



HAL
open science

Analyse rapide d'images 3D de matériaux hétérogènes : identification de la structure des milieux et application à leur caractérisation multi-échelle

Amaury Walbron

► **To cite this version:**

Amaury Walbron. Analyse rapide d'images 3D de matériaux hétérogènes : identification de la structure des milieux et application à leur caractérisation multi-échelle. Mathématiques générales [math.GM]. Université d'Orléans, 2016. Français. NNT : 2016ORLE2015 . tel-01431187

HAL Id: tel-01431187

<https://theses.hal.science/tel-01431187v1>

Submitted on 10 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**ÉCOLE DOCTORALE MATHÉMATIQUES, INFORMATIQUE, PHYSIQUE
THÉORIQUE ET INGÉNIERIE DES SYSTÈMES**

Département Matériaux, CEA Le Ripault / MAPMO, Université d'Orléans

THÈSE présentée par :
Amaury WALBRON

soutenue le : **01 avril 2016**

pour obtenir le grade de : **Docteur de l'université d'Orléans**

Discipline/ Spécialité : **Mathématiques**

**Analyse rapide d'images 3D de matériaux
hétérogènes : identification de la structure
des milieux et application à leur
caractérisation multi-échelle**

THÈSE dirigée par :

Denis ROCHAIS HDR, Expert-Senior, CEA Le Ripault
Romain ABRAHAM Professeur, MAPMO, Université d'Orléans

RAPPORTEURS :

Hermine BIERMÉ Professeure, LMA, Université de Poitiers
Nicolas PASSAT Professeur, CRESTIC, Université de Reims Champagne-Ardenne

JURY:

Christian GERMAIN Professeur, UMR IMS, Université de Bordeaux, Président du jury
Cédric DESCAMPS Docteur, HERAKLES
Jérôme VICENTE Maître de Conférences, IUSTI, Polytech Marseille

Remerciements

Je tiens à remercier l'ensemble des personnes ayant participé, de près ou de loin, directement ou plus indirectement à l'établissement et au bon déroulement de cette thèse.

Tout d'abord mes directeurs de thèse, Denis Rochais et Romain Abraham, auxquels je joins Maïtine Bergounioux. Ils ont, tous ensemble, réussi à encadrer ma thèse de manière vivante. Je leur suis reconnaissant pour l'ensemble du travail qu'ils ont dû effectuer pour moi, pour la marge de manœuvre qu'ils m'ont laissé, tout en apportant avec implication leur rigueur et leurs savoirs scientifiques afin de m'orienter au mieux. Cette dernière et ultime brique (enfin, je l'espère), sur l'édifice de mes études, porte leur empreinte. Leur sympathie, leur attention et leur humour m'ont permis finalement de passer trois années de recherche très agréables.

Je remercie en particulier Denis Rochais pour son accueil au laboratoire du CEA et son grand secours amical dans nombre de domaines, qu'ils soient scientifiques ou non. Ce doctorat fut une nouveauté pour nous deux, que ce soit en tant que directeur de thèse et que doctorant.

Romain Abraham, ensuite, pour ses conseils précieux non seulement en mathématiques, mais aussi dans une multitude d'autres domaines. Son expérience de l'encadrement en particulier aura été crucial pour moi.

Maïtine Bergounioux enfin, par qui cette période de mon parcours a débuté en mars 2012, un an même avant le début de ce doctorat. Elle m'aura permis de découvrir le traitement d'image d'une part, mais aussi, à travers sa connaissance du milieu et son humour, le milieu de la recherche.

À ces remerciements, j'associe bien entendu Sylvain Chupin, avec qui j'ai partagé un bureau vide de toute âme qui vive, comme de nombreux observateurs à l'œil acéré nous l'ont souvent fait remarquer. Outre ce bureau, cible de plaisanteries quotidiennes, j'ai passé en réalité la plus grande partie de ces trois années en sa présence, et, de par le côté symétrique, lui aussi. Il fallait que je loue son mental d'acier pour m'avoir supporté, moi ainsi que mes jeux de mots bien souvent douteux, au jour le jour trois ans de suite. Son aide et son avis sur des sujets divers et variés, qu'il s'agisse de programmation, d'algorithmique, des us et coutumes du CEA m'ont été toujours d'une grande utilité.

Je remercie les rapporteurs de ces travaux de thèse, Hermine Biermé et Nicolas Passat, pour avoir acceptés d'examiner ce document, qui plus est dans des délais que l'on peut raisonnablement qualifier de serrés.

J'associe tout naturellement à ces remerciements les membres du jury, Christian Germain, Cédric Descamps et Jérôme Vicente.

Je remercie aussi l'ensemble des personnes et des institutions ayant permis à cette thèse de se dérouler dans les meilleures conditions au regard des diverses administrations, que ce soit le MAPMO, l'école doctorale, l'université d'Orléans, le CEA, où même dans d'autres entités que j'aurais par mégarde oublié. Je remercie en particulier les secrétariats et leurs membres pour leurs aides précieuses. J'associe à ces remerciements les équipes informatiques pour leur support technique d'une importance capitale.

Pour leur présence quotidienne, je remercie l'ensemble des membres du laboratoire du CEA m'ayant accueilli pendant au plus trois ans, ceux qui sont toujours présents bien sûr, mais aussi ceux qui ont été appelés vers d'autres fonctions, pour utiliser la formule consacrée.

De même je remercie aussi l'ensemble des membres du MAPMO pour leur accueil lors de ces entrevues régulières, que ce soit pour un groupe de travail ou plus simplement une pause café.

Je tiens aussi à remercier ceux ayant contribué à cette thèse, Pierre Debs au MAPMO et François Guillet au CEA.

Un grand merci à Sylvain, Élodie, Pascal et Jonathan pour avoir accepté de transporter, parfois des mois durant, cette personne encombrante que je suis sur vos sièges passagers. Pour cela, je vous suis infiniment reconnaissant.

Je remercie mes parents, Dominique et Laurent ainsi que mon frère Corentin et ma sœur Nausicaa pour leur soutien depuis le début et durant ces trois ans en particulier.

Je remercie finalement Maïlys pour son soutien et sa confiance depuis ces quelques années. Elle m'a aidé à me reprendre quand par moment je ne croyais plus, et sa présence m'a aidé bien plus qu'on ne peut le soupçonner. Merci !

Aucune image n'a été mal-traitée pour cette thèse.

1. Introduction

1.1. Problématique de la thèse

Le présent document porte sur l'étude numérique de différents matériaux composites. Naturellement, on peut commencer par se demander ce qu'est exactement un matériau composite. Au sens large, on peut le définir comme un matériau hétérogène constitué de différentes phases non miscibles. Ces différentes phases peuvent être de nature diverses : il peut s'agir de constituants à la composition chimique totalement différente comme ceux-ci peuvent avoir même composition mais uniquement présenter un agencement différent.

Au sens large, les matériaux composites sont présents depuis très longtemps dans les produits issus de la main humaine, de manière souvent fortuite, mais dont la nature des constituants apporte des avantages importants sur un matériau monophasé. Par exemple, les aciers sont bien souvent des matériaux composites, la structure cristalline du fer variant souvent au sein même de l'acier. D'un point de vue pratique, tout objet en aluminium peut être considéré comme un composite $Al - AlO_2$ (aluminium métal-alumine), dont les propriétés conjuguées apportent un fort intérêt pratique au matériau résultant, entre les propriétés mécaniques de l'aluminium métal et celles de protection face à la corrosion de l'alumine. Dans un autre registre, une éponge ou du bois peuvent être considérés comme des composites.

Cependant, l'essor des matériaux composites au sens où on l'entend actuellement (matériaux à structure microscopique) est relativement récent. En effet, les matériaux composites auparavant utilisés par l'être humain étaient des productions fortuites ou simples à réaliser (couche de surface ou matériau à l'échelle macroscopique), et la technologie ne permettait pas encore de concevoir et produire en quantité industrielle des matériaux composites diversifiés à l'échelle microscopique. Maintenant que celle-ci le permet, un énorme champ de recherche s'est ouvert, permettant de combiner nombres de constituants entre eux pour obtenir des matériaux aux propriétés très diverses et surtout inatteignables avec des matières « monolithiques ».

Bien que les matériaux composites permettent d'atteindre des propriétés indispensables pour des matériaux homogènes, ils posent aussi quelques problèmes pratiques :

- Les procédés de fabrications sont moins éprouvés et robustes que pour les matériaux homogènes.
- Assembler des matériaux composites peut être une opération au coût financier élevé. De plus, les constituants eux-mêmes peuvent être parfois rares et/ou chers.
- L'hétérogénéité des matériaux composites rend difficile l'anticipation des performances de ceux-ci.

Un problème de taille se pose alors. Supposons que l'on recherche un matériau possédant des propriétés précises, et une série de matériaux composites candidats à répondre au problème. Il est alors difficile de déduire simplement lequel des candidats sera le plus approprié, car il est ardu de prévoir « à vue de nez » le comportement d'un composite. D'autre part, préparer un procédé d'élaboration pour chaque composite, en fabriquer des échantillons puis les tester peut vite se révéler contraignant autant d'un point de vue temporel que financier.

Comment s'y prendre dans ce cas ? L'utilisation des matériaux composites se réduirait-elle à celle des matériaux simples et économiques à produire ? Heureusement non, et c'est là que vient tout l'intérêt d'une étude numérique. Celle-ci permet de modéliser les matériaux et de tester leur comportement sur ordinateur face à différentes sollicitations représentées mathématiquement. Grâce à cela, il est possible de vérifier que tel ou tel matériau composite devrait, en théorie, atteindre les propriétés recherchées. Compte-tenu des erreurs de modélisation et du côté « parfait » du matériau créé (qui ne sera probablement pas exactement semblable à ce que rendra un procédé de fabrication réel), il est toujours nécessaire de se donner une petite marge d'erreur, et d'étudier quelques choix judicieux.

Cependant, le travail de la simulation numérique ne s'arrête pas là, et il est encore possible d'utiliser avantageusement la numérisation. En effet, une fois le choix de matériau réduit, quelques échantillons peuvent être réalisés afin de tester leurs propriétés réelles.

Mais ici, on est encore loin de la fabrication à grande échelle, et les échantillons sont généralement de petite taille, ce qui ne permet pas de vérifier le comportement sur de grandes structures. On se sert alors des données expérimentales recueillies pour numériser fidèlement le produit créé. Et grâce à l'outil informatique, les propriétés de structures aux échelles supérieures peuvent être simulées.

Finalement, la réalisation des échantillons permet aussi un retour d'expérience qui permet d'améliorer les codes de calculs afin de les rendre plus prédictifs, afin de réduire encore dans le futur le nombre de matériaux candidats à un problème donné. De même, la modélisation peut aussi permettre de valider des procédés de fabrication. L'étude des échantillons fabriqués permet alors de constater si ceux-ci vérifient bien le cahier des charges (nature des constituants, propriétés, position, quantité, etc).

La problématique de cette thèse porte sur cette étude des échantillons. Afin de pouvoir comparer de manière simple et automatique les matériaux virtuels et réels, il est utile de chercher à numériser ces matériaux, ce qui facilitera ensuite la comparaison. Cette numérisation, qui permettra d'obtenir des images en trois dimensions représentant correctement le matériau réel, peut être effectuée de différentes manières. Ici, le procédé utilisé est la tomographie X, qui permet de retourner une image en 3D de l'objet étudié. Pour ce faire, on commence par mesurer l'absorption aux rayons X de l'objet dans une direction donnée, ce qui permet d'obtenir une image plane (une projection). Avec plusieurs projections dans plusieurs directions, il est alors possible de représenter le volume 3D via des algorithmes de reconstruction.

On recherche alors une manière d'exploiter ces images de matériaux composites réels afin de les comparer aux structures virtuelles originellement créées. Les informations directement accessibles sur le matériau composite réel à partir des images tomographiques sont assez peu nombreuses. En effet, ces images ne permettent en elles-mêmes que de quantifier de manière locale la valeur d'une propriété particulière du matériau (dans le cas de la tomographie X, il s'agit de l'absorption aux rayons X), représentée alors par un niveau de gris. En ne considérant que les images brutes, il est donc uniquement possible de comparer l'absorption locale du matériau réel et de la structure virtuelle.

Dans les cas étudiés ici, les matériaux virtuels imaginés ont le plus souvent une structure déterminée qu'il est possible de décrire plus simplement que voxel par voxel. Ceux-ci sont généralement conçus comme un agencement de différents constituants présents sous la forme de structures géométriques caractérisées par un jeu de paramètres simples. La connaissance simultanée de l'ensemble des caractéristiques des formes présentes dans le matériau et des propriétés physiques des constituants permet alors de construire une image virtuelle permettant la modélisation physique, où chaque voxel peut se voir attribuer des propriétés physiques locales. Celles-ci dépendent bien entendu de l'élément présent au voxel considéré, mais aussi de la position relative de ce voxel dans la forme géométrique associée. En effet, les différents constituants ne sont pas forcément autant de phases isotropes mais ont en réalité des propriétés dépendant intrinsèquement de la structure géométrique choisie.

Ainsi l'image acquise à partir d'un matériau réel élaboré ne peut permettre à elle seule de caractériser les propriétés physiques en chaque voxel, car la connaissance unique d'un niveau de gris local ne permet pas de déterminer la localisation et l'orientation des constituants. Toute la problématique réside ici dans la recherche de localisation et d'orientation. La première permet de retrouver ensuite, pour chaque voxel, le constituant présent, alors que la seconde, qui découle de la première, permet de déterminer enfin les propriétés physiques précises d'un voxel quelconque du matériau réel.

Ce document porte principalement sur l'étude de deux types de matériaux composites :

- Des matériaux fibreux, constitués en général de deux phases, une composant les fibres, baignant dans la seconde phase, à laquelle s'ajoute parfois des pores, c'est-à-dire une troisième phase correspondant au vide. Dans ce type de matériau, le constituant de la matrice possède généralement des propriétés isotropes, alors que les fibres, correspondant à des structures cylindriques, possèdent des propriétés potentiellement différentes dans les trois directions radiales, axiales et orthoradiales. La problématique sur ces matériaux consiste donc en la détection des structures fibreuses, que l'on peut souvent associer à des formes cylindriques appuyées sur des courbes. La connaissance précise des

cylindres et de leurs paramètres permet alors, pour chaque point d'une fibre quelconque, de caractériser son repère cylindrique associé et par conséquent d'en déduire ses propriétés dans un repère cartésien donné.

Dans certains de ces matériaux, les fibres seront de plus réunies sous forme d'ensemble cohérents en espace et en orientation, correspondant à des fils. La détermination de ces fils peut permettre de modéliser fidèlement les propriétés du matériau à une échelle supérieure. On recherche par conséquent à détecter ces fils afin de simplifier la représentation de celui-ci, et ainsi réduire la complexité des problèmes de simulation physique de structures macroscopiques.

- D'autre part, des mousses sont aussi étudiées. Celles-ci présentent le plus souvent des formes plus ou moins sphériques (de la sphère quasi-parfaite au polyèdre en passant par l'ellipsoïde), qui peuvent être aussi bien des billes d'un constituant au milieu d'une matrice que des formes creuses, voire des structures remplies du même constituant que la matrice, la différence de composition ne se faisant alors qu'aux frontières de ces formes sphériques. La problématique sur ce type de matériau est donc de détecter les sphères, leurs positions et leurs rayons. Ces informations permettront ensuite de vérifier le procédé de conception via la distribution des rayons des sphères et par leur répartition plus ou moins homogène, afin de modéliser le comportement physique de ces matériaux et de générer ensuite virtuellement des matériaux aux distributions proches.

Dans les deux cas présentés ci-dessus, on constate que la reconnaissance de formes géométriques est au cœur du problème. Celle-ci nous permettrait donc de caractériser plus simplement les matériaux et de les modéliser à une échelle supérieure, permettant une détermination des propriétés physiques plus aisée. Il est alors possible de vérifier le procédé de fabrication en comparant les résultats obtenus et le matériau virtuel original. On peut aussi valider les codes de calcul numériques en comparant le comportement de l'échantillon réel et celui obtenu à partir de la numérisation.

1.2. Etat de l'art

1.2.1. Introduction

La détection et la reconnaissance de formes est un domaine de recherche assez récent. Toutefois, ce domaine a évolué très rapidement en ce qui concerne la 2D, poussé par une demande importante. On peut par exemple penser aux algorithmes de reconnaissances de visages qu'utilisent les réseaux sociaux, ou leur détection par les appareils photos, comme applications courantes de la reconnaissance de formes. Dans des domaines plus spécialisés de la haute technologie, la détection automatique de formes est aussi très étudiée, que ce soit en robotique, vision assistée par ordinateur, traitement automatique d'images aériennes, reconnaissance d'empreintes digitales... On peut distinguer deux types d'utilisations du domaine :

- La reconnaissance de formes simples : elle permet de s'adapter à des images possiblement très diverses. Par conséquent, elle est peu discriminante et ne permet pas de reconnaître précisément un objet, une personne ou une situation. Les méthodes associées sont souvent analytiques ou géométriques, basées sur les propriétés mathématiques des objets à chercher.
- La reconnaissance de formes complexes : elle permet de détecter précisément un nombre limité de types d'objets différents. Elle est généralement précise, se « trompant » rarement, mais n'est adaptée qu'à une tâche particulière, comme reconnaître des visages. Les méthodes associées sont plus souvent fondées sur l'apprentissage (*machine learning*), qui consiste à apprendre à la machine à identifier un type d'objet par des images tests pour pouvoir ensuite le reconnaître.

Le développement de l'imagerie en 3D étant relativement récent, la majorité des méthodes de reconnaissance de formes est adaptée à des images en deux dimensions, ou en deux dimensions stéréographiques afin de recréer un effet 3D. D'ailleurs, dans beaucoup de cas, l'imagerie 3D n'est pas utilisable car son acquisition est contraignante (comme dans le cas de robots qui se déplacent par exemple). A ce moment-là, la 2D stéréographique est bien plus souple que la 3D brute.

Cependant, quand les contraintes d'acquisition d'une image 3D ne sont pas importantes ou sont compensées par les bénéfices de l'imagerie 3D (imagerie médicale, imagerie de matière inanimée), il devient nécessaire de développer des méthodes de reconnaissance de formes adaptées. La plus grande difficulté du passage de la 2D à la 3D se trouve dans l'explosion du nombre d'éléments de l'image, qui peut rendre un algorithme raisonnable en 2D inutilisable en 3D car trop long ou trop coûteux en mémoire. Dans cet ordre d'idée, on remarquera qu'un algorithme de complexité en $O(N^2)$ s'exécutant en 0.1 s sur une image de $1000 * 1000$ pixels mettra 27 h sur une image 3D de $1000 * 1000 * 1000$ voxels ! En comparaison, un algorithme de complexité en $O(N)$ s'exécutant en 10 s sur une image de $1000 * 1000$ pixels mettra moins de 3 h à s'effectuer en 3D.

La reconnaissance de formes en 3D doit donc ériger en principe fondamental la **complexité asymptotique** des algorithmes. Celle-ci, déjà importante en 2D, pouvait cependant être encore ignorée sur des images ne dépassant que très rarement les 10^7 pixels. En 3D, les images peuvent atteindre les 10^{11} voxels, ce qui rend le contexte asymptotique bien plus important.

Dans le cadre de ce manuscrit, les images étudiées se prêteront au jeu de la reconnaissance de formes simples en trois dimensions : sphères, ellipsoïdes, cylindres, droites de l'espace, courbes... Des méthodes en trois dimensions existent déjà, qu'elles soient dérivées de la Transformée de Hough (Hough, 1962), qui fera l'objet du point suivant, ou du tenseur de structure, objet du point 3. La détection de contours, par son aspect local, ne peut pas être considérée à proprement parler comme de la reconnaissance de formes, mais la connaissance des contours est une étape préliminaire très souvent nécessaire à celle-ci, et permet tout autant d'attacher à une image des propriétés. Par conséquent, elle fera l'objet de la partie 2.5.

1.2.2. Formalisme

Dans la suite du manuscrit, à moins que des mentions contraires ne soient précisées, on se donnera un ensemble de définition $D = \prod_{i < n} [0 ; L_i + 1[\subset \mathbb{R}^n$. On se donnera de plus une fonction $f : D \mapsto \{0 ; 1\}$. On notera $f^{-1}(1)$ le sous-ensemble de D réciproque de 1 par f et $f^{-1}(0)$ le sous-ensemble de D réciproque de 0 par f .

Une forme Ω de \mathbb{R}^n sera définie comme un ensemble quelconque de points de \mathbb{R}^n . On dira que la forme Ω apparaît dans (ou par abus de langage appartient à) l'image issue de f si et seulement si :

$$\Omega \cap f^{-1}(1) = \Omega \cap D$$

A partir de l'ensemble D , on se donnera un sous-ensemble réduit à \mathbb{N}^n que l'on notera $E = D(J) = \prod_{i < n} \llbracket 0 ; L_i \rrbracket \subset \mathbb{N}^n$ (avec $\llbracket 0 ; L_i \rrbracket$ l'ensemble des entiers compris entre 0 et L_i). A partir de f , on construira l'image numérique J définie sur E comme :

$$J(x_0, x_1 \dots x_{n-1}) = \begin{cases} 1 & \text{si } \exists w \in \prod_{i < n} [x_i ; x_i + 1[\text{ tel que } f(w) = 1 \\ 0 & \text{si } \forall w \in \prod_{i < n} [x_i ; x_i + 1[, f(w) = 0 \end{cases}$$

On notera de la même manière $J^{-1}(1)$ le sous-ensemble de $D(J)$ réciproque de 1 par J et $J^{-1}(0)$ le sous-ensemble de $D(J)$ réciproque de 0 par J .

On construira finalement de même la forme Γ issue de Ω comme :

$$\Gamma = \left\{ x = (x_0, x_1 \dots x_{n-1}) \in \mathbb{Z}^n \text{ tel que } \prod_{i < n} [x_i ; x_i + 1[\cap \Omega \neq \emptyset \right\}$$

Ainsi, si Ω est une forme de f , alors Γ est une forme de J .

1.2.3. Transformée de Hough

1.2.3.1. Présentation

La Transformée de Hough est une méthode de détection de formes inventée par Hough en 1962 (Hough, 1962) et popularisée en traitement d'images en 1969 par Rosenfeld (Rosenfeld, 1969) (Illingworth & Kittler, 1988). Depuis, de très nombreuses techniques de reconnaissance de formes fondées sur cette méthode ont été développées et améliorées (Illingworth & Kittler, 1988) (Mukhopadhyay & Chaudhuri, 2015). Le principe de la Transformée de Hough est d'envoyer l'image sur un espace de paramètres représentant les objets que l'on recherche.

