graines calculées. Pour cela, on procède à une légère érosion de H_{max} qui conserve les composantes volumiques en éliminant les surfaciques, linéiques et ponctuelles. L'image résultante est l'image des graines finales qu'on utilise pour le watershed (figure 5.6). Il ne reste qu'à numéroter (ou labelliser) les graines. Un filtrage par taille de composantes connexe n'est pas indispensable lors de ce traitement car l'érosion a permis l'élimination des composantes non-conformes.



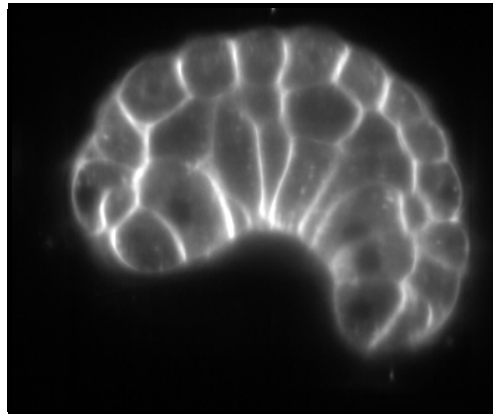
(A) h -maxima de distance.

(B) h -maxima érodés.

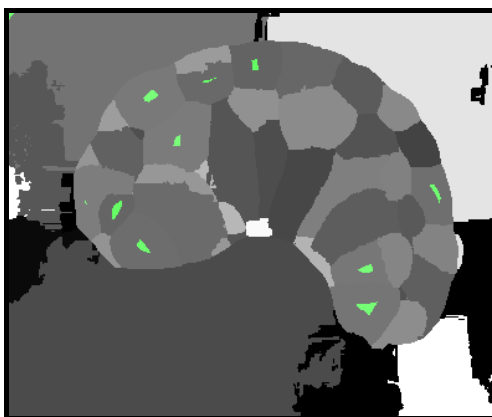
FIG. 5.6: Érosion des h -maxima. On remarque que les petites composantes au niveau des jonctions de cellules sont bien supprimées.

5.4 Résultats

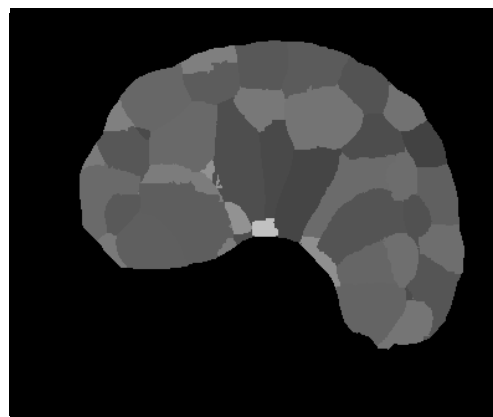
La figure 5.7 présente le résultat final de la segmentation d'une image originale réelle, ainsi que la vérité terrain qui nous est donnée pour cette image. Il n'est pas évident de procéder à une analyse qualitative du résultat, c'est pourquoi nous avons réalisé le plan de tests présenté en section 2.2.5.3 qui permet une évaluation quantitative des segmentations. Ainsi, les segmentations sont présentées sous la forme de diagrammes indiquant le nombre de bonnes détections, de sur-détections et de sous-détections pour chacun des tests réalisés (figure 5.8). Nous avons aussi pu évaluer les segmentations obtenues par une autre méthode de calcul des graines grâce aux données que nous a fourni un doctorant de l'Université Montpellier 2, Léo Guignard.



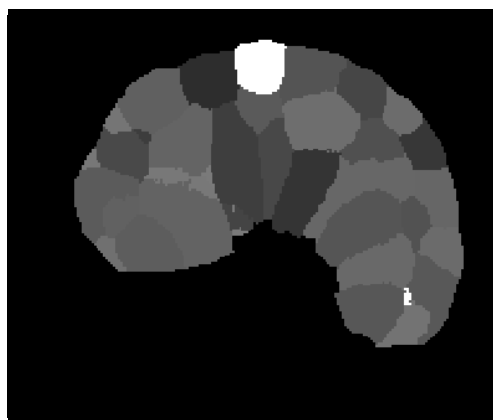
(A) Image initiale.



(B) Résultat de la ligne de partage des eaux.



(C) Post-traitement de la segmentation.



(D) Vérité terrain.

FIG. 5.7: Watershed appliqué à l'image S lissée par un convolution gaussienne. Les graines des cellules apparaissent en vert sur la figure 5.7b. En 5.7c, la segmentation à laquelle on a fusionné les régions touchant le bord de l'image (hypothèse de départ : l'embryon est entièrement inclus dans l'image).

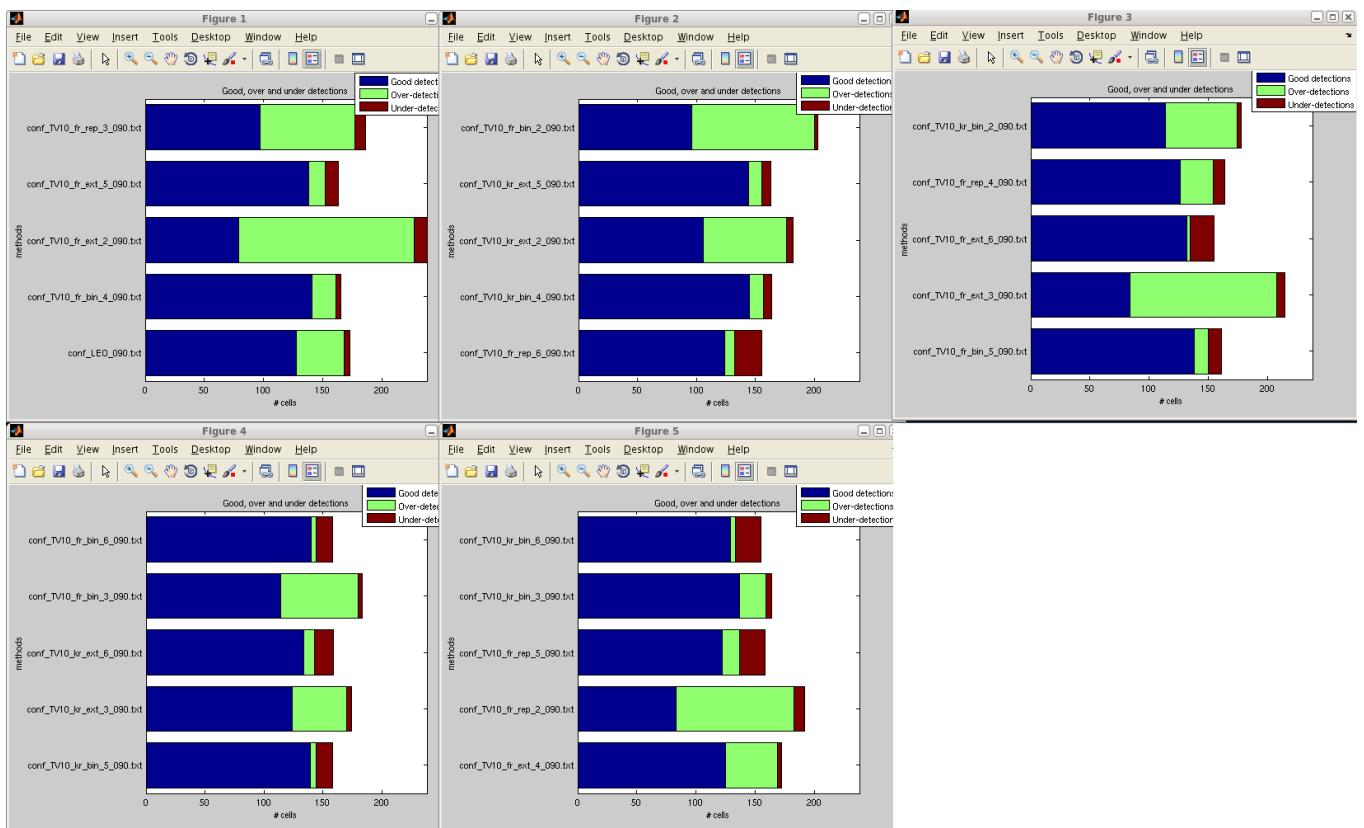


FIG. 5.8: Diagrammes évaluant les résultats obtenus.

Chapitre 6

Conclusion et perspectives

6.1 Apports du travail réalisé

Le travail réalisé tout au long de ce projet de fin d'études ouvre des perspectives intéressantes dans le domaine de l'analyse d'images appliquée à l'étude de la morphogenèse en biologie. Les résultats obtenus valident l'idée qu'une association originale d'approches de différentes natures (détection de structures planaires, techniques de groupement perceptuel, algorithmes de segmentation basés région) peuvent aboutir à une amélioration significative de la qualité des segmentations de cellules dans le contexte de l'imagerie 3D.

En plus des nombreuses mises au point à effectuer pour permettre d'automatiser complètement le processus de segmentation des cellules, deux enjeux majeurs se dégagent pour la suite de ce travail. D'abord, il sera essentiel de rendre plus efficace d'un point de vue algorithmique les étapes de détection de structures planaires et surtout de groupement perceptuel (Tensor Voting). En effet, en particulier pour cette dernière étape, il n'est pas envisageable que pour le traitement d'une image 3D de résolution moyenne, le calcul soit effectué en plus d'une heure dans l'optique de son application à des problèmes de taille plus grande et d'une utilisation sur une longue séquence d'images (film). Enfin, et cela va de paire avec le dernier point, le principal objectif à réaliser sera d'appliquer les procédés que nous proposons pour réaliser un pistage (tracking) des cellules tout au long de leur évolution lors des premières heures de vie de l'embryon en considérant la dimension temporelle du problème.

6.2 Enrichissement personnel

Ces six mois de stage ont été l'occasion pour moi de m'initier à de nombreux outils de développement (valgrind, svn, git, CMake, CCMake), de renforcer mes connaissances dans les langages C, Matlab, mais aussi Maple et Shell, et de découvrir quelques logiciels de visualisation 3D (ImageJ, Amira). Mes connaissances en traitement d'image ont également été fortement enrichies à l'occasion de mes lectures d'articles, de mes entrevues avec mon encadrant Grégoire Malandain, et également à l'occasion de l'école de Printemps à laquelle j'ai eu la chance d'assister

au mois de Mai et qui s'inscrivait dans le thème des nouvelles tendances en vision par ordinateurs, présentant des approches variationnelles et probabilistes.

Enfin, le cadre offert par ce stage, à Sophia Antipolis, pousse à faire des rencontres enrichissantes avec des personnes provenant des quatre coins du monde. J'ai le sentiment d'avoir mis cela à profit afin d'améliorer mes compétences en anglais (ce dont j'avais bien besoin), ainsi qu'en communication orale, grâce à 3 présentations que j'ai effectuées au cours de ce stage avant ma soutenance, et écrite.

Bibliographie

- [de Monvel et al., 2001] de Monvel, J. B., Calvez, S. L., and Ulfendahl, M. (2001). Image restoration for confocal microscopy: Improving the limits of deconvolution, with application to the visualization of the mammalian hearing organ. *Biophysical Journal*, 80(5):2455 – 2470.
- [Dolan and Riseman, 1992] Dolan, J. and Riseman, E. (1992). Computing curvilinear structure by token-based grouping. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 264–270.
- [Fernandez et al., 2010] Fernandez, R., Das, P., Mirabet, V., Moscardi, E., Traas, J., Verdeil, J.-L., Malandain, G., and Godin, C. (2010). Imaging plant growth in 4d: robust tissue reconstruction and lineaging at cell resolution. *Nature Methods*, 7:547–553.
- [Fischer et al., 2007] Fischer, S., Bayerl, P., Neumann, H., Redondo, R., and Cristóbal, G. (2007). Iterated tensor voting and curvature improvement. *Signal Process.*, 87(11):2503–2515.
- [Franken et al., 2006] Franken, E., van Almsick, M., Rongen, P., Florack, L., and ter Haar Romeny, B. (2006). An Efficient Method for Tensor Voting Using Steerable Filters. In Leonardis, A., Bischof, H., and Pinz, A., editors, *9th European Conference on Computer Vision (ECCV 2006)*, volume 3954 of *Lecture Notes in Computer Science*, pages 228–240–240. Springer Berlin / Heidelberg.
- [Jia et al., 2012] Jia, J., Yeung, S.-K., Wu, T.-P., Tang, C.-K., and Medioni, G. (2012). A closed-form solution to tensor voting: Theory and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1482–1495.
- [Keller, 2013] Keller, P. J. (2013). Imaging morphogenesis: technological advances and biological insights. *Science*, 340(6137):1234168.
- [Koffka, 1935] Koffka, K. (1935). *Principles of Gestalt psychology*. Routledge Lond.
- [Kovesi, 1997] Kovesi, P. (1997). Symmetry and asymmetry from local phase. In *Tenth Australian Joint Convergence on Artificial Intelligence*, pages 2–4.
- [Krissian et al., 2000] Krissian, K., Malandain, G., Ayache, N., Vaillant, R., and Troussset, Y. (2000). Model-Based Detection of Tubular Structures in 3D Images. *Computer Vision and Image Understanding*, 80(2):130–171.