On recherche parmi $f^{-1}(1)$ plusieurs occurrences d'un type de forme simple **mesurable et pouvant être totalement déterminée par une équation analytique à m paramètres réels** (comme une droite du plan peut par exemple être représentée par une équation analytique à deux paramètres réels $(m, b) : y = mx + b$), et on notera Ω^y une de ces formes déterminée par le jeu de paramètres $y \in \mathbb{R}^m$. Soit $x \in f^{-1}(1)$. On notera $\mathcal{E}(x)$ l'ensemble suivant :

$$\mathcal{E}(x) = \{y \in \mathbb{R}^m \text{ tel que } x \in \Omega^y\}$$

puis on définira finalement la fonction $a : \mathbb{R}^m \rightarrow \mathbb{R}$ par :

$$\forall y \in \mathbb{R}^m, a(y) = \int \{f(x) = 1 \text{ et } y \in \mathcal{E}(x)\} dx$$

i.e. $a(y)$ est la mesure de l'ensemble des $x \in f^{-1}(1)$ tel que $y \in \mathcal{E}(x)$, i.e. tel que $x \in \Omega^y$, donc on peut par conséquent aussi écrire :

$$\forall y \in \mathbb{R}^m, a(y) = \int_{D(f)} f(x) \cdot 1_{\Omega^y}(x) dx$$

On notera alors la transformée de Hough $t : \mathbb{R}^m \rightarrow [0 ; 1]$ comme la fonction définie par :

$$\forall y \in \mathbb{R}^m, t(y) = \frac{\int_{D(f)} f(x) \cdot 1_{\Omega^y}(x) dx}{\int 1_{\Omega^y}}$$

Cette formulation intégrale de la transformée de Hough a été proposée formellement par Princen, Illingworth et Kittler (Princen, et al., 1992), et permet de voir la transformée comme le résultat d'une convolution de la fonction de départ par un noyau associé à la forme à détecter.

Les images réelles étant des objets discrets, on discrétisera la fonction f en une image binaire J .

Par la discrétisation, l'ensemble \mathcal{E} devient un ensemble \mathcal{E}' défini par :

$$\mathcal{E}'(x) = \{y \in \mathbb{R}^m \text{ tel que } x \in \Gamma^y\}$$

avec Γ^y la discrétisation de Ω^y définie dans le formalisme.

On obtient alors pour a :

$$\forall y \in \mathbb{R}^m, a(y) = \text{card}\{x \in J^{-1}(1) \text{ tel que } y \in \mathcal{E}'(x)\}$$

$a(y)$ représente donc le nombre de points de $J^{-1}(1)$ appartenant à la forme Γ^y , c'est-à-dire que :

$$a(y) = \text{card}(\Gamma^y \cap J^{-1}(1))$$

Via l'équation de la surface de la forme Ω^y , on peut aussi calculer $q(y)$ le cardinal de l'ensemble $\Omega^y \cap D(J)$. De là, on peut dire que la forme Ω^y apparaît dans l'image si et seulement si

$$\frac{a(y)}{q(y)} = 1$$

$t = \frac{a}{q}$ est la Transformée de Hough, et permet de déterminer, pour chaque jeu de paramètres y , si la forme Ω^y appartient à l'image J .

Dans les faits, on commencera par discrétiser et borner l'espace de paramètres (heureusement, sinon l'ensemble de formes passant par un point est généralement infini), ce qui permettra de passer d'une image J de taille $\prod_{u < n} [0 : L_u]$ à une image T , couramment appelée **accumulateur de Hough**, à valeurs dans $\mathbb{Q} \cap [0 ; 1]$ et de taille $\prod_{v < m} [0 : K_v]$, et qui permet facilement de trouver les formes présentes dans l'image.

De plus, on sera généralement plus tolérant sur la présence ou non d'une forme dans une image. En effet, en raison du bruit inhérent aux acquisitions, un paramètre de tolérance ε sera souvent fixé, et on acceptera alors une forme si et seulement si on a :

$$t(y) \geq 1 - \varepsilon$$

Dans de nombreux cas le critère que l'on se donne et le type d'image étudié permettent de négliger l'apport de q , ce qui nous permet de travailler dans \mathbb{N} avec une condition du type

$$a(y) \geq C_0$$

Le pseudo-code de l'algorithme associé à la Transformée de Hough est assez simple :

```

Entrée :  $J, \varepsilon$ 
Créer  $T$ 
Pour tout  $x$  tel que  $J(x)=1$ 
    Pour tout  $y$  tel que la forme définie par  $y$  passe par  $x$ 
         $T(y) \leftarrow T(y)+1$ 
Pour tout  $y$ 
     $T(y) \leftarrow T(y)/Q(y)$  (avec  $Q$  la normalisation par la mesure de
    la forme)
    Si  $T(y) \geq 1 - \varepsilon$ 
        La forme définie par  $y$  appartient à  $J$ 

```

Les deux premières boucles imbriquées permettent de calculer l'accumulateur de Hough (encore non-normalisé), puis la troisième trouve les formes appartenant à l'image.

Le principe de la méthode de Hough est en effet d'arriver au calcul de $a(y)$ non pas par le chemin « classique » consistant à calculer $\text{card}(\Gamma^y \cap J^{-1}(1))$, mais en passant par la formulation $a(y) = \text{card}\{x \in J^{-1}(1) \text{ tel que } y \in \mathcal{E}'(x)\}$. En effet, on peut alors procéder point par point sur $J^{-1}(1)$, en incrémentant l'accumulateur de 1 à chaque fois qu'il y a lieu. Chaque point **vote** pour un certain nombre de formes, et la **détection de pics** permet alors de trouver les formes.

D'un point de vue complexité et temps de calcul, on peut commencer par justifier l'intérêt de la méthode par quelques réflexions sur le fonctionnement de l'algorithme :

- Pourquoi prendre les points de l'image de base, puis vérifier toutes les formes y passant ? Pourquoi ne pas simplement résoudre le problème de manière directe, i.e. en prenant tous les jeux de paramètres et en vérifiant qu'ils appartiennent à l'image directement ?

→ Pour des raisons de quantité de calculs. Dans le cas naïf qui consisterait à tester chaque forme, on parcourrait pour chaque élément de l'accumulateur, l'ensemble de l'image. Dans le cas de la Transformée de Hough, on peut utiliser la

connaissance de l'image, et ne regarder que les points de l'image appartenant à $J^{-1}(1)$. On réduit donc la quantité de calculs d'un facteur $\frac{\text{card } D(J)}{\text{card } J^{-1}(1)}$.

- Dans ce cas, le fait de ne pas pouvoir calculer au même moment le facteur $q(y)$ de normalisation, contrairement à la méthode naïve, ne fait-il pas perdre l'intérêt de l'algorithme ? Il est nécessaire, pour calculer $q(y)$, de parcourir à nouveau l'espace $D(J)$.

→ Dans le cas de formes simples, le calcul du facteur de normalisation est très simple, et beaucoup de jeux de paramètres ont la même valeur de q . Celui-ci est même couramment omis.

D'un point de vue de la complexité asymptotique, on supposera $D(J)$ et T comme deux hypercubes de dimensions respectives L^n et K^m . On notera $C_{L,K}$ la complexité de l'algorithme.

La phase de recherche des pics de l'accumulateur est négligeable devant la phase de calcul de l'accumulateur, ainsi, la complexité est dominée par le nombre d'opérations de ce dernier, qui est de $\text{card}(J^{-1}(1)) \times K^{m-1}$. Ainsi, on a :

$$C_{L,K} = \Theta(\text{card}(J^{-1}(1)) \times K^{m-1})$$

Si l'on suppose assez logiquement que le cardinal de $J^{-1}(1)$ varie linéairement avec la taille de l'image, alors on obtient finalement :

$$C_{L,K} = \Theta(L^n K^{m-1})$$

Cette complexité est remarquablement élevée. En effet, il est extrêmement rare qu'une forme à repérer dans une image n'ait pas au moins autant de paramètres qu'un point de l'image a de coordonnées, i.e. la dimension de l'accumulateur est quasiment toujours supérieure ou égale à celle de l'image. Cela équivaut à dire qu'on a, pour toutes les formes « classiques », $m \geq n$. On prendra généralement un pas de discrétisation assurant $K = L$. Ainsi, on peut assurer que :

$$L^{2n-1} = O(C_{L,L})$$

La complexité se comporte donc asymptotiquement plus mal que la fonction $x \mapsto x^{2n-1}$, pourtant déjà peu intéressante. De fait, le passage de la 2D à la 3D ne se fait qu'avec une explosion de la complexité. Si l'on recherche une structure plane repérée par m paramètres dans une image de dimension 2, on a alors une complexité vérifiant $C_{L,L} = \Theta(L^{m+1})$. Si l'on recherche cette même forme plane dans une image de dimension 3, on rajoute une dimension à l'image, et la forme ne peut plus être repérée par m paramètres mais par $m + p$ paramètres, où p est le nombre de paramètres permettant de repérer un plan dans l'espace. Un plan peut être repéré par 3 paramètres, deux angles et une distance, ce qui donne finalement une complexité vérifiant $C'_{L,L} = \Theta(L^{m+5})$, donc :

$$\frac{C'_{L,L}}{C_{L,L}} = \Theta(L^4)$$

Cet algorithme est-il donc bien adapté à la recherche de formes en trois dimensions ? Dans sa forme la plus ancienne, il semble difficile de faire fonctionner cette méthode sur des images réelles en trois dimensions. Cette méthode reste cependant adaptée pour des images telles que $\text{card}(J^{-1}(1))$ est très faible devant la taille de l'image.

La Transformée de Hough présente tout de même des avantages intéressants, souvent inhérents au traitement point par point de la méthode (Illingworth & Kittler, 1988) :

- La méthode de Hough est assez peu sensible au bruit (Shapiro, 1974) (Cohen & Toussaint, 1977). Le traitement point par point permet en effet de perdre une partie de la forme sans pour autant détruire l'information générale.
- L'algorithme déduit est hautement parallélisable en raison du traitement indépendant de chaque point.
- La valeur de t permet de qualifier la proximité de l'image et la forme recherchée.
- Découlant du point précédent, il est simple de détecter des formes légèrement déformées grâce à la transformée de Hough.

A contrario, la transformée de Hough a des défauts issus eux aussi du traitement indépendant de chaque point :

- Taille rapidement prohibitive de l'accumulateur de Hough.
- Temps de calcul élevé, complexité médiocre.

- Méthode peu fiable pour des images très denses. Si les distances entre formes consécutives sont très faibles, l'algorithme a la possibilité de détecter des formes dans n'importe quel sens.
- Il est impossible de détecter des objets trop déformés.

1.2.3.2. Transformée de Hough pour la détection de segments

Historiquement, la première utilisation de la Transformée de Hough (qu'on notera souvent HT, pour *Hough Transform*) ne concerne pas uniquement la détection de droites et de segments dans des images en deux dimensions, mais celle-ci est cependant l'une des utilisations les plus évidentes. Pour effectuer cette recherche, il est nécessaire de se donner une manière de caractériser une droite avec le moins de paramètres possible. Dans le cas d'une droite du plan, il est possible de la repérer à l'aide d'une longueur et d'un angle (Duda & Hart, 1972) :

On se fixe, dans un repère orthonormé (O, \vec{i}, \vec{j}) une droite (d) . On notera E la projection de O sur (d) , $\rho = OE$ et $\theta = (\vec{i}, \overrightarrow{OE})$. La droite (d) vérifie alors l'équation suivante :

$$x \cos(\theta) + y \sin(\theta) = \rho$$

La droite peut donc être totalement caractérisée par la connaissance des deux paramètres (ρ, θ) (Figure 1). Cette méthode de paramétrisation proposée au départ par Duda et Hart (Duda & Hart, 1972) a l'avantage de ne pas présenter de singularité comme pourrait le faire une paramétrisation avec d'autres jeux de paramètres. Par exemple, le jeu (m, b) représentant une droite par l'équation $y = mx + b$ pose problème pour des droites parallèles à l'axe des ordonnées (Illingworth & Kittler, 1988).

D'autres paramétrisations sont cependant utilisables, comme celle proposée par Wallace (Wallace, 1985), utilisant les intersections des lignes avec les bords de l'image. Celle-ci présente l'avantage de faire évoluer les paramètres dans un sous-ensemble de l'ensemble de définition de l'image, ce qui permet de conserver les mêmes discrétisations pour l'image de base et pour l'accumulateur. Cette paramétrisation est par conséquent complètement adaptée par construction à la structure discrète des images réelles. Dans le cas de la

paramétrisation en (ρ, θ) , les erreurs et approximations dues à la discrétisation de l'espace de paramètres ont été étudiées par Van Veen et Groen (Van Veen & Groen, 1981).

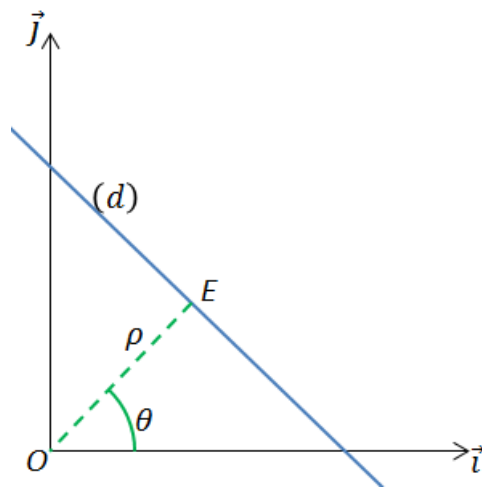


Figure 1 : Schéma explicatif de la paramétrisation en (ρ, θ) d'une droite (d) .

En suivant la paramétrisation en (ρ, θ) , la transformée de Hough d'un point est une sinusoïde. On constatera que la Transformée de Hough est une transformée de Radon (Deans, 1981) (Deans, 2007) (Radon, 1994). Cependant, la finalité et la manière de considérer ces deux transformations n'est pas la même. Dans un but pratique, la transformée de Hough associe à un point de l'image de base un ensemble de l'espace de paramètres, alors que la transformée de Radon en tomographie est vue comme l'association d'un ensemble de points de l'image de départ à un seul point de l'espace d'arrivée. De ce fait, même si le résultat de ces deux transformations est le même, elles ne sont pas écrites de la même manière, et ne sont donc pas équivalentes une fois discrétisées, comme cela a été démontré en 1996 par Toft et Sørensen (Toft & Sørensen, 1996). D'un point de vue temporel, la différence est immédiatement visible : l'algorithme de Hough ne travaillant que sur les points d'intérêt de l'image, il est bien plus rapide qu'un algorithme implémentant la transformée de Radon dans tous les cas. Au niveau des résultats, on peut voir que :

- La transformée de Radon est précise pour des espaces de paramètres sur-échantillonnés. En effet, les algorithmes issus de la transformée de Radon utilisent comme base l'espace de paramètres, ce qui lui permet de raffiner les résultats. En revanche, il est mauvais sur les espaces sous-échantillonnés, car il est alors facile de passer à côté des paramètres exacts et par conséquent de manquer les pics.

- A l'inverse, la transformée de Hough est assez médiocre pour des espaces sur-échantillonnés. En effet, l'algorithme issu de la TH est fondé sur les points de l'image. Comme le nombre de ceux-ci ne bouge pas, l'augmentation de la résolution de l'espace de paramètres va provoquer un étalement des pics, qui seront alors difficiles à repérer. Cependant, pour des espaces sous-échantillonnés, la méthode réunit alors les droites de paramétrisations proches en un seul point, et les pics ressortent alors très facilement de l'image.

Il existe cependant une résolution particulière pour laquelle la transformée de Hough discrète et de Radon discrète donnent exactement le même résultat (Princen, et al., 1992).

1.2.3.3. Transformée de Hough circulaire et ellipsoïdale

La transformée de Hough n'est pas adaptée qu'à la recherche de droites, elle permet aussi de détecter des cercles et des ellipses dans des images 2D (Illingworth & Kittler, 1988) (Mukhopadhyay & Chaudhuri, 2015) (Yuen, et al., 1990).

De la manière la plus simple possible, un cercle dans un plan peut être paramétré par un triplet (x, y, r) , où (x, y) représentent les coordonnées du centre du cercle et r son rayon (Hough, 1962) (Illingworth & Kittler, 1988). L'espace de paramètres associé à la recherche de cercles sur une image 2D est donc une image 3D. Pour tout point de coordonnées (a, b) à considérer sur l'image, l'ensemble des cercles passant par ce point correspond à l'ensemble des paramètres (x, y, r) tel que $(x - a)^2 + (y - b)^2 = r^2$. Dans l'espace de paramètres, cet ensemble correspond à un cône de sommet $(a, b, 0)$, et on retrouve les paramètres du cercle à l'intersection de tous les cônes issus de chaque point (Figure 2). Dans ce cas de recherche de cercles, il est commun de prendre le même pas de discrétisation pour l'espace de paramètres et pour l'image de départ. Cependant, si l'on cherche des parties de cercles dont le centre n'est pas inclus dans l'image, il est nécessaire d'augmenter la taille de l'accumulateur. Sur une image de taille L^2 sur laquelle on recherche des cercles de rayon maximal R , l'accumulateur est donc une matrice de dimension trois de taille comprise entre L^2R et $(L + 2R)^2R$.

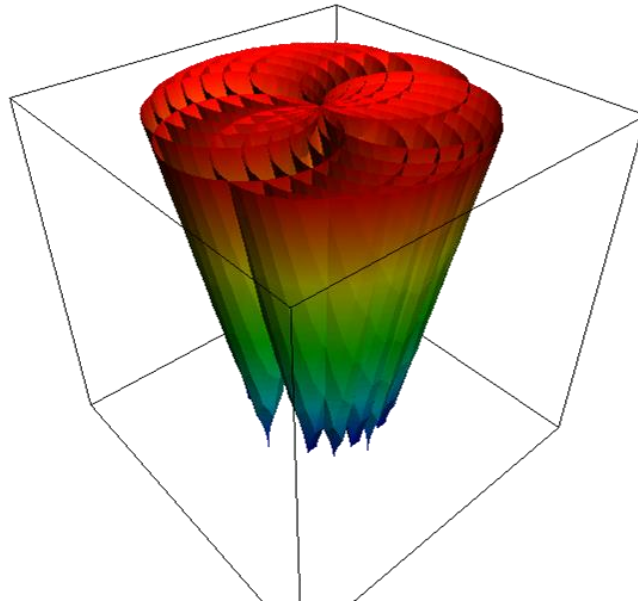


Figure 2 : Vue de l'accumulateur de Hough pour une recherche de cercles. On constate que les cônes associés à chaque point du cercle se rencontrent tous.

Pour sa part, la recherche d'ellipses à partir de la transformée de Hough (Duda & Hart, 1972) demande 5 paramètres, deux pour le centre, deux de forme (les demi-grands axes par exemple) et un d'orientation.

La recherche d'une structure à cinq paramètres demande, si on la réalise de manière brute, une très grande capacité de calcul et surtout de mémoire. Heureusement, différentes manières de procéder ont vu le jour, permettant d'exploiter les propriétés particulières de l'ellipse pour en retrouver les paramètres en plusieurs étapes, réduisant ainsi la taille des accumulateurs à considérer. En particulier, la méthode proposée par Yuen, Illingworth et Kittler (Yuen, et al., 1989) permet de rechercher d'abord les coordonnées du centre avant d'en rechercher les paramètres de forme. Pour cela, la propriété suivante de l'ellipse est utilisée (Figure 3) :

soit \mathcal{E} une ellipse de centre O et P, Q deux points quelconques de l'ellipse. On note $\tau(\mathcal{E}, A)$ la tangente à \mathcal{E} en A . Soit T l'intersection de $\tau(\mathcal{E}, P)$ et $\tau(\mathcal{E}, Q)$ et M le centre du segment $[PQ]$. On peut alors affirmer :

$$O \in (MT)$$

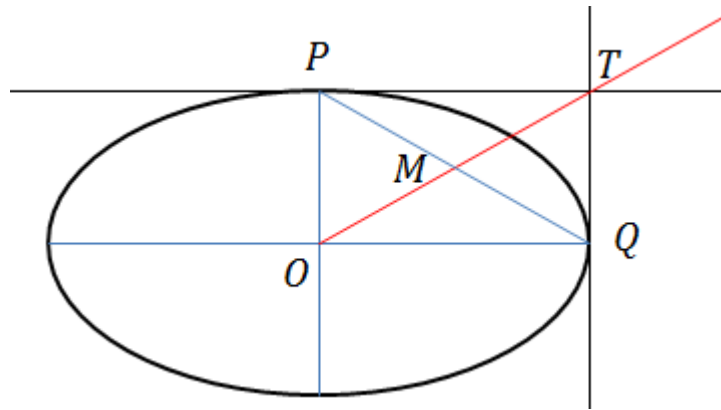


Figure 3 : Schéma explicatif de la méthode permettant de séparer en deux HT la détection d'ellipses. Le centre O de l'ellipse appartient à la droite (MT) , ce qui permet, à partir de deux doublets de points (P, Q) , d'obtenir deux droites sécantes en O .

Il est à noter que cette propriété, si elle permet bien de passer d'un accumulateur de dimension 5 à deux accumulateurs successifs de dimensions respectives 2 et 3 (voire en trois accumulateurs successifs de dimensions 2, 1 et 2 en récupérant l'orientation avant de rechercher les paramètres de forme de l'ellipse (Muammar & Nixon, 1989)), demande cependant de pouvoir calculer précisément la tangente en un point considéré de l'ellipse sans avoir connaissance de celle-ci. Des méthodes locales de calcul de tangente sont alors utilisées, mais leur efficacité est limitée en présence de bruit et ne peuvent renvoyer qu'un nombre très restreint de tangentes possibles selon la taille du voisinage local considéré. Les avantages à utiliser la transformée de Hough sur une image bruitée peuvent alors être perdus.

De manière générale, les améliorations algorithmiques sur la transformée de Hough ellipsoïdale consistent à séparer la recherche des deux paramètres de localisation de la recherche du paramètre d'orientation et de ceux de forme.

Une autre méthode permettant de diviser la détection en deux consiste à rechercher le centre d'une ellipse par des propriétés globales. En particulier, le centre d'une ellipse est aussi le point d'intersection des deux axes de symétrie (qui sont de plus orthogonaux) de celle-ci, ce qui nous permet de le retrouver plus simplement. La recherche du centre d'une ellipse s'exprime donc sous la forme de la recherche d'une droite correspondant à un axe de symétrie puis au point d'intersection avec le second axe.

1.2.3.4. *Récapitulatif*

Finalement, la transformée de Hough ouvre un large champ de possibilités techniques pour la détection et la reconnaissance de formes (Illingworth & Kittler, 1988) (Mukhopadhyay & Chaudhuri, 2015). Toutes les méthodes dérivées de la transformée de Hough partagent un certain nombre de qualités :

- Simplicité de la méthode : la Transformée de Hough est facile à implémenter sur ordinateur. C'est une méthode de traitement « point par point », ce qui permet de la paralléliser aisément.
- Robustesse au bruit : la Transformée de Hough étant une transformation point par point, le bruit n'aura pas d'effet important sur la détection tant que le nombre de « bons points » reste significatif.
- Personnalisation et généralisation : la Transformée de Hough permet de détecter différentes formes, de jouer sur des seuils d'acceptation, et ainsi de varier la méthode selon les images à étudier.

Du côté des défauts, on notera :

- Complexité : le nombre d'opérations à effectuer par un algorithme de Hough est plus faible qu'avec une méthode en brute force, mais reste du même ordre de grandeur pour les HT simples. Dans le cas des HT plus spécifiques, il est possible d'y gagner, mais la complexité reste dans tous les cas un problème majeur.
- Mémoire : la taille de l'accumulateur d'une méthode de HT standard est généralement très importante, ce qui rend ces algorithmes inutilisables sur des machines « classiques ». Des méthodes dérivées corrigent en partie ce problème, mais l'occupation mémoire est toujours un défaut.
- Généralisation « payante » : plus l'algorithme de Hough implémenté se veut généraliste, plus le nombre d'opérations requises augmente.

Les défauts de la transformée de Hough font que son utilisation sur des images 3D brutes est assez complexe. Les problèmes de mémoire et complexité, déjà présents en 2D, se font aigus en 3D. De fait, la HT n'est pas une méthode « naturellement » adaptée au traitement des images en trois dimensions. Cependant, les diverses méthodes dérivées

implémentées au cours du temps montrent que l'utilisation de la HT n'est pas totalement à exclure, et qu'il est possible, moyennant quelques adaptations, de l'utiliser comme méthode principale ou auxiliaire de détection de formes sur des images 3D.

1.2.4. Tenseur de Structure

1.2.4.1. Présentation

La recherche et la détection de formes sur une image donnée consiste en une quête d'informations globales sur celle-ci. En effet, les formes recherchées dans le cadre de ce manuscrit sont des ensembles spatialement cohérents et de taille « raisonnable » (non négligeable devant la taille de l'image) d'une image donnée. En pratique, la détection de structures ne se fait que très rarement en considérant la forme en tant que telle comme une entité élémentaire indivisible. Dans un tel cas, effectivement, la vérification de la présence d'une forme Ω donnée sur une image f ne se fait que par un calcul global, i.e. via un calcul de mesure de $\Omega \cap f^{-1}(1)$. Cette vérification directe n'est pas sans défaut, déjà en raison du calcul de mesure, rarement anodin d'un point de vue algorithmique, mais aussi en raison du nombre de formes à vérifier sur une image si aucune hypothèse préalable n'est faite. La Transformée de Hough, présentée en partie 1.2.3, n'est elle-même qu'une manière détournée et plus efficace de calculer ces mesures, mais sans modifier le principe sous-jacent utilisé pour la détection de formes.

Pour autant, si nous nous attachons à la recherche de structures exprimables sous forme de volumes ouverts Ω^y dont le bord est une surface solution d'une équation $(e) : g^y(x) = 0$ à m paramètres, alors il devient envisageable de retrouver la forme grâce à des propriétés locales de celle-ci. En effet, si g^y est différentiable, alors il est possible, par exemple, de calculer le vecteur normal à la courbe en tout point, qui dépendra de y , ce qui fournira alors des équations vérifiées par le jeu de paramètres.

Il faut alors s'interroger sur les propriétés des formes issues de l'équation (e) , et en particulier sur leur caractérisation et paramétrisation. Le choix des paramètres qui représenteront la forme a en effet de l'importance. Ceux-ci doivent être les plus simples à calculer à partir des équations issues des calculs de différentielles, et avoir si possible du sens d'un point de vue géométrique.

A partir de cela, chaque point de la structure peut donc rendre une valeur du jeu de paramètres utilisé uniquement grâce à des connaissances locales de l'image, et la somme de ces contributions locales permettra alors de corroborer la forme détectée par chaque point séparément.

C'est dans ce contexte qu'a été introduit le tenseur de structure (Bigun & Granlund, 1987) (Kass & Witkin, 1987), ou matrice de moment deux, qui est une matrice issue du calcul du gradient d'une fonction, et qui permet de déterminer les directions dominantes du gradient et de quantifier la prédominance de celles-ci. Le tenseur de structure est depuis couramment utilisé pour la reconnaissance de structures particulières (Kass & Witkin, 1987) (Mulat, et al., 2007) (Bigun, et al., 2004) (Krause, et al., 2010).

On considérera la fonction indicatrice de l'ensemble Ω^y , notée 1_{Ω^y} , et on suppose que $\partial\Omega^y$ est une surface C^1 . On peut alors calculer le gradient de la distribution 1_{Ω^y} :

Théorème de Green : Soit $\Omega \subset \mathbb{R}^n$ un ouvert à bord de classe C^1 . Alors on a

$$\nabla(1_{\Omega}) = -\nu\sigma$$

où $\nu(x)$ est le vecteur unitaire normal à $\partial\Omega$ en x dirigé vers l'extérieur de Ω , et où σ est une distribution sur \mathbb{R}^n définie par

$$\langle \sigma | \phi \rangle = \int_{\partial\Omega} \phi ds$$

avec ϕ appartenant à l'espace des fonctions tests et ds l'élément de surface sur $\partial\Omega$ (voir Figure 4).

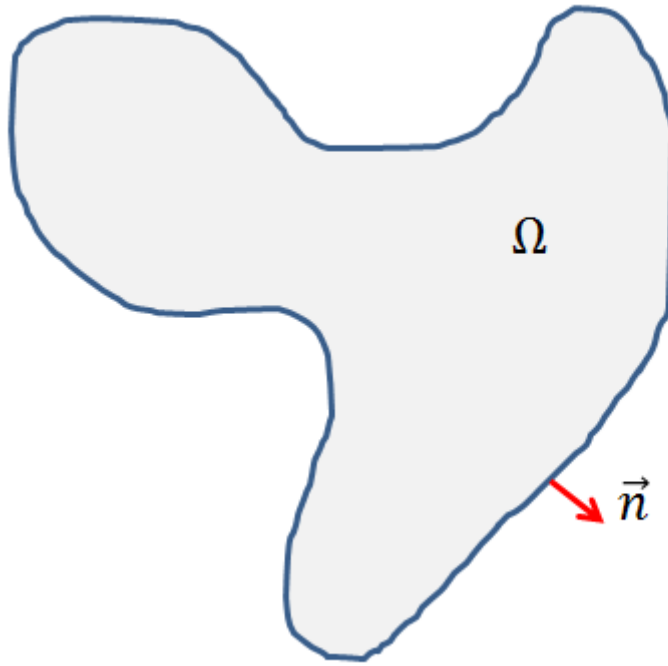


Figure 4 : Schéma illustrant le théorème de Green.

Le calcul du gradient associé à la forme Ω^y est donc directement orienté selon le vecteur normal à Ω^y . Or, g^y étant différentiable, on a finalement $\nabla(1_{\Omega^y}) = \sigma \nabla(g^y)$ sur $\partial\Omega^y$.

A partir de cette égalité, on peut finalement associer l'image réelle avec le type de forme à rechercher. En effet, la détection d'une structure sur une image f consiste à vérifier si pour un ensemble $\Omega \subset f^{-1}(1)$, il existe un jeu de paramètres y tel que $\Omega^y = \Omega$, i.e. tel que $1_{\Omega^y} = 1_{\Omega}$. Il est alors possible de démontrer que $\Omega^y = \Omega$ si et seulement si on a :

$$\begin{cases} \nabla(1_{\Omega}) = \sigma \nabla(g^y) & \text{sur } \partial\Omega^y \\ \nabla(1_{\Omega}) = 0 & \text{sinon} \end{cases}$$

Etant donné que l'on connaît l'équation de g^y et que 1_{Ω} est choisi sur l'image, on obtient donc pour chaque point de $\partial\Omega^y$ un système d'équations sur les paramètres y . Si pour tous les points de $\partial\Omega^y$, le système d'équations donne les mêmes propriétés sur y , alors Ω^y est bien une forme de l'image. A partir de Ω et d'une famille d'équations, on peut alors déterminer si Ω correspond bien à une des équations, et déterminer laquelle.

Dans le cas d'une image discrète J , on obtiendra une forme Γ discrétisée au lieu de Ω . La discrétisation nous apporte un avantage, c'est de pouvoir calculer numériquement une approximation de $\nabla(1_\Omega)$, car en pratique, le calcul de $\nabla(1_\Omega)$ sur une image continue est impossible. Cependant, ce calcul de gradient ne sera pas exact, étant donné les approximations dues à la discrétisation.

Dans le cadre continu, le tenseur de structure (Knutsson, 1989) sur une image f définie sur $D \subset \mathbb{R}^N$ est une matrice de taille $N \times N$ exprimable en tout point x de D par :

$$S_w(x) = \int w(x') (\nabla f(x - x'))^T (\nabla f(x - x')) dx'$$

i.e.

$$S_w = w * \nabla f^T \nabla f$$

où w est une distribution appelée « fenêtre » ou « fonction de poids » ($w(x) = \zeta(\|x\|)$), avec $\zeta : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ décroissante. ζ peut être par exemple une gaussienne centrée en 0 ou une fonction constante ou affine par morceaux). A^T désigne la transposée du vecteur A . On a ainsi, $\forall i, j < N$:

$$(S_w(x))_{i,j} = \int w(x') \times (\partial_{x_i} f \cdot \partial_{x_j} f)(x - x') dx'$$

Pour une image discrète J , le tenseur de structure est alors exprimable sous la forme :

$$S_w(x) = \sum_{x'} w(x') (\nabla f(x - x'))^T (\nabla f(x - x'))$$

où le gradient ∇f sera approché, par exemple par différences finies.

Par construction, S_w est une matrice symétrique donc diagonalisable. De plus, on peut montrer les propriétés suivantes sur S_w :

- Si w est positive, alors S_w est positive.

- Soit x fixé. Si $\forall i < N, \exists x'$ tel que $w(x') > 0$ et $\partial_{x_i} f(x - x') \neq 0$, alors $S_w(x)$ est définie positive.

w étant toujours choisi positif donc S_w est positive dans tous les cas. Ses valeurs propres $(\lambda_n)_{n < N}$ sont donc toutes positives ou nulles. Sa diagonalisation conduit à une matrice exprimant les composantes du gradient moyen au carré dans une base orthonormée de vecteurs propres $(e_n)_{n < N}$. Ainsi, le calcul du gradient moyen directionnel peut se faire via la décomposition du vecteur unitaire dans la base issue de la diagonalisation. De fait, la direction principale du gradient est donc celle du vecteur propre associé à la plus grande valeur propre de S_w .

Dans le cadre d'image en 3D, on a donc, pour chaque point, 3 couples de valeurs-vecteurs propres $(\lambda_1, e_1), (\lambda_2, e_2), (\lambda_3, e_3)$. Il est alors possible, selon les rapports entre les valeurs propres, de prédire la structure locale de l'image :

- Si λ_1 est très grand devant λ_2 et λ_3 , alors le gradient est quasi-totalement orienté selon e_1 . Les isosurfaces locales sont donc proches de plans orthogonaux à e_1 .
- Si λ_1 et λ_2 sont semblables et très grands devant λ_3 , alors le gradient n'est pas du tout orienté selon e_3 , et on a une image assez invariante par rotation dans les plans orthogonaux à e_3 . Ainsi, la structure locale donne des isosurfaces proches de cylindres orientés suivant e_3 .
- Si les trois valeurs propres sont semblables, alors la structure locale est proche d'une symétrie sphérique, la fonction étant localement isotrope.

La connaissance d'une base de vecteurs propres pour le gradient moyen peut donc nous donner des axes privilégiés, et par conséquent, celle-ci peut permettre de contraindre (voire de déterminer complètement) les paramètres y de la forme recherchée.

1.2.4.2. Détection de sphères via le tenseur de structure

Soit $C > 0, (a, b, c) \in \mathbb{R}^3$, et $\zeta : \mathbb{R} \rightarrow \mathbb{R}$ une fonction différentiable positive sur \mathbb{R}_+ . Soit $g : \mathbb{R}^3 \rightarrow \mathbb{R}$ définie par $g(u, v, w) = C - \sqrt{(u - a)^2 + (v - b)^2 + (w - c)^2}$ et $h : \mathbb{R}^3 \rightarrow \mathbb{R}$ définie par $h(u, v, w) = \zeta(g(u, v, w))$. On considère une image f binaire tel que $f(u, v, w) = 1$ si et seulement si $h(u, v, w) > 0$. On cherche alors à caractériser le fait que

$\Omega = f^{-1}(1)$ corresponde à une boule, et on cherche à retrouver des informations sur son centre à partir de la connaissance locale de $\partial\Omega$ et de h .

h est différentiable en tout point et on peut exprimer le gradient en tout point comme :

$$\nabla h(u, v, w) = \zeta'(g(u, v, w)) \frac{1}{g(u, v, w) - C} \cdot (u - a, v - b, w - c)$$

i.e.

$$\nabla h(u, v, w) = \zeta'(g(u, v, w)) \nabla g(u, v, w)$$

et par conséquent, on a :

$$S = \nabla h^T \cdot \nabla h = \left[\frac{\zeta'(g(u, v, w))}{g(u, v, w) - C} \right]^2 \times \begin{pmatrix} (u - a)^2 & (u - a)(v - b) & (u - a)(w - c) \\ (u - a)(v - b) & (v - b)^2 & (v - b)(w - c) \\ (u - a)(w - c) & (v - b)(w - c) & (w - c)^2 \end{pmatrix}$$

S est une matrice de rang 1, donc comme elle est diagonalisable, 0 est valeur propre d'ordre 2 de S , et par conséquent, la dernière valeur propre est égale à $\text{tr}(S) = (\zeta' \circ g)^2(u, v, w)$. De plus, $S \nabla h = (\zeta' \circ g)^2 \cdot \nabla h$ donc ∇h est un vecteur propre pour la valeur propre $(\zeta' \circ g)^2(u, v, w)$.

De même, pour toute fonction à symétrie sphérique autour d'un point (a, b, c) , le tenseur de structure possède deux valeurs propres nulles et une strictement positive dont le sous-espace propre associé est dirigé selon son gradient.

Il est à noter que de plus, la droite orientée par $\nabla h(u, v, w)$ et passant par (u, v, w) passe aussi par le point (a, b, c) . Ainsi, pour tout point de $\partial\Omega$, le centre de symétrie appartient à une droite particulière déterminable à partir du tenseur de structure.

On a l'implication suivante grâce au théorème de Green :

Si Ω est une boule de centre O , alors pour tout point $x \in \partial\Omega$, la droite passant par x et orientée selon un vecteur propre de la plus grande valeur propre de $S_w(x)$ passe par O (Figure 5).

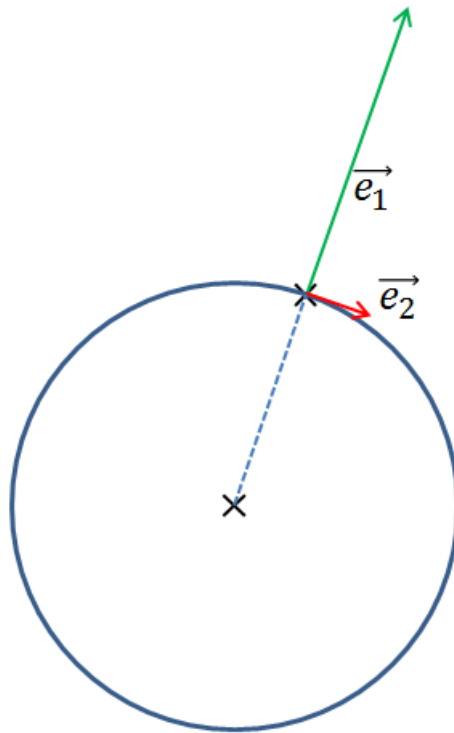


Figure 5 : Dans le cadre d'une sphère, le vecteur propre associé à la plus grande valeur propre du tenseur de structure est dirigé suivant le rayon (\vec{e}_1). Les deux autres vecteurs propres sont associés à des valeurs propres nulles.

On a donc une condition nécessaire pour tester si un ensemble correspond bien à une boule. Si deux des valeurs propres de $S_w(x)$ sont toujours nulles, on a même une condition suffisante, car si Ω n'est pas une boule de centre O , alors il existe deux points de $\partial\Omega$ à distance différente de O . Dans le cas d'une boule, le gradient est totalement dirigé vers le centre. Or, si deux points sont reliés par une courbe différentiable et sont à distance différentes de O , alors le gradient n'est pas toujours identiquement dirigé vers O , et donc soit il existe une deuxième valeur propre non nulle, soit le vecteur propre n'est pas toujours dirigé vers O .

Cependant, on évitera de procéder au calcul d'une droite pour chaque point puis à des calculs d'intersections entre toutes ces droites, ce qui serait finalement assez lourd en calculs. On préférera donc associer directement un point plutôt qu'une droite à chaque pixel considéré.

A chaque point de \mathbb{R}^3 , on peut donc déjà associer une droite comportant le centre de la boule (Figure 6). La connaissance du rayon et d'un sens sur la droite suffit alors à

caractériser ce centre. Le calcul de la courbure permet donc, dans ce cas, de résoudre le problème.

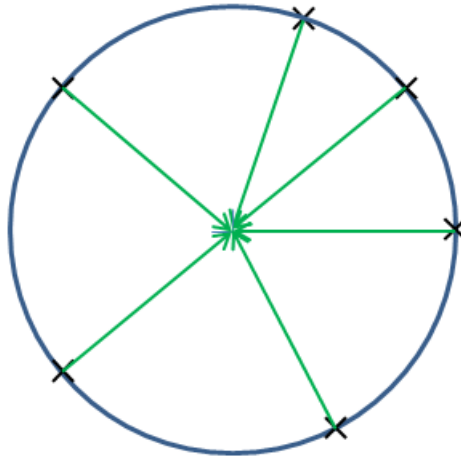


Figure 6 : Pour chaque point d'une sphère, le calcul de la direction principale du gradient et du rayon de courbure donne un vecteur pointant sur le centre de la sphère.

Les rayons de courbures principaux d'une surface paramétrée sont les deux valeurs propres de l'endomorphisme de Weingarten (Gray, et al., 2006). Cet endomorphisme est défini sur une surface paramétrée de \mathbb{R}^4 de la façon suivante :

Soit une surface paramétrée $\mathbf{p} : (u, v, w) \in D \mapsto \mathbf{p}(u, v, w) = (x, y, z, \xi) \in \mathbb{R}^4$ régulière et C^∞ . Pour tout $(u, v, w) \in D$, on note $T = \text{Vect}\left(\frac{\partial \mathbf{p}}{\partial u}, \frac{\partial \mathbf{p}}{\partial v}, \frac{\partial \mathbf{p}}{\partial w}\right)(u, v, w)$ le sous-espace affine tangent à \mathbf{p} en (u, v, w) . De plus, on note \mathbf{n} le vecteur unitaire normal à \mathbf{p} défini par $\mathbf{n} = \frac{1}{\left\| \frac{\partial \mathbf{p}}{\partial u} \wedge \frac{\partial \mathbf{p}}{\partial v} \wedge \frac{\partial \mathbf{p}}{\partial w} \right\| (u, v, w)} \left(\frac{\partial \mathbf{p}}{\partial u} \wedge \frac{\partial \mathbf{p}}{\partial v} \wedge \frac{\partial \mathbf{p}}{\partial w} \right) (u, v, w)$. L'endomorphisme de Weingarten de \mathbf{p} en (u, v, w) est l'endomorphisme \mathcal{W}_p sur T vérifiant :

$$\mathcal{W}_p \left[\left(a \frac{\partial \mathbf{p}}{\partial u} + b \frac{\partial \mathbf{p}}{\partial v} + c \frac{\partial \mathbf{p}}{\partial w} \right) (u, v, w) \right] = \left(-a \frac{\partial \mathbf{n}}{\partial u} - b \frac{\partial \mathbf{n}}{\partial v} - c \frac{\partial \mathbf{n}}{\partial w} \right) (u, v, w)$$

Sur une image J en 3D, qui peut être considérée comme une hypersurface de \mathbb{R}^4 paramétrée par $\mathbf{M} : (x, y, z) \in D \mapsto \mathbf{M}(x, y, z) = (x, y, z, J(x, y, z)) \in \mathbb{R}^4$, des méthodes numériques de calcul des courbures principales sont proposées par Monga et Benayoun (Monga & Benayoun, 1995), menant finalement au calcul suivant :

On note $J_i = \frac{\partial J}{\partial i}$ et $J_{ij} = \frac{\partial^2 J}{\partial i \partial j}$, H est le hessien de J . Alors les courbures principales sont données par :

$$K_{\pm} = -\frac{1}{2} \left(\mathbf{a}^t H \mathbf{a} + \mathbf{b}^t H \mathbf{b} \pm \sqrt{(\mathbf{a}^t H \mathbf{a} - \mathbf{b}^t H \mathbf{b})^2 + 4(\mathbf{a}^t H \mathbf{b})^2} \right)$$

avec

$$\mathbf{a} = \frac{1}{J_x^2(J_y^2 + J_z^2) + 4J_y^2 + J_z^2} \begin{pmatrix} 2J_y J_z \\ -J_x J_z \\ -J_x J_y \end{pmatrix} \text{ et } \mathbf{b} = \mathbf{a} \wedge \nabla J$$

On obtient alors les deux rayons de courbure par $R_{\pm} = \frac{1}{|K_{\pm}|}$.

Dans le cadre d'une sphère, on a $K_+ = K_- = -\frac{1}{2}(\mathbf{a}^t H \mathbf{a} + \mathbf{b}^t H \mathbf{b})$

Ainsi, la connaissance locale d'une sphère permet de retrouver, à partir d'un seul point, le centre et le rayon de celle-ci sous la forme du vecteur $K\mathbf{e}$, K étant le plus grand des deux rayons de courbure exprimé auparavant et \mathbf{e} un vecteur propre unitaire associé à la plus grande valeur propre du tenseur de structure.

Dans un cadre numérique, avec une image discrétisée et des structures imparfaites, creuses, bruitées ou déformées, on se donnera plutôt un accumulateur. En effet, à chaque point x , on pourra associer un couple $(o(x), r(x))$ où $o(x)$ correspond au centre calculé pour le point x et $r(x)$ au rayon calculé. Si la majorité des couples sont finalement assez proches, on pourra considérer la présence d'une sphère sur l'image.

Il est à noter aussi que tout l'intérêt du tenseur de structure pour la recherche de sphères se révèle dans un cadre numérique. En effet, comme cela a été calculé dans le cadre d'une sphère parfaite, une unique valeur propre est non nulle, et le gradient est un vecteur propre associé. Alors quel est l'avantage comparatif du tenseur de structure sur un calcul de gradient simple ? Le principal atout du tenseur de structure est le choix d'un seul espace propre. En effet, dans un cadre numérique, les erreurs d'arrondi peuvent causer des variations de gradient qui se répercutent sur l'orientation de celui-ci. Le tenseur de structure minimise ces erreurs en ne considérant que **la direction principale**. De même, cet avantage permet aussi de suivre des structures légèrement déformées ou bruitées bien plus fidèlement qu'avec un gradient simple.

1.2.4.3. Détection de cylindres via le tenseur de structure

La méthode de détection et reconnaissance de cylindres via le tenseur de structure est assez proche de celle utilisée pour les sphères (Mulat, et al., 2007). Dans le domaine de l'imagerie en 3D, le cylindre est une structure très courante, qu'il soit présent sous forme de fibres dans des matériaux ou sous forme de vaisseaux sanguins sur des images médicales par exemple. La détection de cylindres est par conséquent un domaine assez richement étudié (Mulat, et al., 2007) (Krause, et al., 2010) (Sato, et al., 1998) (Koller, et al., 1995) (Hernandez Hoyos, et al., 2000) (Verdonck, et al., 1996).

Par définition, un cylindre Ω (dans \mathbb{R}^3) de rayon r s'appuyant sur une courbe S est l'ensemble des points x de \mathbb{R}^3 vérifiant $d(x, S) \leq r$. On constatera que la détection de boule s'effectuait via la recherche du point central O . De la même manière, la reconnaissance d'un cylindre se fera par la recherche de la courbe S , que l'on supposera C^∞ . De fait, un point quelconque x de $\partial\Omega$ est à distance r de son projeté $p(x)$ sur S , et localement en x , dans le plan orthogonal à S en $p(x)$, $\partial\Omega$ se comporte comme un cercle de centre $p(x)$ et de rayon r . Dans le cas où le vecteur tangent à S varie peu, on peut déduire des informations à partir des éléments propres du tenseur de structure en x (Mulat, et al., 2007) :

- Le vecteur propre e_1 associé à la plus grande valeur propre du tenseur est un vecteur directeur de la droite issue de x et passant en $p(x)$ (Figure 7).
- Le vecteur propre e_3 associé à la plus petite valeur propre du tenseur est colinéaire au vecteur tangent de S en $p(x)$.