- [Loss et al., 2011] Loss, L., Bebis, G., and Parvin, B. (2011). Iterative tensor voting for perceptual grouping of ill-defined curvilinear structures. *Medical Imaging, IEEE Transactions on*, 30(8):1503–1513.
- [Luengo-Oroz et al., 2007] Luengo-Oroz, M., Faure, E., Lombardot, B., Sance, R., Bourguine, P., Peyrieras, N., and Santos, A. (2007). Twister segment morphological filtering. a new method for live zebrafish embryos confocal images processing. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 5, pages V – 253–V – 256.
- [Marcuzzo et al., 2008] Marcuzzo, M., Quelhas, P., Campilho, A., Mendonça, A. M., and Campilho, A. C. (2008). A hybrid approach for arabidopsis root cell image segmentation. In Campilho, A. C. and Kamel, M. S., editors, *ICIAR*, volume 5112 of *Lecture Notes in Computer Science*, pages 739–749. Springer.
- [Medioni et al., 2000] Medioni, G., Lee, M.-S., and Tang, C.-K. (2000). *A computational framework for segmentation and grouping*. Elsevier, Amsterdam, Lausanne, New York.
- [Meijering et al., 2009] Meijering, E., Dzyubachyk, O., Smal, I., and van Cappellen, W. A. (2009). Tracking in cell and developmental biology. *Seminars in Cell and Developmental Biology*, 20(8):894 – 902. Imaging in Cell and Developmental Biology, Planar Cell Polarity.
- [Min and Medioni, 2006] Min, C. and Medioni, G. G. (2006). Tensor voting accelerated by graphics processing units (gpu). In *ICPR (3)*, pages 1103–1106. IEEE Computer Society.
- [Mosaliganti et al., 2010] Mosaliganti, K., Janoos, F., Gelas, A., Noche, R., Obholzer, N., Marchiraju, R., and Megason, S. (2010). Anisotropic plate diffusion filtering for detection of cell membranes in 3d microscopy images. In *Proceedings of the 2010 IEEE international conference on Biomedical imaging: from nano to Macro*, ISBI’10, pages 588–591, Piscataway, NJ, USA. IEEE Press.
- [Mosaliganti et al., 2012] Mosaliganti, K. R., Noche, R. R., Xiong, F., Swinburne, I. A., and Megason, S. G. (2012). Acme: Automated cell morphology extractor for comprehensive reconstruction of cell membranes. *PLoS Comput Biol*, 8(12):e1002780.
- [Munro et al., 2006] Munro, E., Robin, F., and Lemaire, P. (2006). Cellular morphogenesis in ascidians: how to shape a simple tadpole. *Curr Opin Genet Dev*, 16(4):399–405.
- [Risser et al., 2008] Risser, L., Plouraboue, F., and Descombes, X. (2008). Gap filling of 3-d microvascular networks by tensor voting. *Medical Imaging, IEEE Transactions on*, 27(5):674–687.
- [Thiel and Montanvert, 1992] Thiel, E. and Montanvert, A. (1992). Chamfer masks : discrete distance functions, geometrical properties and optimization. In *11th ICPR*, volume 3, pages 244–247, The Hague, The Netherlands.
- [Tong et al., 2004] Tong, W.-S., Tang, C.-K., Mordohai, P., and Medioni, G. (2004). First order augmentation to tensor voting for boundary inference and multiscale analysis in 3d. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):594–611.

- [Traas and Hamant, 2009] Traas, J. and Hamant, O. (2009). From genes to shape: understanding the control of morphogenesis at the shoot meristem in higher plants using systems biology. *C R Biol*, 332(11):974–85.
- [Truong and Supatto, 2011] Truong, T. V. and Supatto, W. (2011). Toward high-content/high-throughput imaging and analysis of embryonic morphogenesis. *Genesis*, 49(7):555–69.
- [Vincent and Soille, 1991] Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6):583–598.
- [Williams and Jacobs, 1995] Williams, L. and Jacobs, D. (1995). Stochastic completion fields: a neural model of illusory contour shape and salience. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 408–415.
- [Yen and Finkel, 1998] Yen, S.-C. and Finkel, L. H. (1998). Extraction of perceptually salient contours by striate cortical networks. *Vision Research*, 38(5):719 – 741.

Annexe A

Résultats complets des détections

Image (#cellules)	Réponse	Tensor Voting	σ	h_{\min}	TD	OD	UD	Temps TV (s)	
Im1 (32)	Michelin	BINAIRE	2	15	29	7	0	2015	
				30	28	7	0		
			3	15	26	7	0	1885	
				30	26	7	0		
			4	15	28	4	0	1881	
				30	28	4	0		
		5	15	28	7	0	1943		
			30	30	2	0			
		6	15	29	8	0	2031		
			30	29	4	0			
		EXTREMA	2	10	26	7	0	20022	
				15	26	7	0		
	3		10	27	7	0	11345		
			15	27	6	0			
	4		10	28	4	0	8253		
			15	27	5	0			
	5		10	27	4	1	6181		
			15	28	4	0			
	6		10	27	4	1	5932		
			15	28	4	0			
	Mosaliganti		BINAIRE	2	15	26	9	0	2310
					30	27	9	0	
		3		15	28	6	0	2178	
				30	27	8	0		
		4		15	28	4	0	2220	
				30	29	3	0		
		5	15	28	5	0	2103		
			30	30	3	0			
		6	15	29	9	0	2059		
			30	28	4	0			
		EXTREMA	2	10	17	25	0	22384	
				15	19	20	0		
			3	10	26	8	0	13552	
				15	27	5	0		
			4	10	25	7	0	9268	
				15	24	8	0		
		5	10	24	8	2	6839		
			15	27	5	0			
	6	10	28	4	1	5159			
		15	28	5	0				
	REPONSE	2	10	19	14	0	73722		
			15	26	8	0			
3		10	28	4	0	65209			
		15	26	7	0				
4		10	23	9	0	61365			
		15	24	8	0				
5	10	26	6	1	56516				
	15	26	5	1					
6	10	25	8	3	42871				
	15	27	4	3					
REFERENCE					31	9	0		