De la même manière que pour la détection de sphères, le rayon de courbure principal permet alors de déterminer $p(x)$. On a alors associé, à un point x de $\partial\Omega$, son projeté sur S et la tangente à S en ce point.

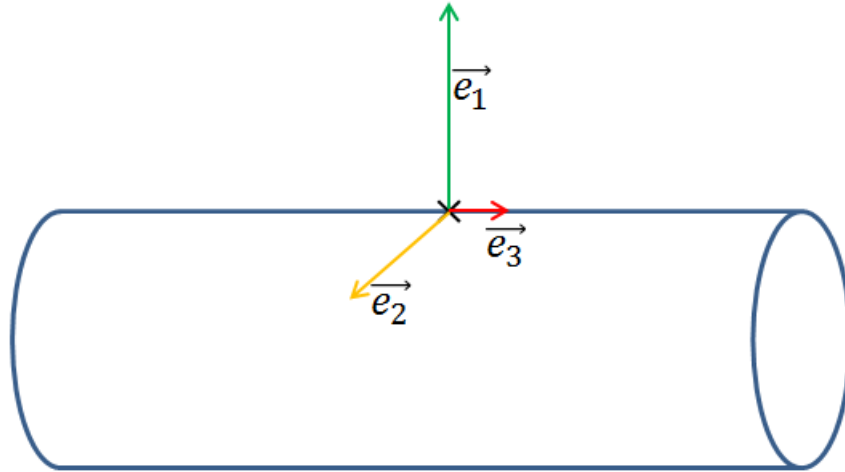


Figure 7 : Sur un cylindre, le vecteur propre associé à la plus grande valeur propre du tenseur de structure est orienté selon le rayon du cercle intersection du cylindre et du plan orthogonal. Le vecteur propre associé à la plus petite valeur propre est pour sa part parallèle à la tangente à la génératrice

Dans le cas où le cylindre ne se recoupe pas et où le vecteur tangent à S varie peu localement, alors pour chaque point p de S , l'intersection de $\partial\Omega$ avec le plan orthogonal à S en p est un cercle de rayon r . Tous les points de ce cercle, via le calcul du tenseur de structure et du rayon de courbure, permettent de retrouver p (Figure 8). Ainsi, dans un cadre continu, le tenseur de structure permet de caractériser entièrement S .

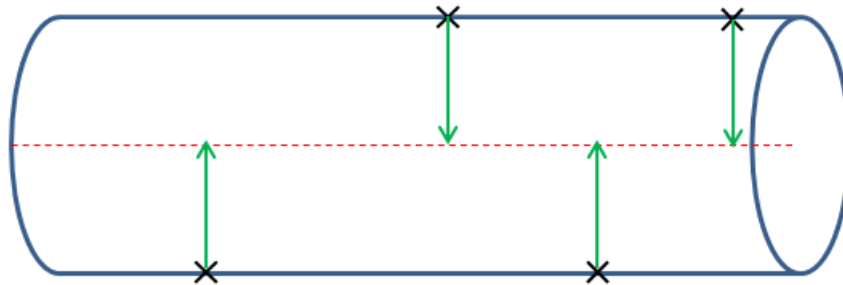


Figure 8 : Sur un cylindre, le vecteur propre associé à la plus grande valeur propre du tenseur de structure et le rayon de courbure donnent un vecteur pointant sur la génératrice.

Dans un cadre réel, avec un cylindre non totalement parfait et sur une image voxelisée, on obtient une série de doublets points-vecteurs, à partir desquels l'axe S doit être retrouvé. Si le cylindre est assez peu déformé, alors l'ensemble des doublets forme une unique composante connexe, proche de S à laquelle un flou aurait été appliqué. Une méthode proposée (Mulat, et al., 2007) pour suivre S voxel par voxel est d'effectuer un calcul de point moyen sur les plans orthogonaux au vecteur propre e_3 , ce qui contre-carre

l'effet de flou dû à la discrétisation. Il est aussi envisageable de retrouver l'équation analytique de S —si celle-ci existe— via une transformée de Hough par exemple.

1.2.4.4. Autres variantes et Améliorations

Le tenseur de structure en lui-même n'est pas une méthode de reconnaissance de formes dans sa globalité, mais juste une matrice particulière permettant par la suite, via des post-traitements, de retrouver des formes voulues. Ainsi, le champ de recherche est assez ouvert, qu'il concerne le tenseur de structure en lui-même, où d'autres méthodes que celle présentée auparavant (Mulat, et al., 2007) pour passer du tenseur à des formes détectées.

D'un point de vue strictement algorithmique, on remarquera déjà que le tenseur de structure utilise toujours de la mémoire superflue, consistant en une matrice symétrique, produit dyadique du vecteur gradient. Rieger et van Vliet (Rieger & van Vliet, 2003) proposent donc de réduire l'espace nécessaire en représentant l'orientation sous une autre forme quadratique proche. Il est alors nécessaire de conserver les propriétés importantes du tenseur posées par Knutsson (Knutsson, 1989) (Knutsson, 1985), permettant l'application de filtres linéaires comme des calculs de moyenne par exemple, sans produire de résultats aberrants.

1.2.4.5. Récapitulatif

Les méthodes de détection de formes fondées sur les propriétés locales des objets recherchés sont des méthodes intéressantes dans le cadre de structure simples. En effet, ces méthodes s'avèrent être :

- Relativement rapides : le calcul des paramètres de la forme se fait pour chaque point grâce aux propriétés locales de la structure. Le calcul en lui-même ne se fait donc que sur un voisinage de chaque point. De même, la phase de récupération et traitement des résultats issus de chaque point se fait aussi assez simplement, les différents jeux de paramètres rendus étant relativement proches.

- Peu coûteuses en mémoire : en effet, les jeux de paramètres étant contraints, l'espace de paramètre n'est pas nécessairement immense.
- Parallélisables : de fait, la méthode ne faisant intervenir que des voisinages, son découpage est assez aisé.
- Assez robustes aux déformations : toute déformation légère de la forme ne se traduit que par un étalement léger des jeux de paramètres. Le calcul des éléments propres permet de corriger des effets de déformation.
- De même, assez robustes à certain types de bruits, en particulier le bruit gaussien. Tant que le bruit ne modifie pas violemment les bordures des structures, celui-ci est très bien traité par les méthodes issues du tenseur de structure.
- Des méthodes agissant sur tous types d'images et pas uniquement des images binaires.

En contrepartie, cette méthode n'est pas utilisable sans précaution dans tous les cas.

On notera entre autres :

- Besoin de structures « séparables », c'est-à-dire que les différentes structures d'une image ne doivent pas trop s'interpénétrer ou se toucher. Dans ce cas, les surfaces en contact risquent fortement de donner des résultats aberrants.
- Compromis entre rapidité et précision. En effet, la taille du voisinage est relativement importante (et donc le temps de calcul l'est aussi de fait) pour obtenir une estimation correcte des paramètres. Dans le cas contraire, l'ensemble des jeux de paramètres possibles est rapidement très étalé. De plus, l'augmentation de la taille du voisinage nuit aussi au parallélisme.
- Peu robuste face à certains bruits. Dans le cas de bruits forts sur les bords des structures, il est parfois impossible d'obtenir un jeu de paramètres privilégié tellement l'étalement des résultats est anarchique.
- Segmentation d'image implicite. Le choix du $\partial\Omega$ à vérifier nécessite donc un critère de détermination. Dans le cadre le plus simple, on prend des surfaces d'isovaleurs, mais celles-ci ne sont pas toujours représentatives et le choix de l'isovaleur porte à réfléchir.

La détection de formes grâce à la connaissance de variables locales telles que le gradient ou le tenseur de structure présente de nombreux avantages pour peu que l'on sache quoi chercher et que l'image vérifie certaines conditions relativement peu contraignantes. Une implémentation de cette méthode est donc complètement envisageable dans notre cas.

1.3. Résultats significatifs obtenus

1.3.1. Matériaux fibreux

Les méthodes développées dans cette thèse, fondées principalement sur des études géométriques en association avec les méthodes existantes de la littérature, permettent d'obtenir des résultats intéressants sur les différents types d'images étudiés. Sur les matériaux fibreux, la problématique consiste en une détection de fibres, suivi d'une phase de classification en orientation et en localisation de ces fibres afin de former des ensembles cohérents, correspondant à des fils.

Dans un premier temps, on s'intéresse aux résultats obtenus sur la détection de fibres. La reconnaissance de formes tubulaires n'est pas à proprement parler un domaine nouveau dans la reconnaissance de formes, mais la grande diversité de structures possibles et de types d'images rend impossible l'existence d'une méthode généraliste et performante pour toute image. Dans la littérature, on note que la détection de formes tubulaires sur des images 3D est souvent traitée :

- Soit de manière très générale : les algorithmes et méthodes décrits sont alors le plus souvent assez fonctionnels, mais, n'utilisant aucun des a priori que l'on peut avoir sur les images, ils sont aussi assez lents. En effet, ces méthodes, pour donner des résultats performants, sont le plus souvent fondées sur des principes très calculatoires, ce qui a tendance à faire exploser le temps de calcul. Les méthodes utilisant la Transformée de Hough (Partie 1.2.3) ou le Tenseur de Structure (Partie 1.2.4) sont de celles-ci. Leurs caractères généralistes en font des méthodes très étudiées, présentant ainsi des nombreuses variantes plus ou moins adaptées à certains problèmes particuliers, avec des gains de temps parfois conséquents. Cependant, les gains maximaux atteignables restent limités par le principe même de fonctionnement de ces algorithmes.
- Soit de manière très spécialisée : des méthodes sont aussi présentées pour résoudre un type de problème particulier, c'est-à-dire pour rechercher des structures sur des images avec un a priori très fort. Ces méthodes sont souvent conçues pour des applications médicales. Elles sont par conséquent utilisables sur

des images de petite taille, mais fonctionnent bien sur des cylindres de rayons divers et s'appuyant sur des courbes et sur des images bruitées.

A l'inverse, les algorithmes construits pour des images de matériaux issus de tomographies X sont relativement peu nombreux. La nature inerte de ces matériaux permet d'obtenir des images souvent bien plus nettes, mais d'une taille bien plus conséquente.

Les résultats significatifs obtenus dans ce cas consistent donc à la conception d'une méthode de détection de fibres sur des images de matériaux composite fibreux (plus particulièrement carbone/résine), d'une complexité assez faible pour rendre la méthode acceptable sur des images de très grande taille, et d'une précision suffisante.

L'algorithme présenté ici est destiné à une utilisation sur des images de taille suffisantes (au minimum 100 voxels au cube), présentant des fibres quasi-rectilignes, pouvant se toucher mais non s'intersecter. Malgré ces restrictions, la méthode développée est utilisable pour différents types de matériaux fibreux, et ne se limite pas aux composites carbone-résine étudiés ici.

On propose donc dans ce document un algorithme de détection de fibres droites fonctionnant en associant entre eux les intersections de fibres sur des plans choisis. Ces associations se font en suivant un critère d'appartenance des droites les reliant à des phases précises de l'image.

On peut affirmer de cet algorithme de détection de fibres droites qu'il est :

- De complexité linéaire avec la taille de l'image dans le cas d'images de matériaux composites où les fibres sont présentes sous forme de fils, i.e. sont réunies en structure cohérente en espace et en orientation. Dans le cas de composites quelconques où les fibres sont séparables, l'algorithme est aussi en complexité linéaire.
- De complexité en $O\left(N^{\frac{5}{3}}\right)$ (i.e. meilleur que la complexité quadratique), avec N le nombre de voxels de l'image, dans le cas général.

Du point de vue de la complexité temporelle, et dans le cadre de l'utilisation sur les composites fibreux carbone-résine, l'algorithme construit ici est le meilleur possible, le

comportement linéaire étant évidemment une limite infranchissable pour un algorithme devant inspecter l'intérieur de l'image. La discussion sur la constante de temps associée reste ouverte. Celle-ci ne dépend pas que de la méthode théorique seule, mais aussi de son implémentation informatique, ce qui rend toute comparaison simple des constantes de temps difficile, celles-ci pouvant varier fortement selon la plateforme de développement.

Taille	100 ³	200 ³	400 ³	800 ³
Temps de calcul (s)	0,18	1,3	29	1900
Nombre de droites	59	193	1016	4938

Tableau 1 : Exemples de temps de calcul obtenus sur des blocs issus d'une image donnée. Les temps de calculs obtenus sur les plus petites tailles peuvent rendre envisageable une exploitation en temps réel de l'algorithme. Sur les plus grosses images, bien que le contexte « temps réel » soit plus difficile, il est toujours possible d'utiliser l'algorithme en un temps très raisonnable.

En ce qui concerne les résultats de l'algorithme sur les images de matériaux composites carbone-résine, ceux-ci se révèlent être particulièrement bons (Figure 9 et Figure 10), avec des taux de détection approchant les 95 % sur des images réelles (mesuré manuellement en l'absence de moyens fiables) et des images virtuelles (mesuré automatiquement par ordinateur). La méthode utilisée présente plusieurs avantages :

- Méthode itérative : les éléments les plus pertinents sont d'abord considérés, ce qui permet de faire varier la qualité de détection de l'algorithme via un seuil de pertinence.
- Apprentissage en place : les éléments incontestables de l'image permettent ensuite de contraindre la détection des éléments moins pertinents. De fait, on diminue ainsi les sources d'incertitude, ce qui permet d'obtenir des bons résultats même avec des seuils de pertinence faibles.

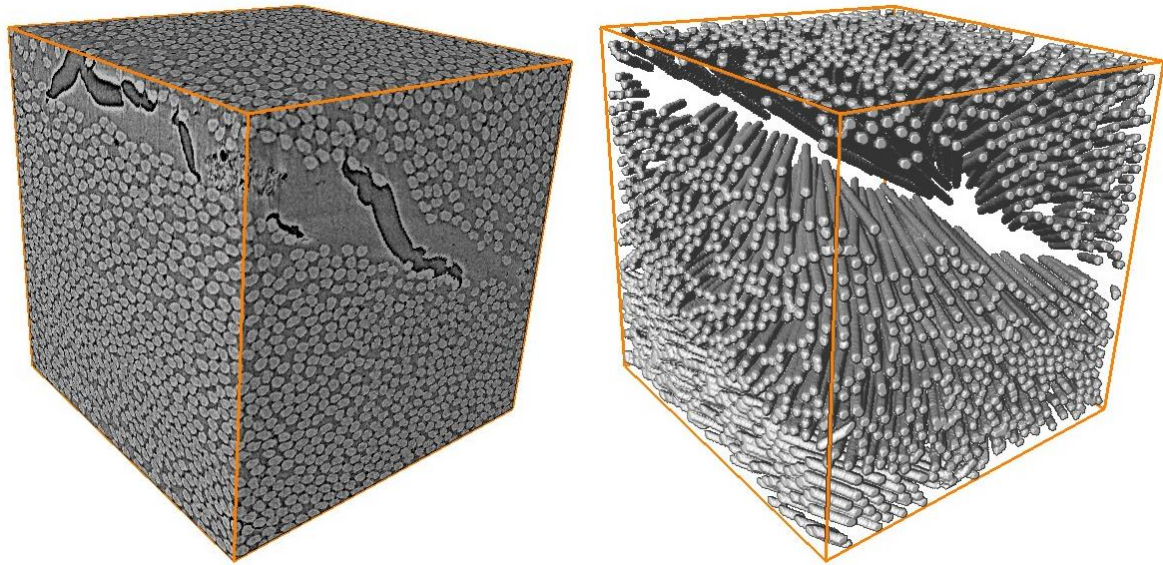


Figure 9 : Image originale d'un matériau fibreux (à gauche) de dimension 400^3 et résultat de la méthode de détection de droites (à droite).

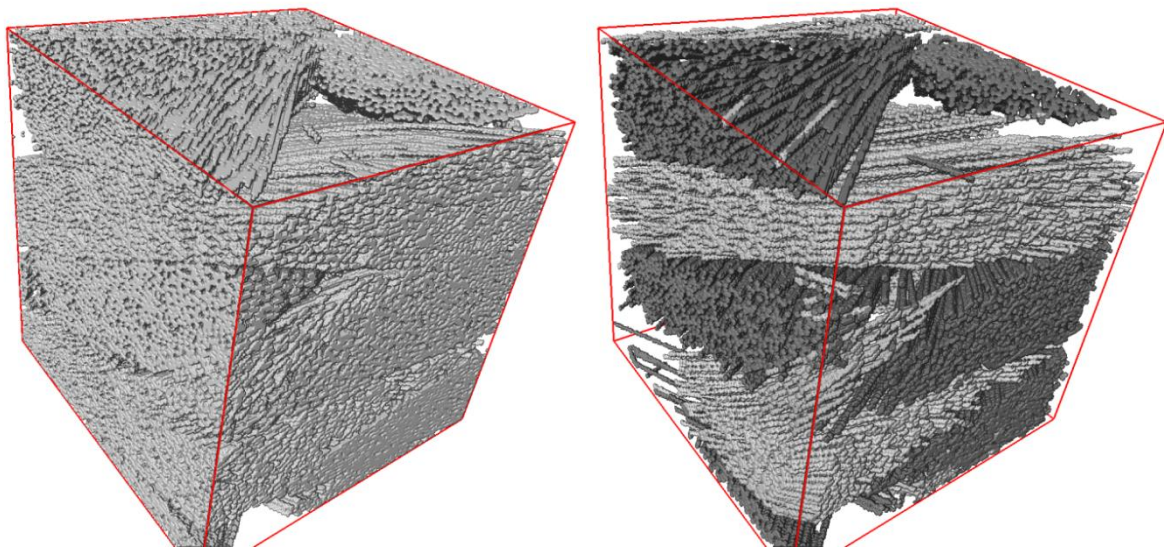


Figure 10 : Image originale d'un matériau fibreux (à gauche) de dimension 400^3 avec une résolution deux fois plus faible que pour la Figure 9. Résultat de la méthode de détection de droites (à droite).

Suite à la détection de fibres, une méthode de classification performante permettant de réunir les groupes de fibres cohérents entre eux est développée. Il est à noter que dans le cadre de la thèse, la détection et la classification sont conçus de manière à être utilisés ensemble. Ainsi, bien que chacune des deux méthodes puisse se voir attribuer une performance individuelle, l'union de celles-ci permet d'obtenir des résultats significativement meilleurs.

L'algorithme de classification permet d'obtenir de manière fiable des regroupements de fibres grâce à leurs orientations et leurs positions. Sa complexité est en $O(S \ln S)$ avec S le nombre de fibres, qui est très faible devant la taille de l'image originale, d'une part, et se comporte en $O\left(N^{\frac{2}{3}}\right)$. Ainsi, cet algorithme, bien que non-linéaire, se comporte finalement en $O\left(N^{\frac{2}{3}} \ln N\right)$, c'est-à-dire de manière sous-linéaire avec la taille de l'image originale, qui plus est sur un nombre de points très faible devant la taille de l'image. De fait, la phase de classification est rapide et négligeable en temps devant la détection de fibres.

L'algorithme de classification est, du point de vue des résultats, réellement performant (Figure 11), et permet en plus d'apporter une validation et éventuellement une correction de la sortie de l'algorithme de détection de fibres. En effet, la méthode consistant à regrouper les fibres, elle est aussi capable de reconnaître des éventuelles données incohérentes et, le cas échéant, de les supprimer simplement. L'algorithme de classification nous fournit une validation et une sécurité supplémentaire sur la méthode précédente, qu'il est alors possible d'exploiter de manière plus « limite », c'est-à-dire en acceptant des droites litigieuses. Grâce à cet algorithme, il est possible d'obtenir des taux de détections approchant les 98 % sur l'algorithme de reconnaissance de fibres (en absence de moyens fiables pour la comparaison des résultats obtenus avec les images de départ, ces valeurs sont obtenues manuellement par comparaison des bords des images).

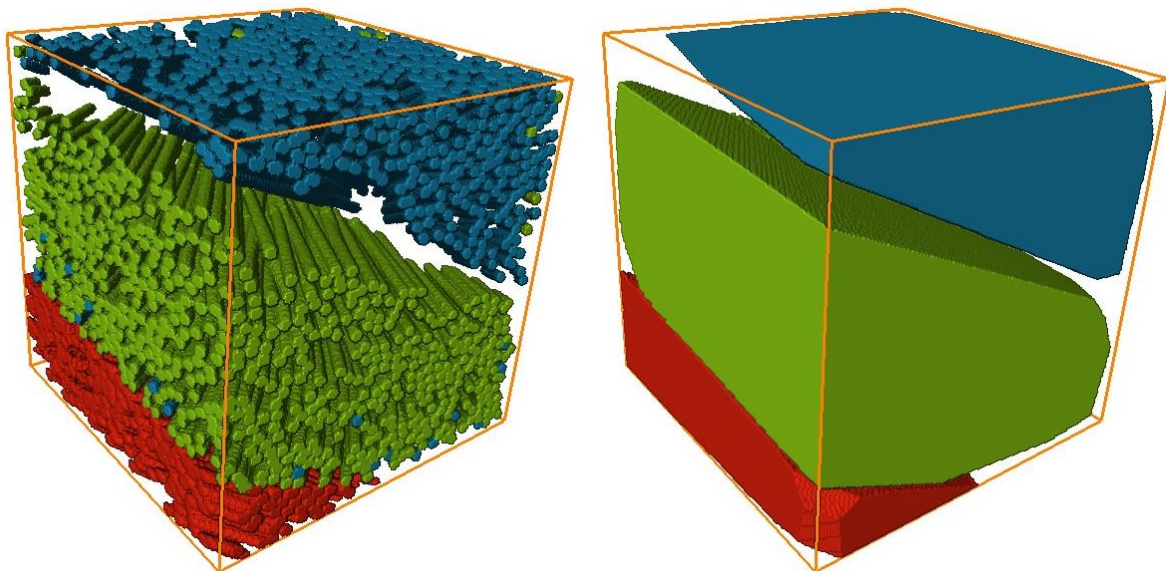


Figure 11 : Résultat de la classification à partir de la Figure 9 (à gauche). A droite, suppression des résultats incohérents et représentation de la structure filaire.

Finalement, on propose une méthode de découpage permettant, dans des cadres où le comportement temporel des algorithmes précédents n'est pas satisfaisant, de retrouver des complexités linéaires avec faible constante de temps.

L'ensemble d'algorithmes proposé permet d'effectuer l'ensemble de la chaîne de modélisation du matériau composite fibreux, de la détection de la fibre à la représentation à plus grande échelle des fils comme une entité homogène unique. Cet ensemble d'algorithmes, conçus pour fonctionner en association, donne des résultats particulièrement intéressants. De plus, chaque algorithme reste utilisable de manière indépendante.

1.3.2. Mousses et matériaux assimilés

Sur les matériaux de types mousses et assimilés, la problématique consiste en une détection de structures sphériques et en leur caractérisation par leur position et rayon, afin de pouvoir décrire le matériau via l'homogénéité de la répartition spatiale des sphères et la présence relative des différents rayons.

La littérature existante sur la détection de structures sphériques est moins fournie que sur la détection de structures fibreuses, en raison probablement d'une utilité moindre dans les domaines de l'imagerie médicale. On notera tout de même l'existence de méthodes étendues à partir d'algorithmes 2D permettant de détecter des sphères dans des cas assez généraux. Intuitivement, les sphères étant des structures présentant beaucoup moins de variations que les fibres (seul le rayon varie), peu de méthodes ciblées sur des cas d'utilisation particuliers existent. Toutefois, bien que d'un point de vue théorique, les sphères ne présentent que peu de variations, il est nécessaire, dans certains cas, d'avoir à détecter des sphères déformées, incomplètes ou incluses dans d'autres structures.

L'algorithme construit durant cette thèse permet d'obtenir des résultats raisonnables rapidement dans un cadre toujours assez généraliste. La méthode fonctionne avec quelques restrictions.

L'algorithme présenté ici est destiné à une utilisation sur des images de tailles diverses, présentant des sphères creuses, qui peuvent être incomplètes ou légèrement déformées, de taille raisonnable (assez grandes pour que l'intérieur de ces sphères soit vide,

et dont le rayon n'est pas non plus trop important), pouvant se toucher mais non s'interpénétrer. De plus, les sphères doivent être totalement vides, et ne pas présenter de structures en leur sein. Ces restrictions, qui semblent fortes de prime abord, sont en fait suffisamment lâches pour que de nombreux matériaux puissent être étudiés par cette méthode.

On présente un algorithme de détection de sphères fondé sur la recherche de maximums de carte de distance, étant donné que l'ensemble des centres des sphères est en surjection avec les maximums locaux de la fonction de distance. Cet algorithme de détection de sphères est de complexité linéaire avec la taille de l'image, avec une constante de temps relativement faible, directement dépendante des rayons des sphères à détecter. Cet algorithme est l'un des plus rapides possible pour la détection, le comportement linéaire étant toujours un seuil de complexité minimal, l'image entière devant de nouveau être étudiée. En ce qui concerne la constante de temps, celle-ci est toujours discutable en raison de l'aspect implémentation. Cependant, on constate que la méthode développée est beaucoup moins calculatoire que les algorithmes fondés sur le Tenseur de Structure, ce qui permet d'obtenir des résultats plus rapidement, en particulier pour des sphères de faible rayon (la constante de temps de la méthode développée variant avec le rayon des sphères à retrouver).

Taille	100^3	200^3	400^3	800^3
Temps de calcul (s)	0,12	1,1	21	850

Tableau 2 : Exemple de temps de calculs obtenus pour une mousse de polyuréthane donnée.

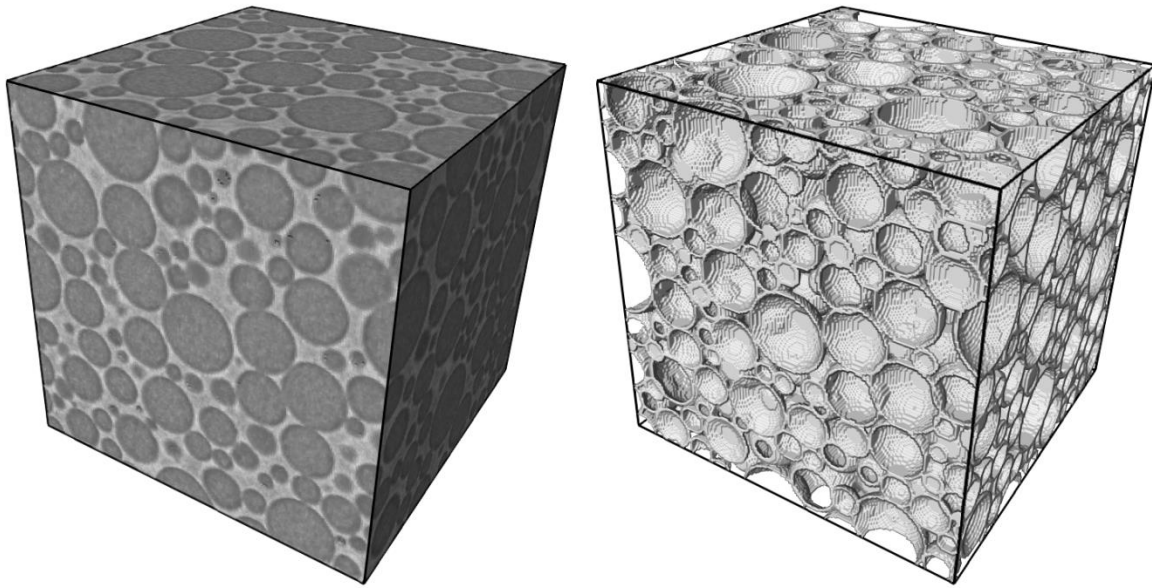


Figure 12 : Image de mousse de polyuréthane (à gauche) et résultat de la détection de sphères (à droite).

Les résultats sur les images de mousses sont variables selon les matériaux étudiés :

- Sur des images nettes assez simples de matériaux présentant des structures quasi-sphériques de taille moyenne, comme par exemple des images de mousse de polyuréthane (Figure 12), la méthode se révèle très rapide et très performante. On obtient des taux de détection avoisinant les 99 % sur ce type de matériau.
- Sur des images denses présentant des structures moins sphériques et de tailles diverses comme des images de mousse de silice (Figure 13), la méthode se révèle toujours plus rapide que les méthodes concurrentes comme le tenseur de structure. Cependant, les résultats ne sont pas de très bonne qualité. La détection des sphères est bien effectuée, mais les résultats sont pollués par la présence de « faux-positifs », c'est-à-dire des sphères inexistantes qui sont détectées aux interstices entre les sphères réelles. Sur ce type de matériau, si on cherche à ne pas accepter de fausses sphères, le taux de détection est aux alentours de 70 %.

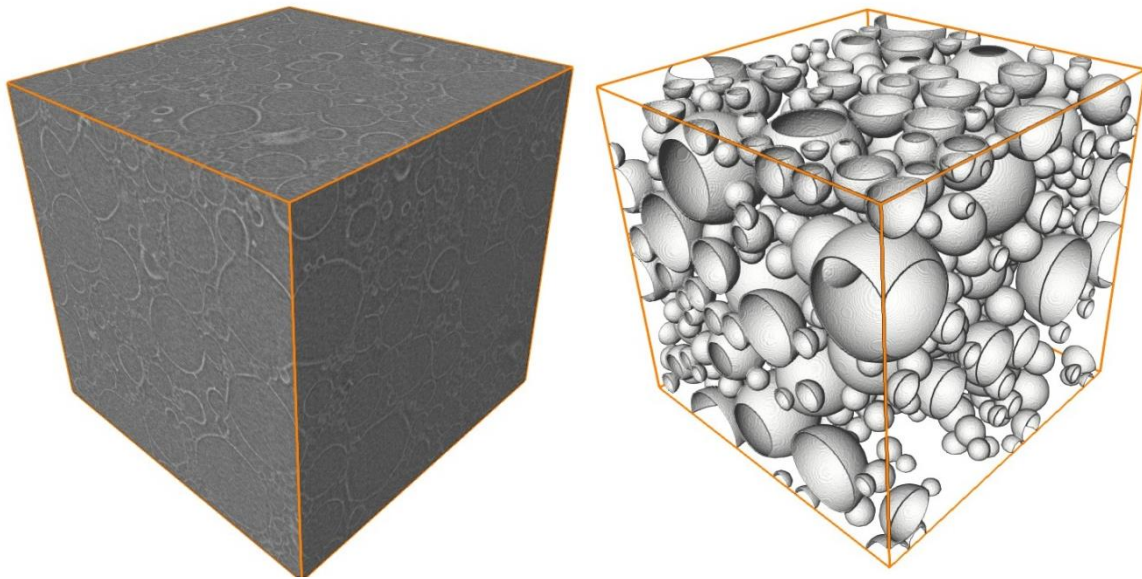


Figure 13 : Image de mousse de silice (à gauche) et résultat de l'algorithme de détection de sphères (à droite).

Des méthodes de vérification des sphères sont donc proposées. Ces méthodes, d'une complexité linéaire, permettent finalement de trier les sphères entre faux-positifs et vraies structures, et complètent donc la détection de manière à obtenir un algorithme de détection de sphères dont les résultats sont semblables à ceux d'autres méthodes, le tout dans des temps plus raisonnables. Sur les images de mousse de silice, on obtient alors des taux de détection proche de 75 %.

2. Recherche de droites : application à des composites Carbone/Résine

2.1. Introduction et propriétés des images utilisées

Nous allons maintenant nous intéresser à des images tomographiques réelles. Ces images représentent un matériau à base de fibres de carbone tressées dans une matrice en résine. La répartition de ces fibres se fait de manière bien particulière :

- Le matériau, à petite échelle, est constitué de fibres de carbone entourées par de la résine.
- A plus grande échelle, ces fibres forment des groupes spatialement cohérents—des fils—qui baignent à nouveau dans une matrice de résine, contenant parfois des pores.
- En remontant encore, les fils s'enchevêtrent et suivent des trajectoires courbes dans le matériau.

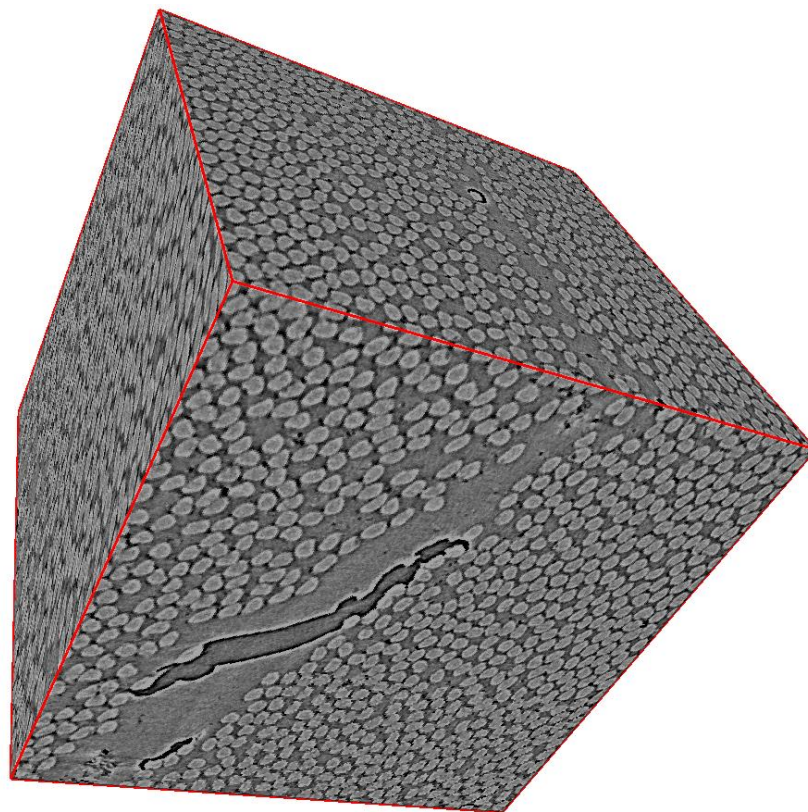


Figure 14 : Image tomographique du matériau carbone/résine considéré. Image de taille 400^3 , avec une résolution de $0,7 \mu\text{m}/\text{vx}$. Les zones claires à l'œil correspondent aux fibres et les zones sombres à la résine. La région entourée de deux halos noirs et blancs est un pore.

Le but de l'étude de ces images consiste, dans un premier temps, en la détermination de l'appartenance de chaque voxel à une des phases, les propriétés d'orientation, s'il y a lieu, de cette phase, et cela à toutes les échelles. C'est-à-dire que, pour chaque voxel de l'image tomographique, on cherchera à savoir :

- S'il appartient à une fibre, à de la résine ou à un pore. S'il appartient à un fil, on déterminera son orientation locale.
- A l'échelle supérieure, s'il appartient à un fil, et le cas échéant, l'orientation locale du fil.

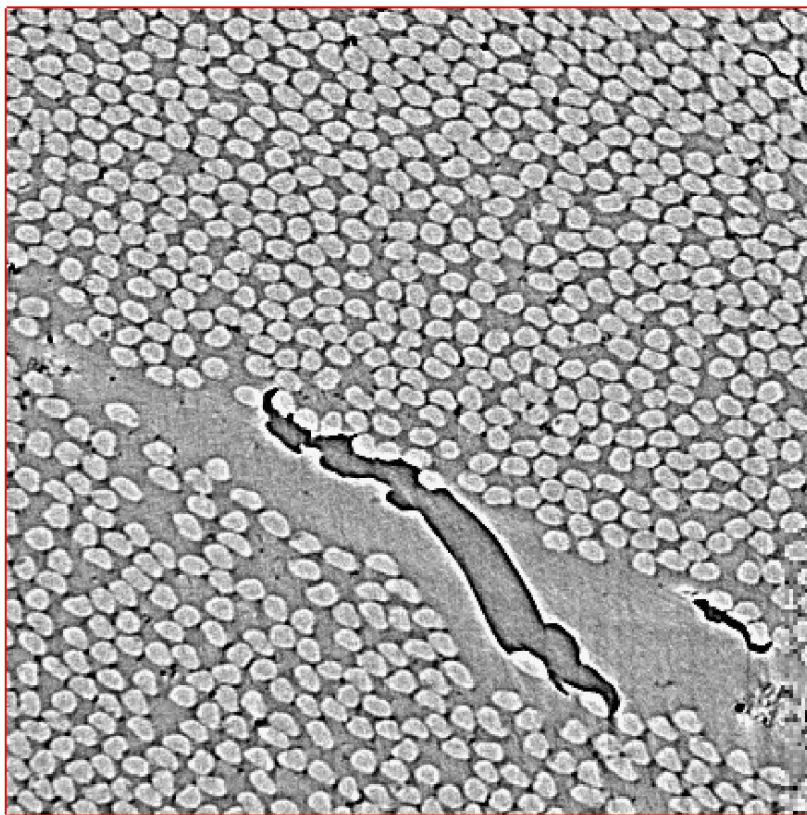


Figure 15 : Coupe de l'image tomographique présentée à la Figure 14.

Les images tomographiques ne donnant que des niveaux de gris, il est évident qu'au départ, aucune de ces informations n'est disponible. De plus, autant l'information d'appartenance à une fibre ou à la résine peuvent potentiellement être trouvées uniquement grâce à la valeur du voxel, autant les informations d'orientation et d'appartenance aux fils sont des informations globales, impossibles à retrouver en observant les éléments un à un.

Les images tomographiques disponibles de ces composites carbone/résine sont à deux résolutions différentes : soit à $0,7 \mu\text{m}/\text{vx}$ (Figure 14 et Figure 15), soit à $1,62 \mu\text{m}/\text{vx}$ (Figure 16 et Figure 17). Les premières sont de taille 2048^3 , et représentent donc un bloc de $1,4 \text{ mm}$ de côté de composite. Les deuxièmes sont de taille $4320 * 4320 * 6800$, ce qui permet donc de représenter un bloc de taille $7 * 7 * 11 \text{ mm}^3$. Ces images sont à valeur dans l'ensemble des `unsigned short`, c'est-à-dire des entiers positifs compris entre 0 et $2^{16} - 1$, stockés sur 2 octets.

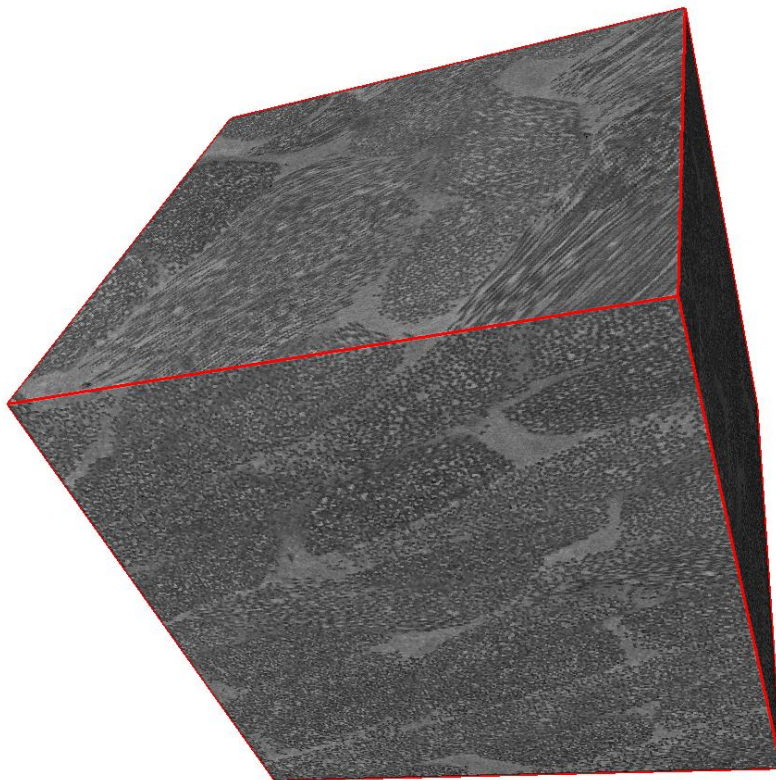


Figure 16 : Autre image tomographique de composite carbone/résine, cette fois de taille 1000^3 avec une résolution de $1,62 \mu\text{m}/\text{vx}$. On observe à ce niveau les fils, ensembles cohérents de fibres.

Sur ces images, la densité de fibres à l'intérieur des fils est particulièrement élevée ($\sim 65 \%$). La séparation entre les fibres n'est pas toujours visible sur les images à haute résolution, et à plus forte raison sur celles à faible résolution. Le diamètre de ces fibres est proche de 10 voxels (respectivement 5 voxels) sur les images fortement (respectivement faiblement) résolues.

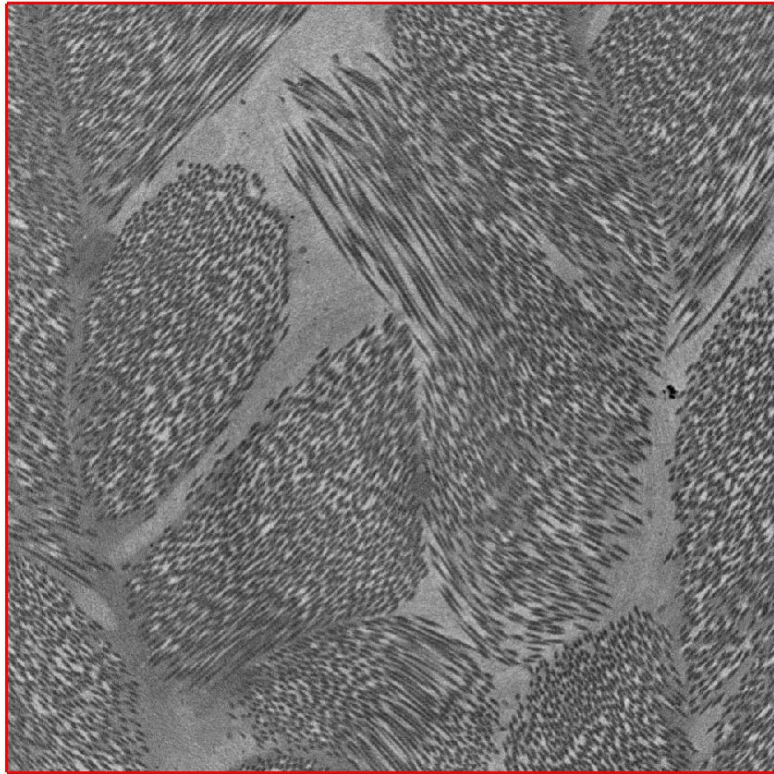


Figure 17 : Coupe de l'image présentée sur la Figure 16.

2.2. Contraintes et facilités accordées par le contexte et les images

La structure géométrique du matériau et les propriétés des images tomographiques obtenues vont bien entendu guider le choix des méthodes utilisées pour leur traitement, en fonction des contraintes et facilités que celles-ci accordent.

Du côté des contraintes, on notera :

1. La taille des images est importante, plus de huit milliards de voxels pour le premier type et plus de cent milliards pour les autres. Cette taille va très fortement limiter le champ des techniques utilisables, celles-ci devant être peu gourmandes en temps comme en mémoire.
2. On pourra travailler sur des sous-images, mais la taille de celle-ci doit être assez grande pour que la longueur des fibres présentes soit significativement plus grande que leur diamètre. Ainsi, une méthode entièrement fondée sur une approche de *divide-and-conquer* ne peut pas s'appliquer ici. On se posera ici une limite de taille de travail de 100^3 , voire 200^3 , afin d'obtenir un facteur au moins 10 entre le diamètre d'une fibre et la taille de l'image.
3. Les fibres représentées sur les images tomographiques sont creuses en niveau de gris, c'est-à-dire que le niveau de gris de leur intérieur est semblable à celui de la résine (Figure 18 et Figure 19). Les méthodes de segmentation les plus communes ne nous permettent donc pas d'obtenir la séparation fibres/résine de manière satisfaisante. De plus, toute segmentation pourra donc potentiellement faire apparaître des trous le long des fibres, voire des séparations nettes entre deux morceaux.
4. La présence des pores et leur proximité en niveau de gris avec les autres phases du matériau empêchent d'effectuer une séparation simple fibres/résine même après la segmentation.
5. La proximité spatiale et les contacts entre les différentes fibres nous empêchent de retrouver directement les fibres par des considérations simples de connexité. Ceci est de plus confirmé par le point 3, qui ajoute aux contacts entre fibres la potentielle rupture de fibres sur la segmentation.

6. La densité de l'image peut amener des cylindres courbes virtuels traversant un fil dense à obtenir des valeurs de densité proches des fibres réelles de l'image. Un algorithme non-adaptatif aura donc beaucoup de chance de détecter de fausses fibres.

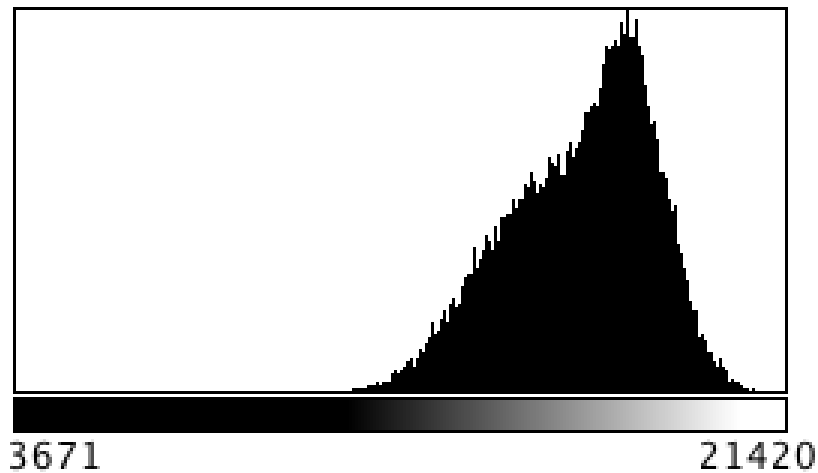


Figure 18 : Histogramme d'une image de composite carbone/résine. Il est possible de deviner la présence de deux pics, mais ceux-ci ne sont manifestement pas résolus.