Image (#cellules)	Réponse	Tensor Voting	σ	h_{\min}	TD	OD	UD	Temps TV (s)	
Im2 (162)	Michelin	BINAIRE	2	15	111	65	2	2893	
				30	114	60	4		
			3	15	136	29	2	2562	
				30	137	22	5		
			4	15	149	15	3	2422	
				30	145	12	7		
		5	15	144	8	10	2372		
			30	139	5	14			
		6	15	144	7	10	2722		
			30	129	4	22			
		EXTREMA	2	10	106	70	6	20681	
				15	130	32	4		
	3		10	124	46	4	13794		
			15	135	24	4			
	4		10	132	31	6	8899		
			15	139	15	10			
	5		10	144	11	8	7009		
			15	137	8	15			
	6		10	134	9	16	6694		
			15	129	7	21			
	Mosaliganti		BINAIRE	2	15	92	94	7	3297
					30	96	104	3	
		3		15	109	77	5	3157	
				30	114	66	3		
		4		15	139	29	4	2965	
				30	141	20	4		
		5	15	139	16	10	2684		
			30	138	12	11			
		6	15	140	8	13	2578		
			30	140	4	14			
		EXTREMA	2	10	79	148	11	22918	
				15	52	251	15		
			3	10	84	124	7	15416	
				15	96	109	4		
			4	10	125	44	3	10566	
				15	134	30	4		
		5	10	138	14	11	7574		
			15	139	6	14			
	6	10	132	3	20	5997			
		15	129	3	22				
	REPONSE	2	10	83	100	9	78154		
			15	55	196	9			
3		10	97	80	9	70101			
		15	102	68	6				
4		10	126	28	10	64681			
		15	139	18	8				
5	10	122	15	21	58473				
	15	132	4	19					
6	10	124	8	23	49663				
	15	121	2	29					
REFERENCE					128	40	5		

TAB. A.1: #cellules : nombre de cellules de la VT. {Michelin/Mosaliganti} : type de fonction de réponse choisi. {REPONSE/EXTREMA/BINAIRE} : procédé utilisé pour le Tensor Voting (figure 2.2). σ : échelle utilisée (en pixels) pour le calcul des dérivées d'intensité. h_{\min} : unique paramètre variable de la phase de détection (voir section 5.3.1). TD/OD/UD : bonnes/sur-/sous-détections. Temps TV : temps mis pour effectuer l'étape de Tensor Voting (en secondes).

Annexe B

Tensor Voting

B.1 Solution analytique : *Closed-Form solution to Tensor Voting*

Il n'existe pas une unique manière de réaliser le vote de tenseurs. Dans la littérature, on trouve différentes variantes du tensor voting. Certaines variantes proposent un vote qui prend en considération des informations sur la courbure locale [Fischer et al., 2007], d'autres ont pour but d'obtenir une expression analytique du vote dans le cas général [Jia et al., 2012]. Nous nous sommes intéressés au problème d'exprimer le vote d'un tenseur de façon analytique. L'intérêt de connaître cette expression est que nous pouvons de cette façon mettre au point une extraction basée modèle des structures planes et linéiques.

B.1.1 Votes Stick

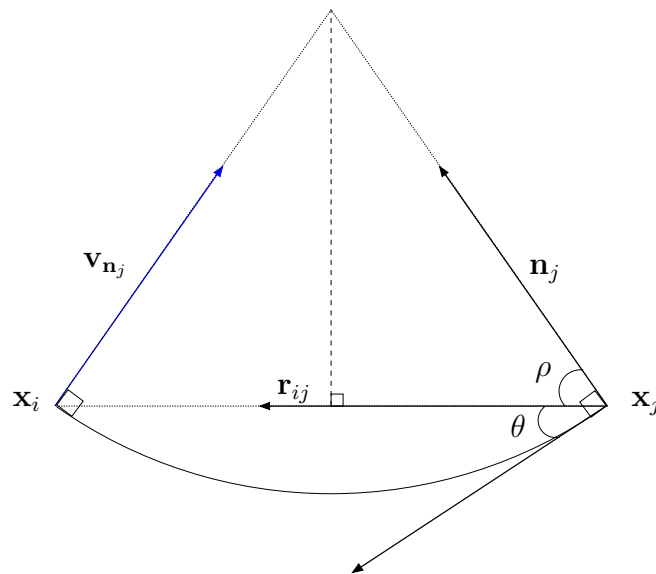


FIG. B.1: Vote reçu.

Dans [Jia et al., 2012], les auteurs proposent une redéfinition du champ de tenseurs Stick qui permet le calcul d'une formule analytique du vote reçu en un point venant d'un tenseur quelconque. Le vote $\mathbf{V}_{ij}(\mathbf{N}_j)$ d'un tenseur stick $\mathbf{N}_j = \mathbf{n}_j \mathbf{n}_j^T$ du voxel \mathbf{x}_j reçu en le voxel \mathbf{x}_i est défini par :

$$\mathbf{V}_{S,ij}(\mathbf{N}_j) = \mathbf{V}_S(\mathbf{x}_i, \mathbf{x}_j, \mathbf{n}_j) = \eta_{ij\mathbf{n}_j} \mathbf{v}_{\mathbf{n}_j} \mathbf{v}_{\mathbf{n}_j}^T \quad (\text{B.1})$$

où la nouvelle fonction de décroissance est :

$$\eta_{ij\mathbf{n}_j} = \eta(\mathbf{x}_i, \mathbf{x}_j, \mathbf{n}_j) = c_{ij}(1 - (\mathbf{r}_{ij}^T \mathbf{n}_j)^2) \quad (\text{B.2})$$

Ici, \mathbf{r}_{ij} est le vecteur unitaire pointant de \mathbf{x}_j vers \mathbf{x}_i , et c_{ij} est une fonction exponentielle paramétrée par l'échelle de vote σ :

$$c_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) \quad (\text{B.3})$$

Le vecteur $\mathbf{v}_{\mathbf{n}_j}$ est le même que celui proposé dans l'approche d'origine (figure B.1) :

$$\mathbf{v}_{\mathbf{n}_j} = \underbrace{(\mathbf{I} - 2\mathbf{r}_{ij}\mathbf{r}_{ij}^T)}_{\mathbf{R}_{ij}} \mathbf{n}_j = \mathbf{R}_{ij} \mathbf{n}_j \quad (\text{B.4})$$

B.1.2 Votes Ball

Nous rappelons l'expression du vote \mathbf{V}_B d'un tenseur Ball défini par 4.4 :

$$\mathbf{V}_{B,ij} = \mathbf{V}_B(\mathbf{x}_i, \mathbf{x}_j) = \int_{\phi \in \mathcal{D}_B} \mathbf{V}_{S,ij}(\mathbf{N}_\phi) d\phi \quad (\text{B.5})$$

où \mathcal{D}_B constitue le domaine de définition de la variable ϕ tel que $\{\mathbf{N}_\phi\}_{\phi \in \mathcal{D}_B} = \{\mathbf{N} = \mathbf{n}\mathbf{n}^T\}_{\|\mathbf{n}\|=1}$.