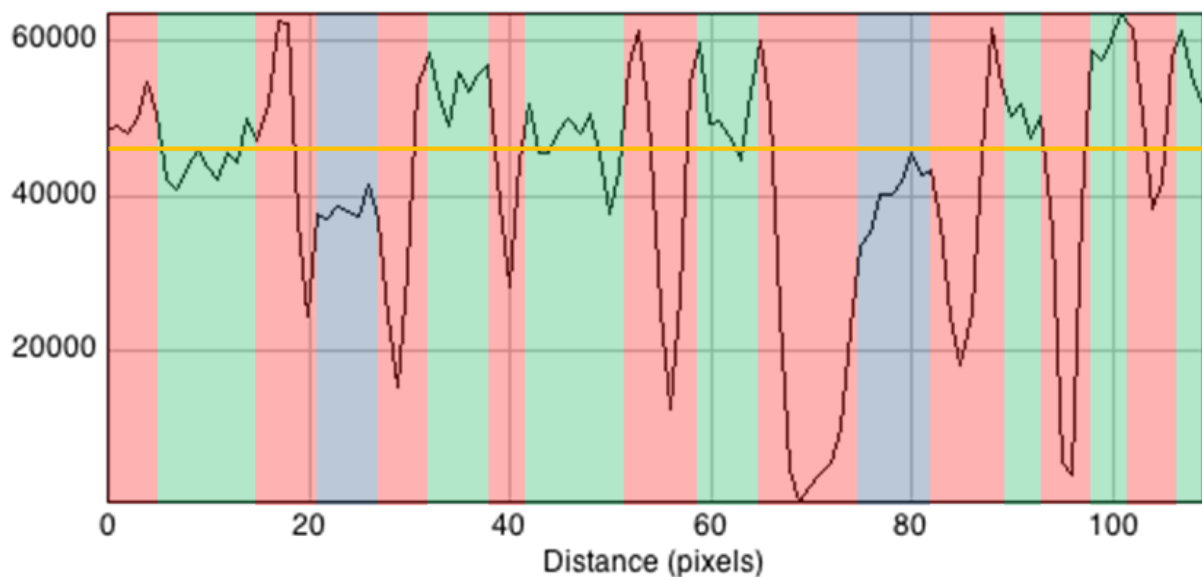


Figure 19 : Profil de niveau de gris d'une image de composite le long d'une droite. Les zones vertes correspondent à l'intérieur d'une fibre, les rouges aux bords des fibres et les bleues à la résine. Comme le montre la ligne orange, des zones de l'intérieur des fibres ont des niveaux de gris parfois semblables à la résine.

Concernant les avantages, on notera :

1. Malgré les limitations, le traitement de l'image sous forme de sous-blocs est possible. Celle-ci améliore de fait les demandes des algorithmes en temps et en espace.
2. Les fibres sont assez fidèlement représentées par des cylindres, c'est-à-dire que leur section par un plan orthogonal sera très proche d'un cercle.
3. Sur des sous-images de taille raisonnable (inférieures à 800^3 sur les images à forte résolution), la courbure des fibres est suffisamment faible pour que l'on puisse les assimiler à des cylindres droits. Cette propriété ouvre la porte à des méthodes très rapides pour la détection.
4. Les fils étant constitués de fibres d'orientations très proches, la direction de celles-ci peut être très fortement contrainte à partir du moment où l'on connaît quelques-unes d'entre elles. Ceci permet en outre de s'opposer au point 6 des contraintes.

A partir de ces quelques caractéristiques du matériau et de l'image, il est donc nécessaire de choisir et développer une méthode de traitement qui nous permettra d'obtenir les différentes propriétés demandées dans la partie 2.1.

2.3. Avantages et Inconvénients d'une méthode géométrique. Comparaison théorique des méthodes.

On constatera dans un premier temps que les caractéristiques demandées sont multiples. De fait, on peut donc diviser l'algorithme que l'on cherche à créer en différentes étapes nécessaires à la réalisation de chacun des résultats :

1. Un algorithme de segmentation, associant à chaque voxel une classe associée au constituant effectivement présent sur celui-ci.
2. Un algorithme de détection de fibres, permettant de connaître la position et l'orientation de chacune d'entre elles.
3. Un algorithme de détection de fils, calculant aussi la localisation (par un calcul d'enveloppe) et l'orientation générale de celui-ci.
4. Eventuellement, un algorithme de fusion de fils et/ou fibres si les calculs sont effectués sur des sous-images.

Deux possibilités s'offrent alors à nous : la première consiste à voir chaque méthode comme un tout indépendant, c'est-à-dire ignorant complètement les résultats des autres (approche simultanée). La seconde consiste en une approche séquentielle, considérant les autres résultats comme autant de connaissances supplémentaires permettant de faciliter ainsi le calcul. Il est évident que la phase 4 est dépendante des phases 2 et/ou 3 : elle est donc exclue de ces considérations.

L'algorithme de segmentation ayant un but local, par opposition aux deux autres, il est maladroit de développer une méthode se servant des résultats des deux autres algorithmes. La reconnaissance de fils, d'un côté, ne représente même pas la différence entre les voxels des fibres et ceux de la résine à l'intérieur d'un fil. L'information portée par les fils n'est donc pas pertinente pour une segmentation. D'un autre côté, le résultat de la recherche de fibres étant clairement dirigé vers une simplification de la représentation du matériau, les fibres y sont représentées comme des structures « parfaites », c'est-à-dire ignorant les potentielles imperfections de celles-ci. De fait, le résultat de la recherche de fibres n'est pas non plus adapté au calcul de la segmentation.

L'algorithme de détection de fibres, peut être obtenu avec trois prérequis différents. En premier lieu, il est tout à fait envisageable de déterminer les fibres uniquement à partir de l'image initiale. La méthode présentée dans la partie 1.2.4, à base de tenseur de structure, est par exemple capable de trouver les fibres sans pré-traitement préalable. D'un autre côté, il est aussi possible, de manière plus évidente, de considérer la segmentation comme une information supplémentaire pour la détermination des fibres. Connaître pour chaque voxel le constituant associé est en effet une information intéressante pour dégager de manière simple des tendances d'orientation et de localisation. Par exemple, un calcul de régions connexe peut renvoyer directement des fibres, dont il ne restera plus qu'à extraire une représentation « parfaite ». Finalement, la connaissance des fils peut être ajoutée aux deux méthodes déjà définies, et ajouter ainsi une contrainte forte sur les orientations des fibres, simplifiant la recherche.

L'algorithme de détection de fils, pour conclure, peut être pensé de trois façons. Encore une fois, il est possible d'utiliser une méthode directe sans prérequis. La connaissance de la segmentation peut être un plus pour la méthode. Mais le meilleur prérequis à cet algorithme est la connaissance des fibres. Il est alors possible de classer directement les fibres selon la position et l'orientation, et ensuite, de déterminer l'orientation comme la moyenne de la classe et l'enveloppe par divers moyens, qu'ils soient issus de la morphologie mathématique ou d'un calcul de triangulation.

L'obtention des fils à partir des fibres étant relativement aisée (et représentant un sens « naturel » de simplification), on ne considèrera que les possibilités faisant ce choix. On passe alors de dix possibilités à seulement deux (Figure 20). De plus, le fonctionnement séquentiel permet de profiter des résultats obtenus à l'étape précédente. Le choix entre les deux possibilités restantes se fera suivant le choix de l'algorithme de recherche de fibres.

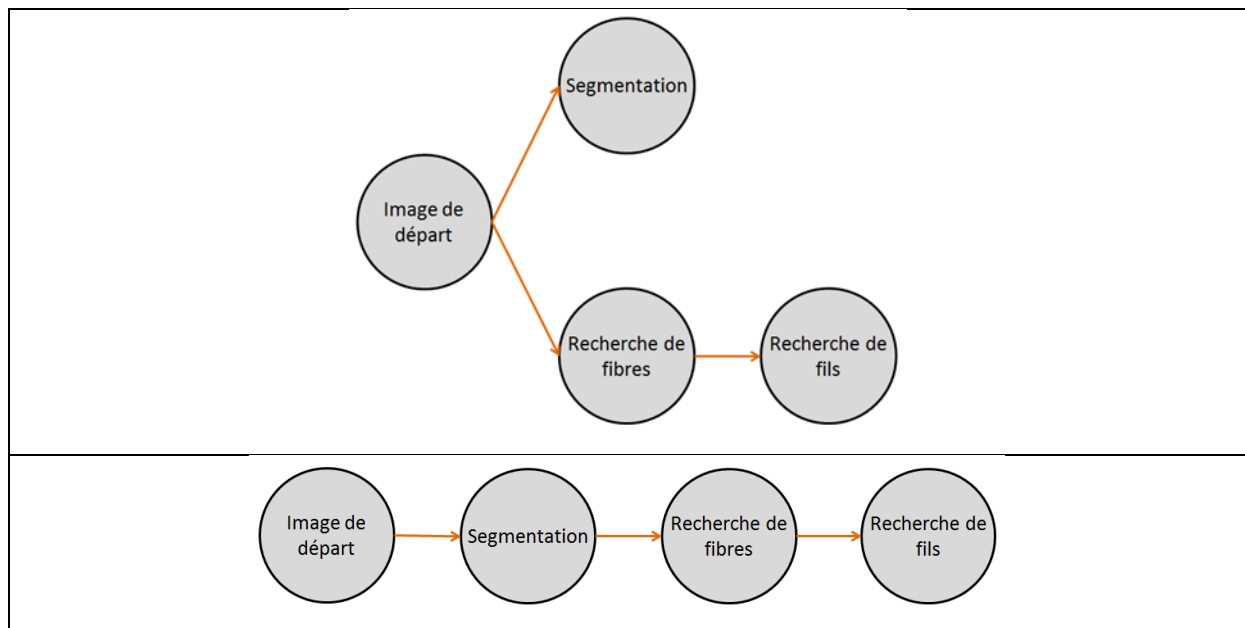


Figure 20 : Les deux fonctionnements envisageables pour la résolution du problème posé.

Commençons par nous intéresser à l’algorithme de segmentation. Son but est d’associer une classe à chaque voxel de l’image, chaque classe représentant un constituant de l’image. Dans un premier temps, on négligera les pores et on ne s’intéressera qu’à séparer les fibres de la résine. On a donc ici un problème de segmentation à deux classes.

L’idée la plus évidente est d’effectuer un seuillage simple sur l’image. Cependant, dans notre cas, cette méthode n’est pas satisfaisante. En effet, comme cela a été rappelé dans les contraintes propres à l’image, le niveau de gris des intérieurs de fibres est semblable à celui de la résine, rendant le seuillage simple inefficace.

On pourrait chercher, avec le seuillage simple, à n’obtenir que les enveloppes des fibres, puis de les remplir, mais la représentation des enveloppes est parfois fragile, c’est-à-dire que le seuillage simple va aussi, à certains endroits—rares mais suffisants pour être gênant—traverser les frontières des fibres. De fait, la distinction entre « intérieur » et « extérieur » de fibre n’est pas aussi évidente.

L’application d’un filtre avant le seuillage pourrait permettre de corriger le problème. Les filtres moyens ne semblent clairement pas de bonnes idées, atténuant les seuils et aggravant la difficulté de segmenter. Les filtres médians, pour leur part, fournissent des résultats tout à fait honorables. Cependant, ces derniers ont parfois tendance à traiter plus

fortement les intervalles entre les fibres que l'intérieur de celles-ci, ce qui provoque ensuite des fusions de fibres supplémentaires.

On opte finalement pour une méthode de segmentation supervisée par champs de Markov (partie 2.4.1). L'initialisation des fonctions de distributions est effectuée en sélectionnant manuellement des parties représentatives sur l'image. Des distributions gaussiennes sont choisies pour leur proximité aux données, et les paramètres associés (moyenne et variance) sont alors déterminés.

Concernant la méthode de détection de fibres, plusieurs possibilités s'offrent aussi à nous, dont celles présentées dans la partie 1.2.3.2 et 1.2.4.3 :

1. Une détection de fibres par Transformée de Hough. Cette méthode étant utilisée plutôt pour détecter des droites que des cylindres complets, il est alors nécessaire dans un premier temps d'extraire de l'image des droites exploitables. Comme on l'a mentionné dans la partie 2.2, les fibres sont représentées assez fidèlement par des cylindres de même rayon. Ainsi, la détermination de génératrice des cylindres nous permet de retrouver les fibres. Or, la génératrice correspond à l'ensemble des points les plus éloignés de l'extérieur du cylindre, donc de la fibre. Ainsi, une opération de calcul de distance suivie d'un seuillage, ou une squelettisation morphologique permettent d'obtenir une image préparée pour la Transformée de Hough.

Les méthodes probabilistes dérivées de la Transformée de Hough, comme la RHT—Transformée de Hough Aléatoire, ou *Randomized Hough Transform* (Xu & Oja, 1990)—et la PHT—Transformée de Hough Probabilisée, ou *Probability Hough Transform* (Kiryati & Bruckstein, 1991), voir partie 3.4.1—s'articulent remarquablement bien avec le fait que les fibres d'un même fil soient d'orientations quasi-similaire. En effet, le côté itératif de ces méthodes va permettre, au fur et à mesure, de contraindre le domaine de recherche des fibres.

2. Une méthode utilisant le Tenseur de Structure. Dans ce cas, une segmentation « propre » n'est pas requise. Un filtre atténuant le bruit est nécessaire afin de ne pas nuire au traitement, et le choix d'une valeur seuil caractérisant la bordure des

fibres est alors requis (seuillage implicite). Cette méthode n'utilise que très peu les avantages accordés par la structure de l'image, ce qui laisse supposer un algorithme « passe-partout » non conçu pour des performances temporelles optimales.

Pour autant, la complexité algorithmique asymptotique des méthodes fondées sur le tenseur de structure est remarquablement bonne. Celle-ci peut donc donner des résultats potentiellement intéressants.

3. Les hypothèses que l'on a sur notre image nous permettent aussi de réfléchir à une méthode géométrique. En effet, la génératrice d'un cylindre est une droite intersectant les bords de l'image en deux points, centres de deux ellipses des bords de l'image (Figure 21). De fait, le calcul de l'ensemble des ellipses sur le bord de l'image et la déduction des centres associés nous donne des informations pertinentes pour la détection des fibres. Il « suffit » alors d'associer les centres deux à deux en constatant la présence ou l'absence d'une droite pleine les reliant.

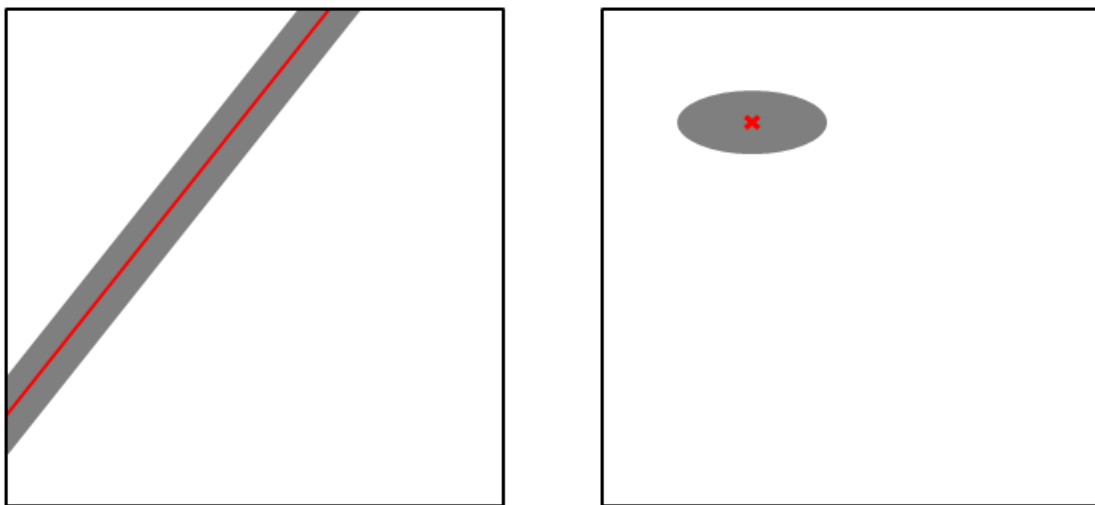


Figure 21 : Vue de profil et de coupe d'un cylindre traversant l'image. La ligne rouge représente la génératrice du cylindre, et à droite, l'ellipse représente l'intersection de la fibre avec le plan, et le point rouge, centre de l'ellipse, est aussi l'intersection de la génératrice avec le plan de coupe.

Cette méthode présente l'avantage de faire une partie du travail de détection uniquement sur les bords, ce qui réduit l'espace considéré et donc une partie du temps de calcul. De plus, l'association des centres peut être faite de manière itérative, permettant alors d'ajouter des contraintes d'orientation au fur et à mesure de l'exécution. La méthode

de « matching » géométrique est donc adaptée—dans le sens où elle utilise tous les présupposés sur l’image—et assez peu calculatoire.

On va maintenant décrire de manière explicite le fonctionnement théorique de ces algorithmes afin d’obtenir une idée de leurs complexités asymptotiques.

On considère une image de taille L^3 , contenant M fibres de rayon r , où r est constant selon L , et $M = O(L^2)$. Ces fibres sont réunies en m fils, et on a aussi $m = O(L^2)$.

Commençons par la méthode utilisant la **transformée de Hough** :

1. Segmentation de l’image par méthode de champs de Markov cachés (partie 2.4.1).
2. Squelettisation du résultat pour n’obtenir que les génératrices des cylindres.
3. Plusieurs choix de Transformée de Hough :
 - HT standard ou FHT—Transformée de Hough rapide ou Fast Hough Transform (Li, et al., 1986)—détection en une passe.
 - PHT avec évolution de contraintes pour tenir compte du fait qu’il n’existe que m zone d’orientations valables.

La complexité de la segmentation par champs de Markov se comporte telle que $C_{\text{Markov}} = O(n_{it} \times L^3)$, avec n_{it} le nombre d’itérations nécessaires à la convergence de la segmentation. Cette valeur est difficilement évaluable.

La squelettisation demande $r - 1$ opérations d’érosions successives car le rayon des fibres est connu et constant. Sa complexité vérifie par conséquent $C_{\text{Squelett}} = O(r \times L^3)$.

La caractérisation d’une ligne en 3D demande 4 paramètres et l’accumulateur de Hough est donc de dimension 4 dans notre cas. En supposant que l’on prenne un accumulateur de taille L^4 , on a finalement des complexités dominées par les valeurs suivantes : $C_{\text{HT}} = O(L^6)$, $C_{\text{FHT}} = O(L^5 \ln L)$ et $C_{\text{PHT}} = O\left(\frac{L^6}{r}\right)$ (la domination pour la PHT est donnée dans le pire des cas, en moyenne la complexité est meilleure).

La Transformée de Hough est l’étape qui a la pire complexité asymptotique de l’algorithme ; c’est pourquoi elle est l’étape limitante de la méthode. Finalement, on peut en

déduire, selon la HT choisie, une complexité asymptotique dominée par $O(L^6)$ voire dans les meilleurs cas $O(L^5)$.

Concernant la méthode de **tenseur de structure**, les étapes sont les suivantes :

1. Filtrage médian.
2. Calcul du gradient et du tenseur de structure.
3. Application d'un flou sur le tenseur de structure.
4. Calcul des éléments propres.
5. Calcul des rayons de courbure.
6. Enregistrements de ces données dans un accumulateur.
7. Simplification de l'accumulateur, représentation des droites.

A l'inverse de la méthode de Hough, la méthode issue du tenseur de structure représente les fibres de manière très locale, c'est-à-dire que chaque voxel d'une fibre peut ainsi se voir associé le point de la génératrice la plus proche et l'orientation de celle-ci en ce point. Dans le cas de courbes, cette méthode reste donc utilisable, mais le fait de considérer la fibre point par point ne tient pas compte des avantages fournis par l'image (partie 2.2) et rend la méthode assez longue en temps de calcul.

La complexité de l'ensemble des étapes 1 à 5 est linéaire par rapport à la taille de l'image, c'est-à-dire qu'elle vérifie $C_{\text{STRUCT}} = O(L^3)$, cependant la méthode est calculatoire, et la constante de temps associée est élevée. Concernant l'accumulation des résultats de chaque point considéré, la complexité de cette phase dépend fortement de la quantité d'information à retranscrire. Deux choix sont envisageables :

- Transmission uniquement des coordonnées du point. La détection de droite reste alors à faire sur la base des points détectés. Cela peut se faire localement par une méthode des noyaux, mais il est alors nécessaire de prendre une fenêtre de voisinage plus grande que la dispersion des points considérés. On a alors une méthode en $O(L^3)$, mais uniquement locale et peu précise.
- Transmission des coordonnées et de l'orientation locale de la fibre. Il est alors possible de détecter des droites de manière globale par des méthodes de K-

Means par exemple. L'algorithme est plus précis, mais sa complexité est alors en $O(L^5)$.

Finalement, le fonctionnement de **l'algorithme géométrique** est le suivant :

1. Segmentation de l'image par méthode de champs de Markov cachés.
2. Calcul de la carte de distance sur les faces de l'image.
3. Détermination des maximums locaux de cette carte de distance.
4. Calcul de la présence d'une droite ou non entre les paires de maximums locaux
5. Plusieurs choix d'acceptations :
 - Acceptation de tout ce qui est détecté, une seule passe.
 - Acceptation de la meilleure, considération des contraintes qui stipulent qu'il n'existe que m zones d'orientations possible, puis bouclage.

La complexité de la segmentation par champs de Markov se comporte telle que $C_{\text{Markov}} = O(n_{it} \times L^3)$, comme explicité pour la méthode de Hough.

En utilisant l'algorithme de Danielsson, le calcul de la carte de distance est une opération linéaire avec la taille, c'est-à-dire que l'on a $C_{\text{Distance}} = O(L^2)$. De même, la détermination des maximums locaux peut se faire aussi de manière linéaire, i.e. $C_{\text{Maximums}} = O(L^2)$.

On obtient alors $2M$ points, chaque point correspondant à l'extrémité d'une droite. Ce nombre de points se comporte donc asymptotiquement en $O(L^2)$. De fait, le nombre de doublets se comporte en $O(L^4)$, et le calcul de la présence d'une droite entre les deux points est une opération linéaire en L . Ainsi, le point 4 a donc une complexité asymptotique en $O(L^5)$, qui limite par conséquent l'algorithme (cependant, on verra grâce aux présupposés sur l'image, qu'il est possible d'atteindre une complexité de l'ordre de $O(L^3)$ dans certains cas).

On obtient finalement trois méthodes de complexités asymptotiques proches pour détecter les fibres de nos images. La méthode de Hough étant moins bonne asymptotiquement et la méthode issue du tenseur de structure ayant une constante de temps élevée, la méthode géométrique sera choisie par la suite. De plus, sa flexibilité

permettra, on le verra par la suite, de très nombreuses améliorations temporelles permettant d'atteindre des temps de calcul très prometteurs.

On supposera maintenant connu l'ensemble des fibres de l'image, sous une forme de liste de paramètres caractérisant les fibres. On recherche donc à associer entre elles les fibres proches en orientation et en localisation. Pour ce faire, nous allons nous servir de la méthode des K-means (MacQueen, 1967) ainsi que de la triangulation de Delaunay (Delaunay, 1934). La méthode des K-means permettra de rassembler les fibres en fonction de leurs orientations. Cependant, plusieurs fils peuvent avoir des orientations très semblables, d'une part, et on ne peut pas écarter la possibilité de détecter localement des fausses fibres inexistantes dans l'image réelle. Pour corriger ces défauts, une classification spatiale avec la triangulation de Delaunay sera ajoutée, permettant, en plus du clustering, une vérification et une correction des erreurs sur les résultats de l'algorithme de détection de droites.

2.4. Algorithme développé

2.4.1. Champs de Markov

On présente ici un modèle de segmentation par champs de Markov cachés que l'on utilisera sur les images de composites carbone/résine afin de séparer les fibres et les pores d'une part, et la résine d'autre part. Les modèles markoviens (MRFs : Markov Random Fields) sont une famille particulière de modèles stochastiques exprimant l'interdépendance entre les propriétés de séquences de données ou de champs aléatoires. Leurs applications sont nombreuses, et en particulier, ils donnent naissance à tout un ensemble de méthodes de traitement d'image et de segmentation (Geman & Geman, 1984). Ces modèles ajoutent, en plus de la considération sur la valeur du voxel, une considération spatiale, accordant à un voxel une région d'appartenance probable selon les valeurs de ses voisins.