À partir des équation B.4, B.2 et B.5, on obtient :

$$\begin{aligned} \mathbf{V}_{B,ij} &= \int_{\phi \in \mathcal{D}_B} \eta_{ij\mathbf{n}_\phi} \mathbf{v}_\phi \mathbf{v}_\phi^T d\phi \\ &= c_{ij} \mathbf{R}_{ij} \left(\int_{\phi \in \mathcal{D}_B} \left(1 - (\mathbf{r}_{ij}^T \mathbf{n}_\phi)^2\right) \mathbf{n}_\phi \mathbf{n}_\phi^T d\phi \right) \mathbf{R}_{ij}^T \\ &= c_{ij} \mathbf{R}_{ij} \left(\int_{\phi \in \mathcal{D}_B} \left(\mathbf{N}_\phi - \mathbf{n}_\phi \left(\mathbf{n}_\phi^T \mathbf{r}_{ij} \mathbf{r}_{ij}^T \mathbf{n}_\phi\right) \mathbf{n}_\phi^T\right) d\phi \right) \mathbf{R}_{ij}^T \\ &= c_{ij} \mathbf{R}_{ij} \left(\underbrace{\int_{\phi \in \mathcal{D}_B} \mathbf{N}_\phi d\phi}_{\mathbf{M} \propto \mathbf{T}_B} - \underbrace{\int_{\phi \in \mathcal{D}_B} \mathbf{N}_\phi \mathbf{r}_{ij} \mathbf{r}_{ij}^T \mathbf{N}_\phi d\phi}_{\mathbf{A}(\mathbf{r}_{ij})} \right) \mathbf{R}_{ij}^T \end{aligned} \quad (\text{B.6})$$

Cette nouvelle forme présente deux termes calculables séparément : \mathbf{M} et $\mathbf{A}(\mathbf{f}_{ij})$. On remarque que \mathbf{M} est la somme de tenseurs stick selon toutes les orientations possibles. Par symétrie, cette somme est nécessairement égale à un tenseur ball et est donc proportionnelle à $\mathbf{T}_B = \mathbf{I}$ où \mathbf{I} est la matrice identité. Nous distinguons le cas 2D et le cas 3D pour le calcul de ces deux termes.

B.1.2.1 Cas 2D

Dans le plan, ϕ correspond à l'angle tel que $\mathbf{n}_\phi = (\cos \phi, \sin \phi)^T$. On peut choisir comme domaine $\mathcal{D}_B = [0, \pi[$. De plus, nous avons :

$$\mathbf{N}_\phi = \mathbf{n}_\phi \mathbf{n}_\phi^T = \begin{pmatrix} \cos^2 \phi & \cos \phi \sin \phi \\ \cos \phi \sin \phi & \sin^2 \phi \end{pmatrix} \quad (\text{B.7})$$

Nous calculons \mathbf{M} en réalisant l'intégrale sur chacune des composantes de \mathbf{N}_ϕ :

$$\begin{aligned} \mathbf{M}_{11} &= \int_{\phi \in [0, \pi[} \cos^2 \phi d\phi \\ &= \frac{1}{2} \int_{\phi \in [0, \pi[} (1 + \cos(2\phi)) d\phi \\ &= \frac{1}{2} \left([\phi]_{\phi=0}^{\pi} + \underbrace{\int_{\phi \in [0, \pi[} \cos(2\phi) d\phi}_0 \right) \\ &= \frac{\pi}{2} \end{aligned}$$

$$\begin{aligned} \mathbf{M}_{22} &= \int_{\phi \in [0, \pi[} \sin^2 \phi d\phi \\ &= \frac{1}{2} \int_{\phi \in [0, \pi[} (1 - \cos(2\phi)) d\phi \\ &= \frac{\pi}{2} \end{aligned}$$

$$\begin{aligned} \mathbf{M}_{12} = \mathbf{M}_{21} &= \int_{\phi \in [0, \pi[} \cos \phi \sin \phi d\phi \\ &= \frac{1}{2} \int_{\phi \in [0, \pi[} \sin(2\phi) d\phi \\ &= 0 \end{aligned}$$

et on obtient :

$$\mathbf{M} = \frac{\pi}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (\text{B.8})$$

Pour le calcul de $\mathbf{A}(\mathbf{r}_{ij})$, nous nous plaçons dans le cas particulier où $\mathbf{r}_{ij} = \mathbf{r} = (1, 0)^T$. On a alors :

$$\mathbf{r}\mathbf{r}^T = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

donc :

$$\mathbf{N}_\phi \mathbf{r}\mathbf{r}^T \mathbf{N}_\phi = \begin{pmatrix} \mathbf{N}_{\phi 11}^2 & \mathbf{N}_{\phi 11} \mathbf{N}_{\phi 12} \\ \mathbf{N}_{\phi 11} \mathbf{N}_{\phi 12} & \mathbf{N}_{\phi 12}^2 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 + 2 \cos(2\phi) + \frac{1}{2}(1 + \cos(4\phi)) & \sin(2\phi) + \frac{1}{2} \sin(4\phi) \\ \sin(2\phi) + \frac{1}{2} \sin(4\phi) & \frac{1}{2}(1 - \cos(4\phi)) \end{pmatrix}$$

L'intégration composante par composante donne :

$$\begin{aligned} \mathbf{A}_{11}(\mathbf{r}) &= \frac{1}{4} \int_{\phi \in [0, \pi[} \left(1 + 2 \cos(2\phi) + \frac{1}{2}(1 + \cos(4\phi)) \right) d\phi \\ &= \frac{3\pi}{8} \end{aligned}$$

$$\begin{aligned} \mathbf{A}_{22}(\mathbf{r}) &= \frac{1}{4} \int_{\phi \in [0, \pi[} \left(\frac{1}{2}(1 - \cos(4\phi)) \right) d\phi \\ &= \frac{\pi}{8} \end{aligned}$$

$$\begin{aligned} \mathbf{A}_{12}(\mathbf{r}) = \mathbf{A}_{21}(\mathbf{r}) &= \frac{1}{4} \int_{\phi \in [0, \pi[} \left(\sin(2\phi) + \frac{1}{2} \sin(4\phi) \right) d\phi \\ &= 0 \end{aligned}$$

et on obtient :

$$\mathbf{A}(\mathbf{r}) = \frac{\pi}{8} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \quad (\text{B.9})$$

Les équations B.7, B.8 et B.9 nous donnent :

$$\begin{aligned} \mathbf{V}_B(\mathbf{r}) &= c_{ij} \frac{\pi}{8} \mathbf{R} \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix} \mathbf{R}^T \\ &= c_{ij} \frac{\pi}{8} \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix} \\ &= c_{ij} \frac{\pi}{8} (\mathbf{r}\mathbf{r}^T + 3\mathbf{t}\mathbf{t}^T) \end{aligned} \quad (\text{B.10})$$

où $\mathbf{t} = (0, 1)^T$ est un vecteur unitaire normal à \mathbf{r} .