Soit $E = D(\mathbf{X}) = \prod_{i < 3} [0 : L_i] \subset \mathbb{N}^3$ l'ensemble de définition de l'image $\mathbf{X} = \{X_e ; X_e \in [0 : 2^n - 1], e \in E\}$ considérée. On notera $e_{i,j,k}$ le voxel (i, j, k) de l'image.

Soit un ensemble d'étiquettes H représentant les différentes classes possibles pour le résultat de la segmentation.

Définition : Soit $\mathbf{Y} = \{Y_e, Y_e \in H, e \in E\}$ un champ de variables aléatoires défini sur l'ensemble E . \mathbf{Y} est un champ de Markov pour un système de voisinage \mathcal{N} donné si et seulement si il vérifie les deux conditions suivantes :

1. Positivité : $\forall \mathbf{y} \in H^E, \mathbb{P}(\mathbf{Y} = \mathbf{y}) > 0$
2. Condition markovienne :

$$\forall y_e \in H, \mathbb{P}(Y_e = y_e | \mathbf{Y}_{E \setminus e} = \mathbf{y}_{E \setminus e}) = \mathbb{P}(Y_e = y_e | \mathbf{Y}_{\mathcal{N}(e)} = \mathbf{y}_{\mathcal{N}(e)})$$

où $E \setminus e$ représente l'ensemble de définition de l'image privé de e et $\mathcal{N}(e)$ représente le voisinage de e (Li, 2009) (Celeux, et al., 2003).

Définition : Soit S un ensemble quelconque muni d'un système de voisinage \mathcal{N} . Une **clique** est un sous-ensemble $T \subset S$ tel que tous les éléments de T soient voisins deux à deux pour le système \mathcal{N} (en particulier, tout singleton est une clique).

La densité de probabilité associée à un champ de Markov est une distribution de Gibbs d'après le théorème d'Hammersley-Clifford (Geman & Geman, 1984), ce qui permet

d'exprimer le champ aléatoire via des propriétés locales plutôt que par des propriétés globales. De plus, on peut écrire la densité de probabilité de la manière suivante :

$$\mathbb{P}(\mathbf{Y} = \mathbf{y}) = \frac{\exp(-U(\mathbf{y}))}{Z}$$

où U est une énergie de la forme

$$U(\mathbf{y}) = \sum_{c \in \mathcal{C}} V_c(\mathbf{y})$$

qui correspond à la somme de potentiels des cliques sur l'ensemble des cliques possibles, et Z la constante de partition du système (Li, 2009), qui est posée de manière à ce que $\sum_{\mathbf{y}} \mathbb{P}(\mathbf{Y} = \mathbf{y}) = 1$, donc $Z = \sum_{\mathbf{y}} \exp(-U(\mathbf{y}))$.

Dans les faits, la constante de répartition Z n'est pas calculable même par ordinateur car le nombre de configurations possibles est énorme.

Toute la liberté de choix qui s'ensuit sur les modèles de Markov provient du choix de l'énergie, d'une part, et du choix du système de voisinage d'autre part. Les cas les plus classiques sont les choix de voisinages isotropes d'ordre 1,2 où 3 en 3D, ce qui correspond respectivement aux 6, 18 et 26-voisinages.

Dans notre cadre, on cherche à effectuer une segmentation à partir des données \mathbf{x} . On recherche donc une segmentation $\mathbf{Y} = \mathbf{y}$ maximisant $\mathbb{P}(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x})$. On pourra se placer dans un cadre bayésien, dans lequel on aura, grâce au théorème de Bayes :

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) = \frac{\mathbb{P}(\mathbf{X} = \mathbf{x} | \mathbf{Y} = \mathbf{y}) \mathbb{P}(\mathbf{Y} = \mathbf{y})}{\mathbb{P}(\mathbf{X} = \mathbf{x})}$$

$\mathbb{P}(\mathbf{X} = \mathbf{x})$ étant une constante dans notre cas, on est donc confronté à la maximisation de $\mathbb{P}(\mathbf{X} = \mathbf{x} | \mathbf{Y} = \mathbf{y}) \mathbb{P}(\mathbf{Y} = \mathbf{y})$. Si l'on suppose que \mathbf{Y} vérifie l'hypothèse markovienne, alors on peut utiliser le théorème d'Hammersley-Clifford. On peut donc écrire $\mathbb{P}(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x})$ sous la forme :

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) = \frac{\mathbb{P}(\mathbf{X} = \mathbf{x} | \mathbf{Y} = \mathbf{y})}{\mathbb{P}(\mathbf{X} = \mathbf{x})} \frac{\exp(-U(\mathbf{y}))}{Z}$$

et on cherche donc à maximiser $\exp(\ln \mathbb{P}(\mathbf{X} = \mathbf{x} | \mathbf{Y} = \mathbf{y}) - U(\mathbf{Y}))$. On a donc

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) \propto \exp -\mathcal{U}(\mathbf{y} | \mathbf{x})$$

avec, si on suppose les voxels conditionnellement indépendants,

$$\mathcal{U}(\mathbf{y} | \mathbf{x}) = - \sum_{e \in E} \ln p(x_e | y_e) + \sum_{c \in C} V_c(\mathbf{x})$$

avec $p(x_e | y_e) = \mathbb{P}(X_e = x_e | Y_e = y_e)$.

Ainsi, $\mathbf{Y} | \mathbf{X}$ est un champ de Gibbs, et, par le théorème d'Hammersley-Clifford (Hammersley & Clifford, 1971), il est aussi un champ de Markov.

La détermination de la forme de l'énergie \mathcal{U} est l'un des points importants des modèles de champs de Markov. Le modèle de Potts (Wu, 1982) homogène est une manière d'écrire cette énergie et la corrélation entre les voxels de la manière suivante : la fonctionnelle en énergie s'écrit comme la somme d'un potentiel donné V_2 sur tous les doublets de \mathbf{y} voisins pour le système de voisinage considéré \mathcal{N} et d'un potentiel simple V_1 sur tous les points correspondant au calcul précédent ($V_1(y_e, x_e) = -\ln p(x_e | y_e)$), assorti d'un paramètre β nommé paramètre d'interaction spatiale, c'est-à-dire en ne considérant que les cliques à un et deux points. Ainsi, on peut exprimer U de la manière suivante (Li, 2009) (Celeux, et al., 2003) :

$$\mathcal{U}(\mathbf{y} | \mathbf{x}) = \sum_{e \in E} \left(-\ln p(x_e | y_e) + \beta \sum_{f \in \mathcal{N}(e)} V_2(y_e, y_f) \right)$$

Le premier terme de cette expression est la contribution « d'attache aux données », qui dépend d'une densité de probabilité associée à chaque étiquette. Le second terme, correspondant à une contribution spatiale régularisante, est, dans le modèle de Potts, exprimable sous :

$$V_2(y_e, y_f) = -\delta(y_e, y_f)$$

avec δ la fonction de Kronecker.

Dans le cas où $\text{card } H = 2$, on reconnaîtra le modèle d'Ising (Ising, 1925).

La segmentation d'images par champ de Markov caché a donc pour but de retrouver la réalisation \mathbf{y} minimisant l'énergie $\mathcal{U}(\cdot | \mathbf{x})$. Pour obtenir ce résultat, les méthodes courantes consistent à se donner une réalisation \mathbf{y}^0 , puis à converger vers un minimum de la fonctionnelle énergie via des algorithmes de relaxation. Geman et Geman (Geman & Geman, 1984) proposent une méthode construisant les images via des tirages successifs suivant la loi de Gibbs. L'énergie peut être considérée comme une somme de contributions locales pour chaque voxel, cette contribution étant quantifiée par

$$\mathcal{U}_l(y_e | \mathbf{y}_{\mathcal{N}(e)}, x_e) = -\ln p(x_e | y_e) - \beta \sum_{f \in \mathcal{N}(e)} \delta(y_e, y_f)$$

Via cette quantification, on peut calculer, voxel par voxel, la valeur de y_e minimisant l'énergie locale et par là-même l'énergie globale. La méthode de l'échantillonneur de Gibbs procède donc comme suit pour le calcul de \mathbf{y}^n connaissant \mathbf{y}^{n-1} :

- Choix d'un site $e \in E$.
- Calcul, pour toutes les valeurs possibles de $y_e \in H$, de la probabilité locale

$$\mathbb{P}(Y_e = y_e | \mathbf{y}_{\mathcal{N}(e)}, x_e) = \frac{\exp -\mathcal{U}_l(y_e | \mathbf{y}^{n-1}_{\mathcal{N}(e)}, x_e)}{\sum_{h \in H} \exp -\mathcal{U}_l(h | \mathbf{y}^{n-1}_{\mathcal{N}(e)}, x_e)}$$

- Choix de la valeur de y_e^n via un tirage aléatoire suivant la loi $\mathbb{P}(\cdot | \mathbf{y}_{\mathcal{N}(e)}, x_e)$.

Dans les faits, pour améliorer la vitesse de convergence, le choix du site se fera de manière déterministe en effectuant tous les points de l'image successivement.

Il est possible aussi d'utiliser un algorithme de Métropolis (Metropolis, et al., 1953) pour faire converger le résultat. La seule différence avec l'échantillonneur de Gibbs porte sur le choix de la valeur de y_e^n qui se fait alors en prenant une classe au hasard, en calculant la différence d'énergie, et en acceptant le changement si la différence est négative, et en l'acceptant uniquement avec une probabilité donnée si cette différence est positive.

Ces deux méthodes, grâce à leur aspect probabiliste, peuvent localement accepter des modifications augmentant l'énergie, ce qui leur permet de ne pas bloquer sur tout minimum local de $\mathcal{U}(\cdot | \mathbf{x})$. Cependant, la convergence vers le minimum global se fait en temps potentiellement infini, ce qui fait que l'on ne peut pas considérer en pratique ces algorithmes comme convergeant vers la meilleure solution. La méthode de convergence du

recuit simulé (Kirkpatrick, et al., 1982) permet, via un « lissage » dégressif de la fonction minimisée grâce à l'ajout d'un paramètre de température, d'améliorer la convergence, mais encore une fois sans assurer sa rapidité.

Le choix de la réalisation de départ \mathbf{y}^0 se révèle donc capital pour assurer une convergence rapide vers un résultat convenable. Dans le cas où \mathbf{y}^0 est choisi très proche de la réalisation finale voulue, il est même envisageable d'effectuer une optimisation déterministe voxel par voxel, par exemple en suivant une optimisation de Powell (Powell, 1969) (Powell, 1970) sur l'ensemble des étiquettes. Pour obtenir cette réalisation \mathbf{y}^0 , on choisit manuellement des zones représentatives de chaque région différente, puis on cherche une loi de probabilité dont la densité est semblable, à un facteur multiplicatif près, à l'histogramme de la région représentative.

Concernant le choix du paramètre β , il est possible d'utiliser des méthodes d'optimisation. Le meilleur choix d'optimisation serait de chercher le maximum de vraisemblance $\mathbb{P}(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}, \beta)$, que l'on obtiendrait en cherchant le maximum de la fonction de vraisemblance :

$$L(\beta) = \mathbb{P}_\beta(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) = \frac{\exp -\mathcal{U}(\mathbf{y} | \mathbf{x}, \beta)}{Z(\mathbf{x}, \beta)}$$

Le calcul de $Z(\mathbf{x}, \beta)$ étant impossible en temps raisonnable vu le nombre de configurations possibles, on peut utiliser à la place le maximum de la fonction de pseudo-vraisemblance (Geman & Graffigne, 1987), que l'on peut écrire comme :

$$PL(\beta) = \prod_{s \in E} \mathbb{P}(Y_s = y_s | \mathbf{y}_{N(s)}, x_s, \beta)$$

ce qui nous donne, d'après le calcul de la probabilité locale :

$$PL(\beta) = \prod_{s \in E} \frac{\exp -\mathcal{U}_l(y_s | \mathbf{y}_{N(s)}, x_s, \beta)}{\sum_{h \in H} \exp -\mathcal{U}_l(h | \mathbf{y}_{N(s)}, x_s, \beta)}$$

On utilisera préférentiellement le logarithme de la pseudo-vraisemblance, exprimable par :

$$\ln PL(\beta) = \sum_{s \in E} \left(-\mathcal{U}_l(y_s | \mathbf{y}_{N(s)}, x_s, \beta) - \ln \sum_{h \in H} \exp -\mathcal{U}_l(h | \mathbf{y}_{N(s)}, x_s, \beta) \right)$$

Le comportement de cette fonction est le même que celui de la fonction :

$$LL(\beta) = \sum_{s \in E} \left(\beta \sum_{t \in \mathcal{N}(s)} \delta(y_s, y_t) - \ln \left(\sum_{h \in H} \exp \beta \sum_{t \in \mathcal{N}(s)} \delta(h, y_t) \right) \right)$$

La log-pseudo-vraisemblance est plus aisément calculable ; il est alors possible d'utiliser des méthodes d'optimisation classiques comme une descente de gradient pour trouver son maximum (Besag, 1975) (Besag, 1977). La consistance de l'estimateur de la pseudo-vraisemblance et la convergence des méthodes ont été démontrées par Geman et Graffigne (Geman & Graffigne, 1987).

D'autres méthodes d'estimation des paramètres existent (Li, 2009), qu'il s'agisse de la méthode des codages (Besag, 1974), de l'Expectation-Maximization (EM) (Celeux, et al., 2003), méthode des moindres carrés (Derin & Elliott, 1987), voire dans des cas plus complexes de méthodes de Monte Carlo à chaînes de Markov (MCMC) (Wang, et al., 2000), comme l'algorithme de Metropolis-Hastings (Hastings, 1970) (Geman, 1991).

Dans la grande majorité des cas, on sera confronté à un modèle d'Ising à un seul paramètre comme présenté ci-dessus, ce qui rend l'estimation suffisamment simple avec l'estimateur de pseudo-vraisemblance.

Dans le cas des composites carbone/résine, on se trouve dans le cadre du modèle d'Ising, et on explicite ici le protocole à suivre pour l'algorithme de traitement d'image développé :

On suppose qu'on connaît pour le moment notre image de départ X , le paramètre β , le nombre de labels H et les distributions issues de l'initialisation associées à chaque classe, ici des gaussiennes $\mathcal{N}(m_h^0, \sigma_h^0)_{h \leq H}$. Le protocole est le suivant :

1. Création de l'image résultat Y^0 .
2. Initialisation de cette image $\forall s \in S, Y^0(s) = l$ tel que $p(X(s)|l) = \max_h p(X(s)|h)$
3. Démarrage de la boucle principale ; on note i l'indice d'itération courante :

- a. Pour tout $s = (u, v, w) \in S$ tel que $u + v + w \equiv i[2]$, on calcule, pour tous les labels h , les énergies locales associées :

$$u_{loc}(s, h) = -\ln p(X(s)|h) + \beta \sum_{t \in \mathcal{N}(s)} 1 - \delta(h, Y^{i-1}(t))$$

- b. Calcul de Y^i en actualisant les labels selon la méthode de l'échantillonneur de Gibbs
- c. Calcul des $\mathcal{N}(m_h^{i+1}, \sigma_h^{i+1})_{h \leq H}$:

$$m_h^{i+1} = \frac{1}{\text{card}\{s \in S, Y^i(s) = h\}} \sum_{s \in S, Y^i(s) = h} X(s)$$

$$(\sigma_h^{i+1})^2 = \frac{1}{\text{card}\{s \in S, Y^i(s) = h\}} \sum_{s \in S, Y^i(s) = h} X(s)^2 - (m_h^{i+1})^2$$

- d. Eventuellement optimisation de β .

Pour effectuer l'algorithme sans avoir à modifier à la volée la matrice, on décide donc de calculer sur un « damier » 3D où les voxels n'interagissent pas entre eux. Il est à noter que cette méthode fonctionne uniquement pour du 6 ou du 18-voisinage. Pour un 26-voisinage, il faut 8 passes pour considérer tous les voxels sans interactions.

Dans un cadre parallèle, on notera en plus que ce calcul par damier permet une parallélisation aisée du code, avec un échange de bords à chaque fin d'itération. Au contraire, si on effectue le calcul à la volée, on est alors obligé soit de contraindre les bords à échanger leurs valeurs avec les processeurs voisins avant que ceux-ci n'agissent, mais on agit alors de manière séquentielle, soit à ne modifier les bords qu'une itération sur deux, et l'intérieur qu'une fois sur deux, mais on perd alors l'isotropie de traitement, et les résultats différeront alors de fait d'une version séquentielle.

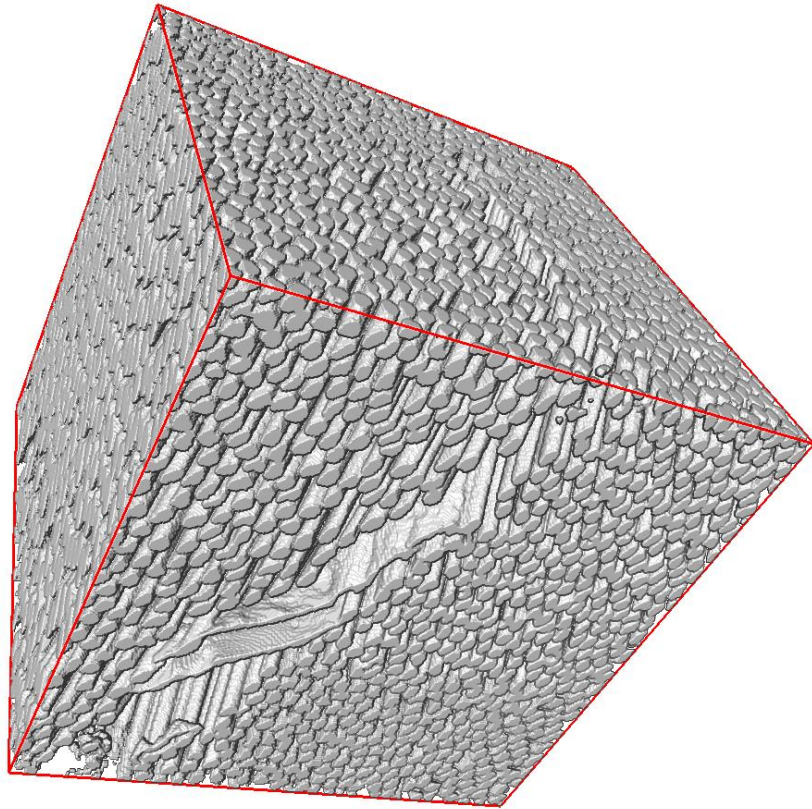


Figure 22 : Résultat de la segmentation par champs de Markov de l'image présentée Figure 14. On a choisi ici $\beta = 0,5$, et deux classes à distributions gaussiennes.

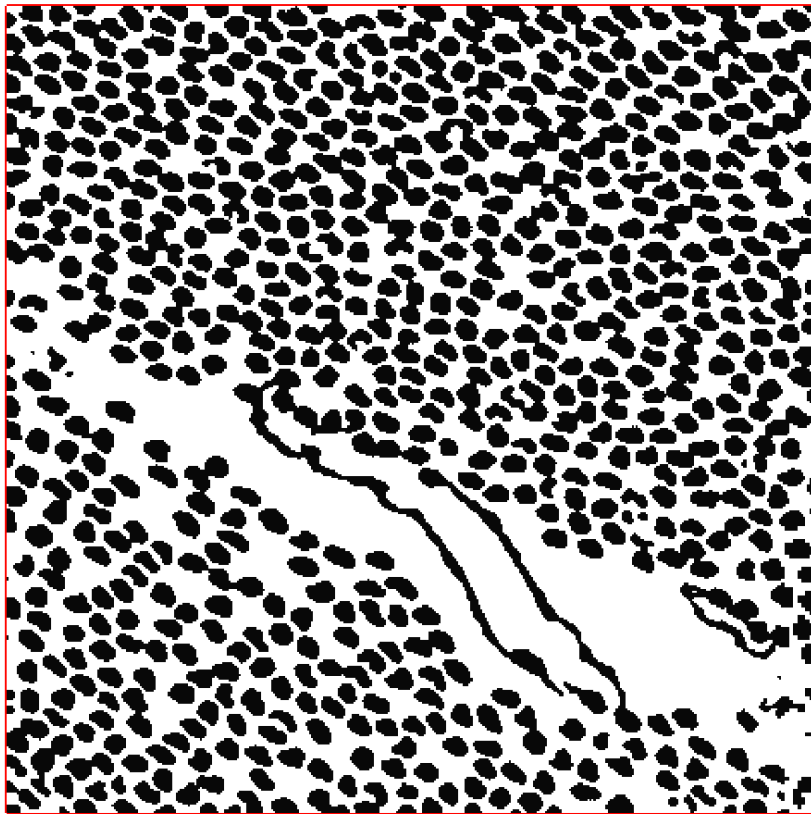


Figure 23 : Coupe de la segmentation correspondant à la coupe de la Figure 15.

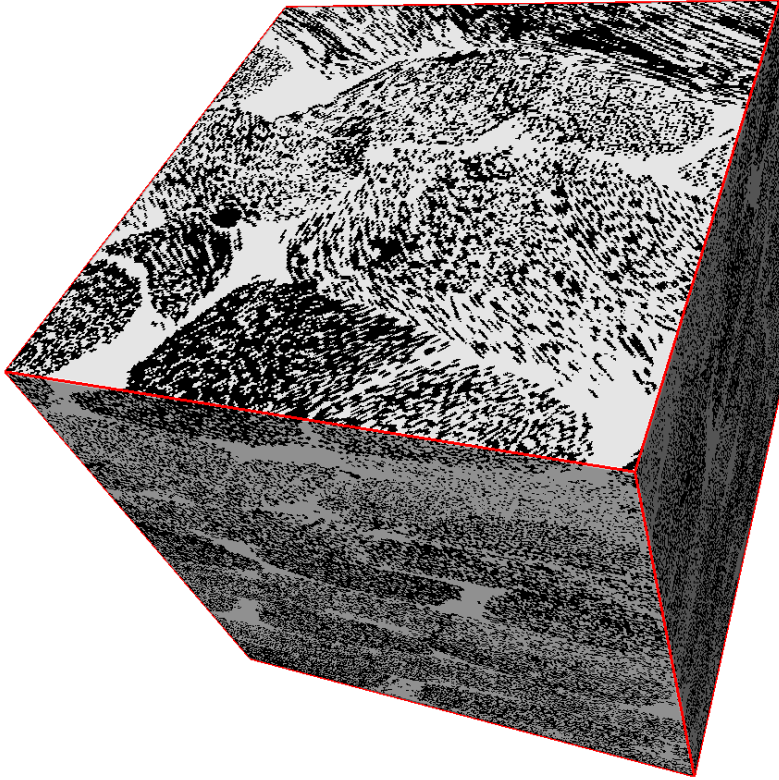


Figure 24 : Résultat de la segmentation par champs de Markov de l'image présentée Figure 16. On a choisi ici $\beta = 0,18$, et deux classes à distributions gaussiennes.