Par rotation du problème, on exprime ainsi le vote d'un tenseur ball pour \mathbf{r}_{ij} quelconque :

$$\begin{aligned}\mathbf{V}_{B,ij} &= c_{ij} \frac{\pi}{8} \left(\mathbf{r}_{ij} \mathbf{r}_{ij}^T + 3 \mathbf{t}_{ij} \mathbf{t}_{ij}^T \right) \\ &= c_{ij} \frac{\pi}{8} \left(\mathbf{I} + 2 \mathbf{t}_{ij} \mathbf{t}_{ij}^T \right)\end{aligned}\tag{B.11}$$

avec le vecteur unitaire $\mathbf{t}_{ij} \perp \mathbf{r}_{ij}$.

B.1.2.2 Cas 3D

Dans l'espace, $\phi = (\phi_1, \phi_2)$ désigne le doublet des coordonnées sphériques du vecteur unitaire $\mathbf{n}_\phi = (\cos(\phi_1) \cos(\phi_2), \sin(\phi_1) \cos(\phi_2), \sin(\phi_2))^T$. On a $\mathbf{N}_\phi = \mathbf{n}_\phi \mathbf{n}_\phi^T$. L'ensemble de définition de ϕ est $\mathcal{D}_B = \{(\phi_1, \phi_2) \mid \phi_1 \in [0, \pi[\wedge \phi_2 \in [0, \pi]\}$.

Le calcul de l'expression B.6 donne :

$$\mathbf{V}_{B,ij} = \mathbf{V}_B(\mathbf{x}_i, \mathbf{x}_j) = \int_{\phi \in \mathcal{D}_B} \mathbf{V}_{S,ij}(\mathbf{N}_\phi) d\phi\tag{B.12}$$

$$= c_{ij} \frac{4\pi}{15} (\mathbf{I} + \mathbf{t}_1 \mathbf{t}_1^T + \mathbf{t}_2 \mathbf{t}_2^T)\tag{B.13}$$

où $\{\mathbf{t}_1, \mathbf{t}_2\}$ sont deux vecteurs unitaires orthogonaux à \mathbf{r}_{ij} .

B.1.3 Théorème de la solution analytique pour le Tensor Voting

L'article [Jia et al., 2012] formule le théorème suivant :

Théorème 1. Le vote induit en \mathbf{x}_i par un tenseur symétrique du second ordre \mathbf{K}_j localisé en \mathbf{x}_j est donné par :

$$\mathbf{S}_{ij} = c_{ij} \mathbf{R}_{ij} \mathbf{K}_j \mathbf{R}'_{ij}$$

où $\mathbf{r}_{ij} = (\mathbf{x}_i - \mathbf{x}_j) / \|\mathbf{x}_i - \mathbf{x}_j\|$, $c_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right)$, $\mathbf{R}_{ij} = \mathbf{I} - 2\mathbf{r}_{ij} \mathbf{r}_{ij}^T$ et $\mathbf{R}'_{ij} = (\mathbf{I} - \frac{1}{2}\mathbf{r}_{ij} \mathbf{r}_{ij}^T) \mathbf{R}_{ij}$.

Cette formulation est cependant incorrecte puisqu'on remarque qu'avec un tenseur votant de type plate, l'application du théorème 1 ne correspond pas à la valeur théorique de l'intégration de 4.5.

On remarque en effet la présence d'un élément de preuve discutable dans l'article, ce qui explique cette observation.

Nous avons cherché à recalculer la solution analytique du problème dans le cas général, mais nous n'avons pas abouti à une formulation satisfaisante et avons abandonné l'exploitation de la fonction de décroissance proposée dans [Jia et al., 2012].

Annexe C

Article

Le manuscrit ci-joint est en cours de rédaction et devrait être soumis au [IEEE *International Symposium on Biomedical Imaging*](#) (ISBI). Il s'agit d'un congrès scientifique annuel qui se tiendra à Pékin entre le 28 avril et le 2 mai 2014 et qui traite des avancées en traitement du signal appliqué à la biologie et à la médecine.

Embryo Cell Membranes Reconstruction by Tensor Voting for Cells Detection

Gaël Michelin*, Léo Guignard†, and Grégoire Malandain*

*INRIA, 06900 Sophia Antipolis, France

†Univ. Montpellier 1, Univ. Montpellier 2, CNRS, CRBM, UMR 5237, 34293 Montpellier, France

Abstract—Imaged-based studies of developing organs or embryos produce a huge quantity of data. To handle such high-throughput experimental protocol, automated computer-assisted methods are highly desirable. We propose in this article an original method to extract cell membranes. Unavoidable segmentation gaps are corrected by perceptual grouping (tensor voting). To reduce the computational cost of this last step, we propose different methodologies to reduce the number of voters. Assessment on real data allows us to deduce the most efficient approach.

I. INTRODUCTION

The organism formation during animal or plant development is a major question in developmental biology. One challenge is to describe shape evolution as a geometrical output of gene activity [8], [10]. Due to recent progress in microscopy, time-lapse 3D images of living organisms can be recorded for several hours at a high frequency that permits to monitor cell deformations and divisions [3]. This results in huge data sets that require highly robust and efficient computational image analysis tools for segmenting the cells, registering the images if necessary, and extract the cell lineages [11].

The present work focuses on the cell segmentation step. The fluorescence of cell membranes or walls makes watershed [12] a method of choice [1], [7]. However, imaging noise and artifacts as well as fluorescence defects may cause segmentation errors. While deformable models may help to produce well-shaped cells, they may be computationally expensive if a large number of cells have to be segmented. Tensor voting is a means to correct for defaults in segmentation [6]. [7] demonstrates that, combined to a membrane enhancement filter, it overperformed the watershed algorithm applied on original data, but requires an important computational effort.

As contributions, we propose an alternative membrane enhancement filter. In addition, we propose different strategies to select membrane center points to reduce the computational effort for tensor voting. Several segmentation approaches can be built by combination of the filters and selection strategies. Quantitative assessment on real data allows us to select the most efficient ones.

[to be finished]

Contact: {gael.michelin,gregoire.malandain}@inria.fr

Fig. 1. From left to right: 2D cross-section of a 3D image, computed filter response, extrema, and thresholded extrema.

II. MEMBRANE ENHANCEMENT

A. Membrane enhancement filters

In 3D images, line-like or plane-like structures can be characterized by the study of the Hessian matrix [5]. Assuming that the structures are bright over a dark background, and that the eigenvalues λ_i of the Hessian matrix are ordered by their magnitude ($|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$), a line-like structure is characterized by large (and negative) λ_3 and λ_2 and a small $|\lambda_1|$ (the associated eigenvector \mathbf{v}_1 gives the line direction), while a plane-like structure is characterized by a large (and negative) λ_3 (the associated eigenvector \mathbf{v}_3 gives the orthogonal direction to the plane) and small λ_2 and $|\lambda_1|$. This properties have been exploited in medical imaging to detect vessels, either by building an ad-hoc response function based on the eigenvalues [2] or by integrating an edge detection response around a vessel center candidate [4].

Mosaliganti and col. have adapted Frangi's formulation to design a membrane enhancement filter [7], and we propose here to adapt Krissian's approach to membranes. Although straightforward, this has not been done yet to the best of our knowledge. The response function at a point P of an image I is calculated by integrating an edge response at a distance r to a membrane center candidate:

$$R_K(P) = \begin{cases} 0 & \text{if } \lambda_3 \geq 0 \\ \frac{1}{2} (\nabla I(P - r\mathbf{v}_3) \cdot \mathbf{v}_3 - \nabla I(P + r\mathbf{v}_3) \cdot \mathbf{v}_3) & \text{else} \end{cases} \quad (1)$$

B. Extrema extraction

The two above filters are designed so that the response is maximal at the membrane center. Thus suppressing the non-maxima will help to keep only pertinent information while suppressing the spurious one. This is done by extracting the directional (with respect to \mathbf{v}_3) extrema of the response, i.e.

$$E_X(P) = \begin{cases} 0 & \text{if } R_X(P) \leq R_X(P \pm \mathbf{v}_3) \\ R_X(P) & \text{else} \end{cases} \quad (2)$$

where X is respectively K for Krissian like filter and F for Frangi like one.

C. Extrema thresholding

[to be done]

III. TENSOR VOTING

The above filters are designed to enhance plane-like structures. However, they will fail to enhance them at junctions or when the original image signal is too weak. Perceptual grouping, by the means of tensor voting, may address these two points.

A. Structural representation

The principle of tensor voting is to represent the local structure (at each point P) by a tensor, $\mathbf{T}(P)$, i.e. 3×3 real positive, symmetric matrix. Its decomposition in eigenvalues $\kappa_3 \geq \kappa_2 \geq \kappa_1 \geq 0$ and associated eigenvectors \mathbf{e}_i allows to rewrite it as a linear combination of three generic tensors (figure 2):

$$\mathbf{T} = \sum \kappa_i \mathbf{e}_i \mathbf{e}_i^t = \kappa_1 \underbrace{(\mathbf{e}_1 \mathbf{e}_1^t + \mathbf{e}_2 \mathbf{e}_2^t + \mathbf{e}_3 \mathbf{e}_3^t)}_{\mathbf{T}_B} + (\kappa_2 - \kappa_1) \underbrace{(\mathbf{e}_2 \mathbf{e}_2^t + \mathbf{e}_3 \mathbf{e}_3^t)}_{\mathbf{T}_P} + (\kappa_3 - \kappa_2) \underbrace{\mathbf{e}_3 \mathbf{e}_3^t}_{\mathbf{T}_S} \quad (3)$$

$\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ defines a basis where

$$\mathbf{T}_S = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{T}_P = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{T}_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

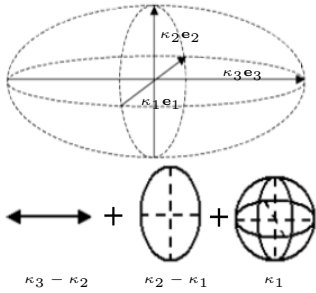


Fig. 2. Tensor decomposition into Stick, Plate and Ball elements.

These generic tensors are respectively named *stick*, *plate* and *ball* unit tensors.

- A stick tensor expresses uncertainty of data orientation in the two directions \mathbf{e}_1 and \mathbf{e}_2 , it corresponds to a planar structure.
- A plate tensor expresses uncertainty in the direction \mathbf{e}_1 and corresponds to a line structure.
- A ball tensor does not express any orientation preference, it is the case in junction points.

B. Tensor voting

Tensor voting consists in building a tensor map from the votes of points P or *tokens*, that can be points without structural information ($P \in \mathcal{B}$), or points from lines ($P \in \mathcal{P}$) or planes ($P \in \mathcal{S}$), i.e. associated to some privileged directions. For each type de structure, a tensor voting field is built (see [6] for details) that aims at expanding the structures

along their preferential directions ((figure 3). The result of tensor voting is then a tensor image \mathbf{J}

$$\mathbf{J}(M) = \sum_{\mathcal{X} \in \{\mathcal{B}, \mathcal{P}, \mathcal{S}\}} \sum_{P \in \mathcal{X}} \alpha(P) \mathbf{V}_{\mathcal{X}}(\mathbf{PM}, \mathbf{e}_i(P)) \quad (4)$$

where $\mathbf{V}_{\mathcal{X}}(\mathbf{PM}, \mathbf{e}_i(P))$ denotes the vote of token P of type $\mathcal{X} \in \{\mathcal{B}, \mathcal{P}, \mathcal{S}\}$ at point M , weighted by $\alpha(P)$.

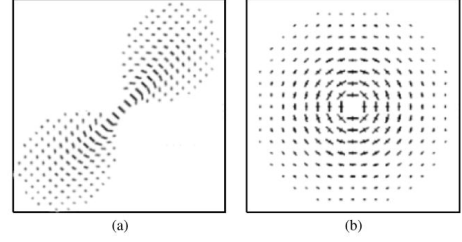


Fig. 3. Voting fields: (a) a cut of the voting field for stick component; (b) a cut of the voting field for plate component (direction \mathbf{e}_1 is normal to the page).

For the present application, we are only interested in plane-like structures (the membranes), hence for an image $I \in \{R_X, E_X, B_X\}$, we only consider stick tensor votes with

$$\mathbf{J}(M) = \sum_{P/I(P)>0} I(P) \mathbf{V}_{\mathcal{S}}(\mathbf{PM}, \mathbf{e}_i(P)) \quad (5)$$

Since we only consider point with non-null intensity as tokens, the computation cost of tensor voting is obviously ordered from $I = R_X$ to B_X , since it linearly depends on the number of non-null points P . Please note that votes are weighted with the filter response for $I \in \{R_X, E_X\}$, the thresholded extrema value being either 0 or 1.

IV. CELL SEGMENTATION

From the tensor map \mathbf{J} , a surfaceness map S is computed with $S(M) = \kappa_3(\mathbf{J}(M)) - \kappa_2(\mathbf{J}(M))$ (see eq. (3)) that is subsequently used to segment the cells. For that purpose, the watershed method is used. However, this approach is known to be prone to oversegmentations and since some gaps may still exist at junction, we design a dedicated seed extraction method, and the labeled seeds will be used as sources for the watershed instead of the minima of the S image.

First, h -minima are extracted from S [9]. Since membrane gaps form bridges between two adjacent cells, we recognize them by computing a distance map inside the extracted h -minima and then by extracting the h -maxima of the distance map. These labeled h -maxima are used as sources for a watershed segmentation with a regularized (i.e. convolved by a Gaussian) version of S , i.e. $S * G_\sigma$.

V. EXPERIMENTS

A. Computational issues

The described method rely on a number of parameters. First, Frangi's and Krissian's filters require the computation of the image derivatives, which is achieved by convolving the image with the Gaussian derivatives. Although these filters can be embedded into a multi-scale approach to handle

difference of sizes of the structures to segment, we choose to use only one scale denoted by σ_R since the membranes have a homogeneous thickness. We test $\sigma_R \in \{3, 4, 5\}$. The threshold ... The size of the tensor voting field V_X is governed by another standard deviation σ_T that have to be chosen accordingly to the size of the gaps to be filled ... The threshold ... The height for h -minima and h -maxima ...

B. Data

We imaged a simple chordate organism, *Phallusia mammillata* embryos, to assess quality of technics described in this paper. Embryo's membranes are marked by a lipophilic dye (FM4-64 which respond at $\sim 700nm$ from an excitation at $\sim 500nm$). We started the imaging at the end of the 32 cells stage of the embryo and stops it during its 172 cells stage. The embryo was imaged every one minute from 4 different angles for 2 hours with a light-sheet microscope, a MuViSPIM [c]. The 4 angles were then affinely registered and fused to mitigate image acquisition defects due to, for example, light absorption and/or microscope anisotropy [c].

One special characteristic of this setup is that the die is slowly internalized inside the cytoplasm. It has, as impact, a degradation through time of the signal over noise ratio. This defects will allow us to have different image quality and therefor two different experiment conditions. We choose for our tests the 1st and the 91th (after 1 and an half hour of imaging) from the sequence.

C. Evaluation methodology

The purpose of this work is to design an efficient method for cell segmentation. We want to assess the use as tokens for tensor voting of either the filter response R_X , the extracted extrema E_X , or the thresholded extracted extrema B_X for two filters, namely the Frangi-like one ($X = F$) and the Krissian-like one ($X = K$). This yields 6 token images to be compared, multiplied by the number of tested parameter sets. Notet that using R_F as token image is similar to the ACME method [7], thus we have a direct comparison with this approach.

We have at hand a ground truth (GT) segmentation of two 3D images, obtained by manually correcting the segmentation (denoted REFERENCE) computed with [1]. Briefly, this is a watershed on the (regularized) original data with an ad-hoc seed/source detection. Since we use a different image for the watershed, there will be unavoidable differences at the cell borders between the ground truth segmentation and the ones we obtain. In addition, we are more interested in evaluating the topological errors (i.e. the number of over-segmentations and of missed cells) than the precision of the border of the segmented cells. For these reasons, we design three measures to quantify these errors based on the comparison of the detected seeds for watershed (see section IV) against the segmented cell of the ground truth.

- True detections TD characterize a one-to-one mapping between a GT cell and a seed: the cell contains only one seed, and this seed does not intersect any other cell.

- An over-detection OD occurs if a cell contains more than one seed, and there will be as many over-segmentations of this cell as there are supplementary seeds (a cell containing 3 seeds counts for 2 OD)
- An under-detection UD can occur by two different ways, firstly if one cell does not contain any seed, and secondly if a seed intersects more than one cell.

D. Results

VI. CONCLUSION

APPENDIX

ACKNOWLEDGMENT

REFERENCES

- [1] Romain Fernandez, Pradeep Das, Vincent Mirabet, Eric Moscardi, Jan Traas, Jean-Luc Verdeil, Grégoire Malandain, and Christophe Godin. Imaging plant growth in 4-d: robust tissue reconstruction and lineaging at cell resolution. *Nature Methods*, 7(7):547–553, 2010.
- [2] Alejandro F. Frangi, Wiro J. Niessen, Koen L. Vincken, and Max A. Viergever. Multiscale vessel enhancement filtering. In William M. Wells, Alan Colchester, and Scott Delp, editors, *Medical image computing and computer-assisted intervention (MICCAI 1998)*, volume 1496 of LNCS, pages 130–137, Cambridge, UK, 1998. Springer.
- [3] Philipp J Keller. Imaging morphogenesis: technological advances and biological insights. *Science*, 340(6137):1234168, June 2013.
- [4] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousslet. Model-based detection of tubular structures in 3d images. *Computer Vision and Image Understanding*, 80(2):130–171, 2000.
- [5] C. Lorenz, I.-C. Carlsen, T.M. Buzug, C. Fassnacht, and J. Weese. Multi-scale line segmentation with automatic estimation of width, contrast and tangential direction in 2D and 3D medical images. In J. Troccaz, E. Grimson, and R. Msges, editors, *CVRMed-MRCAS'97*, number 1205 in LNCS, pages 233–242, Grenoble, France, March 1997. Springer.
- [6] Gerard Medioni, Mi-Suen Lee, and Chi-Keung Tang. *Computational Framework for Segmentation and Grouping*. Elsevier Science Inc., New York, NY, USA, 2000.
- [7] Kishore R Mosaliganti, Ramil R Noche, Fengzhu Xiong, Ian A Swinburne, and Sean G Megason. Acme: automated cell morphology extractor for comprehensive reconstruction of cell membranes. *PLoS Comput Biol*, 8(12), 2012.
- [8] Edwin Munro, Francois Robin, and Patrick Lemaire. Cellular morphogenesis in ascidians: how to shape a simple tadpole. *Curr Opin Genet Dev*, 16(4):399–405, August 2006.
- [9] Pierre Soille. *Morphological image analysis: principles and applications*. Springer, 1999.
- [10] Jan Traas and Olivier Hamant. From genes to shape: understanding the control of morphogenesis at the shoot meristem in higher plants using systems biology. *C R Biol*, 332(11):974–85, November 2009.
- [11] Thai V Truong and Willy Supatto. Toward high-content/high-throughput imaging and analysis of embryonic morphogenesis. *Genesis*, 49(7):555–69, July 2011.
- [12] Luc Vincent and Pierre Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.