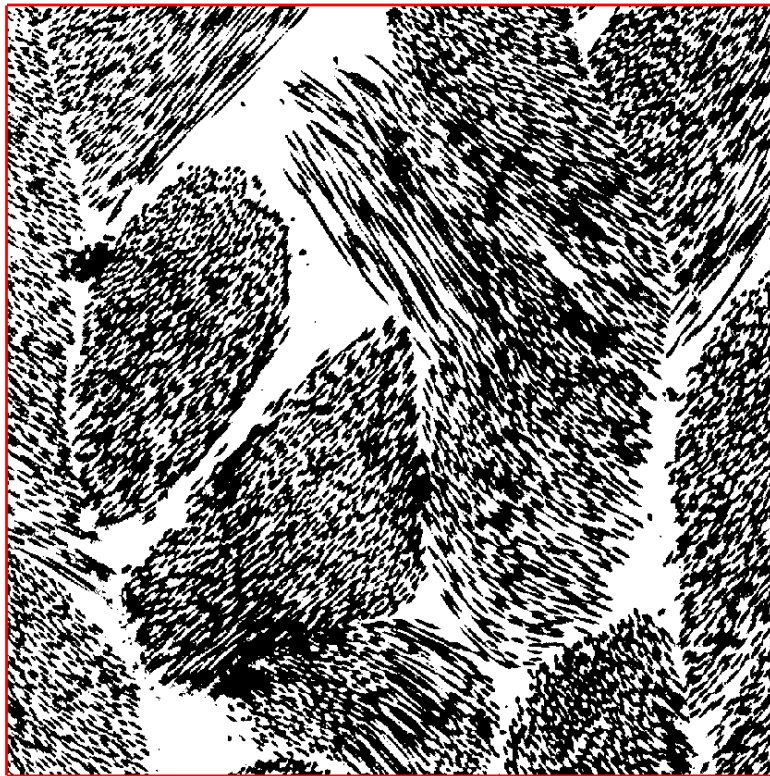


Figure 25 : Coupe de la segmentation correspondant à la coupe de la Figure 17.

2.4.2. Détection de Lignes

Ré-explicitons ici le principe de l'algorithme géométrique le plus simple afin de détailler son fonctionnement précis.

La segmentation a été effectuée dans la partie ci-dessus. Le reste est par conséquent :

1. Calcul de la carte de distance sur les faces de l'image.
2. Détermination des maximums locaux de cette carte de distance.
3. Calcul de la présence d'une droite ou non entre les paires de maximums locaux
4. Acceptation de tout ce qui est détecté.

On se donne donc une image binaire $J : E \rightarrow \{0 ; 1\}$ définie sur un ensemble $E = \llbracket 1 ; L \rrbracket^3$ (par commodité, on considèrera donc l'image comme étant cubique). On note $\partial E = \{(u, v, w) \in E \text{ tel que } u \in \{1 ; L\} \text{ ou } v \in \{1 ; L\} \text{ ou } w \in \{1 ; L\}\}$. On définit les sous-ensembles de E notés $J^{-1}(1)$ et $J^{-1}(0)$ comme les images réciproques de 1 et de 0 par l'image J .

On note ∂J l'image définie sur E par :

$$\partial J(s) = \begin{cases} J(s) & \text{si } s \in \partial E \\ 0 & \text{sinon} \end{cases}$$

qui est donc l'image conservant uniquement les données aux bords de J .

On commence donc par s'intéresser au calcul d'une carte de distance euclidienne sur une image 2D :

Soit un ensemble discret $E = \prod_{i < n} \llbracket 0 ; L_i \rrbracket \subset \mathbb{N}^n$ sur lequel on se donne un sous-ensemble F . On définit la **carte de distance** de F sur E comme l'image $D_F : E \rightarrow \mathbb{R}_+$ définie par $D_F(x) = d_{L^2}(x, F)$ où d_{L^2} correspond à la distance euclidienne.

La distance euclidienne n'est cependant pas une distance aisément déterminable, car elle n'est pas adaptée à l'ensemble discret \mathbb{N}^n . En effet, dans le cas de la distance L^1 , la connaissance de $d_{L^1}(\cdot, F)$ sur le voisinage de x dans \mathbb{N}^n noté $\mathcal{N}(x)$ permet de déterminer

sans ambiguïté $d_{L^1}(x, F)$. La distance L^1 se propage simplement par voisinage, ce qui permet de calculer une carte de distance associée avec une complexité linéaire (Figure 26).

La distance L^2 n'étant pas adaptée à la discrétisation des images, il est donc impossible de la propager par voisinage (Figure 27). La manière naïve de calculer la carte de distance euclidienne consiste simplement en un calcul point par point, qui consiste à déterminer, pour chaque point $x \in E$, la distance minimale avec F en la déterminant individuellement pour chaque point de F . Une autre variante un peu plus acceptable consisterait à faire progresser des sphères concentriques autour de chaque point $x \in E$ jusqu'à ce que l'une d'entre elle contienne un point de F . Ces méthodes élémentaires ont l'avantage d'être totalement exactes, mais elles se montrent aussi extrêmement lentes dans la très grande majorité des cas (avec une complexité algorithmique en $O(N^2)$), et il est par conséquent inenvisageable de les intégrer dans un quelconque algorithme.

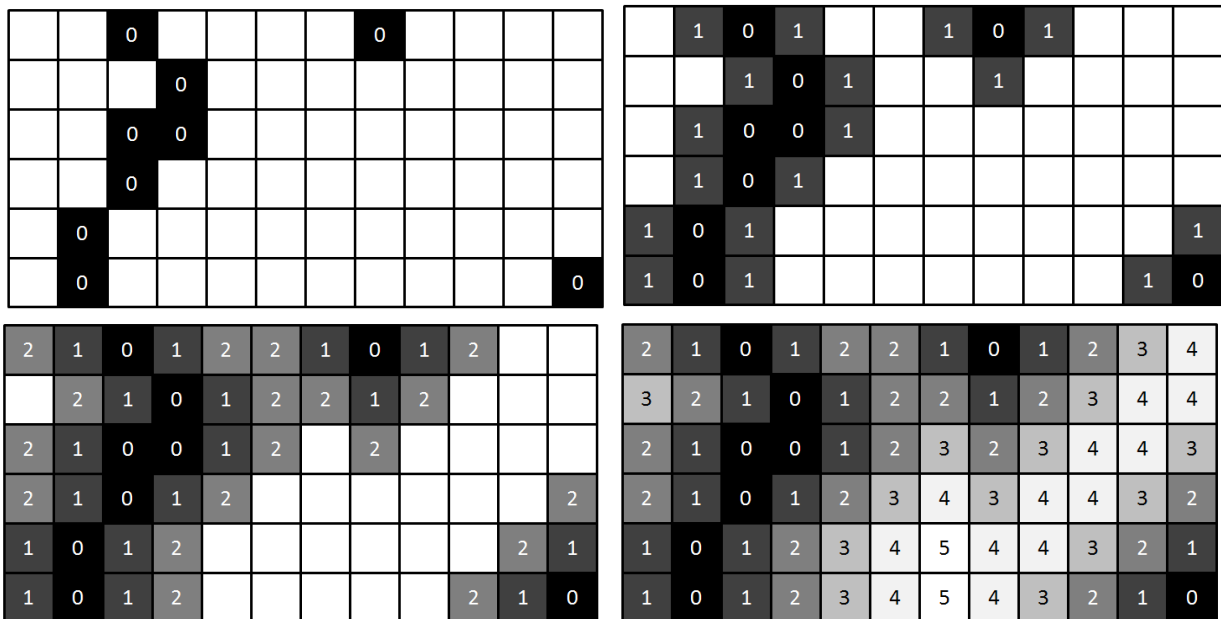


Figure 26 : Exemple de propagation de la distance L^1 sur une image discrétisée. Par simple itérations, il est possible de calculer la distance pour tous les points.



Figure 27 : Carte de distance L^2 pour l'image utilisée sur la Figure 26. Elle n'est pas déterminable par simple propagation.

Dans la majorité des cas, sur une image discrète, les utilisations faites des cartes de distances peuvent se contenter d'erreurs sur la distance inférieure à l'unité, des arrondis étant déjà souvent effectués pour simplifier les calculs. Dans ce cas, il est envisageable, sous réserve de bons résultats temporel d'utiliser une telle technique pour satisfaire les besoins en carte de distance d'algorithmes futurs si l'on exhibe une méthode approchée de calcul de la distance euclidienne présentant des erreurs inférieures à l'unité. Des méthodes d'approximation de la distance euclidienne par propagation via un masque fixé ont été proposées d'abord par Rosenfeld et Pfaltz (Rosenfeld & Pfaltz, 1966) (Rosenfeld & Pfaltz, 1968), étudiant les approximations euclidiennes pour différents types de masques (en croix, carré, hexagonal, octogonal, etc...) (Figure 28).

6	5	4	3	4	5	6
5	4	3	2	3	4	5
4	3	2	1	2	3	4
3	2	1	0	1	2	3
4	3	2	1	2	3	4
5	4	3	2	3	4	5
6	5	4	3	4	5	6

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

4	4	3	3	3	4	4
4	3	2	2	2	3	4
3	2	2	1	2	2	3
3	2	1	0	1	2	3
3	2	2	1	2	2	3
4	3	2	2	2	3	4
4	4	3	3	3	4	4

Figure 28 : Masques de distance présentés par Rosenfeld et Pfaltz. Le premier masque, en croix, correspond au masque associé à la norme 1, le deuxième masque en carré à la norme ∞ , et le troisième est un masque octogonal, approchant la norme 2.

Une implémentation en deux passes sans masque mais avec propagation de deux coordonnées (Figure 29) au voisinage au lieu d'une distance simple est proposée par Danielsson (Danielsson, 1980), qui permet de maîtriser l'erreur de manière satisfaisante.

	1;1	0;1	1;1						1;0		1;0	
	1;0		1;0		1;1	0;1	0;1	1;1	1;1	0;1	1;1	
	1;1	0;1	1;1		1;0			1;0				
					1;1	0;1	0;1	1;1				
					1;1	0;1	1;1					
	1;1	0;1	1;1		1;0		1;0					
	1;0		1;0		1;1	0;1	1;1					
	1;1	0;1	1;1									

2;1	1;1	0;1	1;1	2;1	1;2	0;2	0;2	2;0	1;0		1;0	2;0
2;0	1;0		1;0	2;0	1;1	0;1	0;1	1;1	1;1	0;1	1;1	2;1
2;1	1;1	0;1	1;1	2;0	1;0			1;0	2;0	0;2	1;2	2;2
2;2	1;2	0;2	1;2	2;1	1;1	0;1	0;1	1;1	2;1	0;3	1;3	2;3
2;2	1;2	0;2	1;2	2;1	1;1	0;1	1;1	2;1	2;2	3;2	1;4	2;4
2;1	1;1	0;1	1;1	2;0	1;0		1;0	2;0	3;0	4;0	5;0	6;0
2;0	1;0		1;0	2;0	1;1	0;1	1;1	2;1	3;1	4;1	5;1	6;1
2;1	1;1	0;1	1;1	2;1	1;2	0;2	1;2	2;2	3;2	4;2	5;2	6;2

Figure 29 : Exemple de propagation des coordonnées dans l'algorithme de Danielsson.

L'algorithme proposé par Danielsson est fondé sur le principe suivant, vrai pour la distance L^1 : Si l'on a un point p , et que pour chaque voisin q de p , on connaisse les coordonnées de $\vec{d}(q)$ le vecteur reliant q au point de F le plus proche, alors si on note $q_m \in \mathcal{N}(p)$ le voisin de p minimisant $\|\vec{d}(q)\|$, les coordonnées du vecteur $\vec{d}(p)$ sont :

$$\vec{d}(p) = \vec{d}(q_m) + \overrightarrow{q_m p}$$

Cette propriété est malheureusement fautive sur L^2 (Figure 30 et Figure 31). Elle se base sur le fait que chaque point a au moins un voisin qui appartient à la même cellule de Voronoï, ce qui est faux pour la distance euclidienne quel que soit le voisinage choisi.

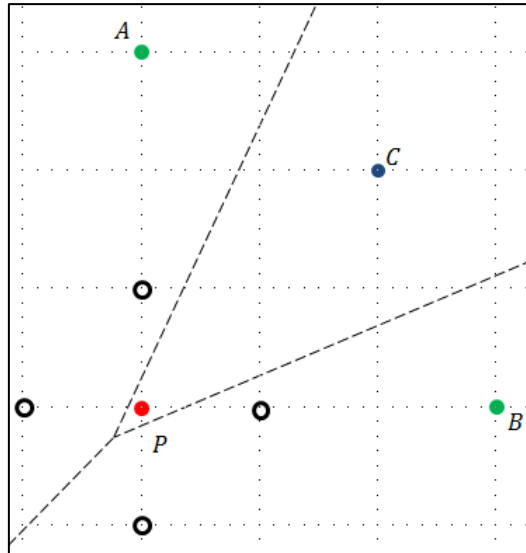


Figure 30 : Contre-exemple à la propriété pour la distance euclidienne. Ici, P appartient à la cellule de Voronoï de C , mais aucun de ses voisins (pour un 4-voisinage) n'appartient à la même cellule.

1;1	0;1	1;0		1;0
1;0		1;0	0;1	1;1
0;1	0;1	1;1	0;2	1;2
	1;0	2;0	3;0	1;3
0;1	1;1	2;1	3;1	4;1

Figure 31 : Exemple d'erreur pour l'algorithme de Danielsson. Ici, par propagation des coordonnées, le point à l'intersection des cellules de Voronoï se voit doter du vecteur (3 ;0), alors que le vecteur réel est (2 ;2), mais ne peut pas être propagé par un 4-voisinage.

Cependant, on peut constater que dans le cas où cette propriété n'est pas vérifiée, alors la cellule de Voronoï associée au point considéré a une forme particulière, impliquant des cellules voisines très proches, dont on peut déduire des encadrements sur l'erreur entre la valeur réelle de $\|\vec{d}(p)\|$ et celle qui sera donnée par la méthode (Figure 32).

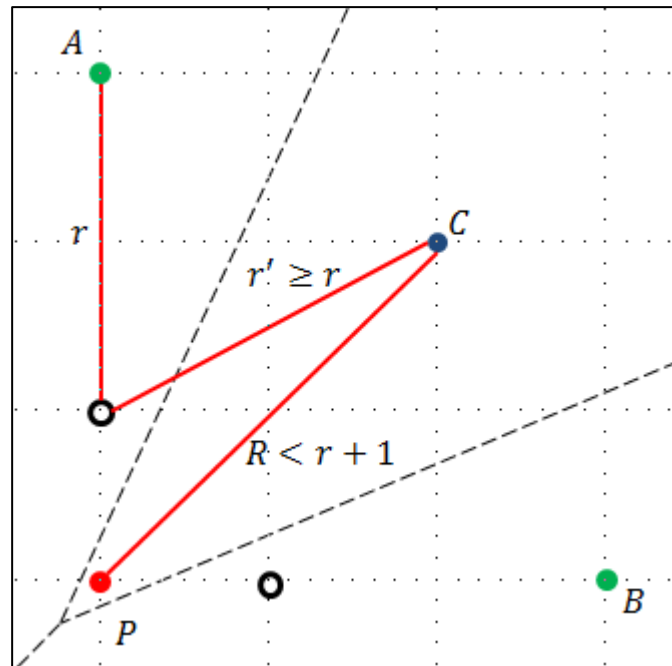


Figure 32 : Contraintes sur la distances entre P et C en fonction de celle entre A et P .

En deux dimensions, Danielsson a montré que l'erreur induite par l'algorithme $e = r + 1 - R$ ne dépassait jamais 0,58 pixel en théorie pour des images 2D, et était même en pratique encore plus petite, le cas théorique ne pouvant pas apparaître sur une image discrète. En effet, le pire cas limite apparaît quand la cellule de Voronoï de C forme un angle droit en P , et qu'on a $r' = r$ (Figure 32). Dans un repère orthonormé en P , les points ont les coordonnées suivantes : $P(0 ; 0)$, $A(0 ; r + 1)$, $B(r + 1 ; 0)$, $C(x, y)$, et C vérifie le système d'équations suivant :

$$\begin{cases} (x - 1)^2 + y^2 = r^2 \\ x^2 + (y - 1)^2 = r^2 \end{cases}$$

Or par symétrie, $x = y$, donc on a finalement :

$$2x^2 - 2x + 1 - r^2 = 0$$

donc

$$x = \frac{1 + \sqrt{2r^2 - 1}}{2}$$

et en définitive

$$e = r + 1 - \sqrt{r^2 + \sqrt{2r^2 - 1}}$$

Le maximum de cette fonction est bien atteint pour $r = 1$, et $e_{\max} = 2 - \sqrt{2} \sim 0,58$.

En dimension trois (et par extension en dimension quelconque, où la généralisation de la méthode en dimension supérieure a été effectuée par Ragnemalm (Ragnemalm, 1993) et l'étude comparative de la distance avec d'autres méthodes classiques en dimension supérieure par Borgefors (Borgefors, 1984) (Borgefors, 1986)), l'erreur est légèrement plus importante. En effet, nous avons maintenant besoin d'un point supplémentaire, et la cellule de Voronoï de C doit former un tétraèdre rectangle en P . Cette fois-ci, l'erreur dans le pire des cas s'exprime par :

$$e = r + 1 - \sqrt{r^2 + \frac{2\sqrt{3r^2 - 2} - 1}{3}}$$

qui atteint aussi son maximum en $r = 1$, et on a donc $e_{\max} = 2 - \frac{2}{\sqrt{3}} \sim 0,85$. En pratique, l'erreur maximale est même de 0,60.

Concernant la complexité de cet algorithme, celui-ci ne faisant qu'un nombre fini de passes sur l'image, on peut prouver le comportement linéaire de la méthode. De plus, on peut relier le nombre d'opérations C_n de la méthode pour une image de dimension n à celui nécessaire pour les sous-images de dimension $n - 1$. Si l'image est un hypercube de taille L^n , alors on a :

$$C_n = 2L^n + 2LC_{n-1}$$

donc

$$C_n = \left(\sum_{k=1}^n 2^k \right) L^n$$

qui est linéaire en $+\infty$ suivant la taille de l'image. On constatera que la constante de dimension $(\sum_{k=1}^n 2^k)$ augmente rapidement avec n , cette méthode devient donc de moins en moins intéressante en pratique (et non pas asymptotiquement) si la dimension augmente.

Il est à noter qu'il est heureusement aussi possible de calculer des distances euclidiennes exactes sans passer par la propagation, le tout dans des temps plus raisonnables. Une des idées utilisées, proposée par Paglieroni (Paglieroni, 1992), consiste en un découpage d'une image de taille M^2 en M sous-images linéiques de taille M correspondant à autant d'hyperplans. Sur chaque sous-image, le calcul de la distance est effectué puis les résultats sont fusionnés simplement en regardant les évolutions de la distance dans le sens orthogonal au découpage. Le traitement de la distance se fait donc de manière récursive sur la dimension de l'image. Ces méthodes ont l'avantage de fournir, en plus de la distance, le point de F associé à cette distance minimale, ce qui leur permet aussi d'obtenir une complexité en $O(N^{3/2})$. On notera que cette technique prend implicitement en considération le diagramme de Voronoï (Paglieroni, 1992).

Des améliorations sur la complexité algorithmique ont ensuite été proposées, réduisant l'impact de la phase de fusion des résultats le long des images linéiques, d'abord en $O(N \ln N)$ par une méthode de *divide-and-conquer* (Kolountzakis & Kutulakos, 1992), puis par une intégration de calculs de diagrammes de Voronoï locaux (Breu, et al., 1995), permettant une application de la méthode à n'importe quelle dimension d'image (Maurer, et al., 2003).

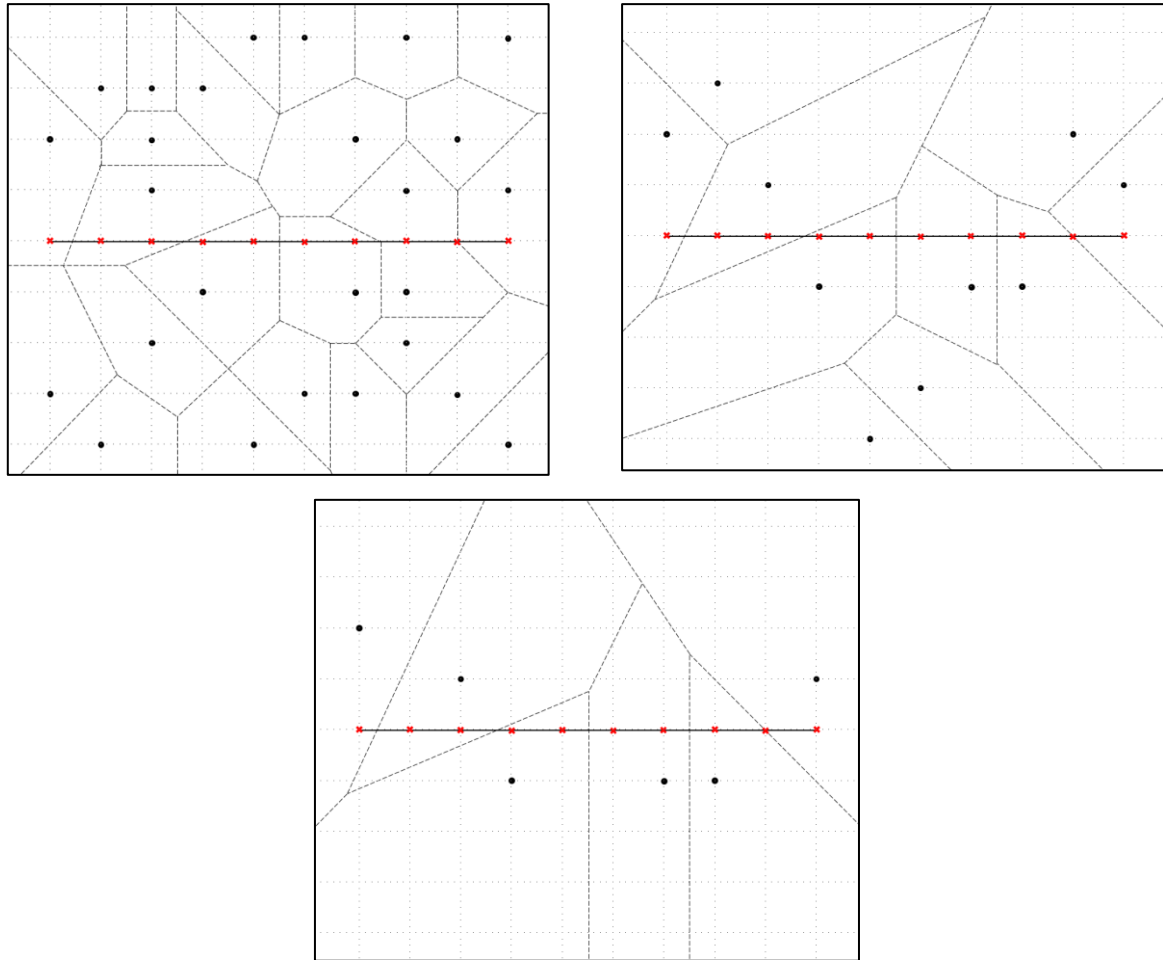


Figure 33 : Exemple de fonctionnement sur une ligne de l'algorithme fondé sur les diagrammes de Voronoï locaux.

La Figure 33 montre un exemple de fonctionnement de l'algorithme. L'image de base est présentée en haut à gauche, chaque point d'intersection de la grille représente un point de l'image, les points de F étant marqués par un point noir. Les lignes pointillées représentent les cellules de Voronoï. L'algorithme va calculer le diagramme de Voronoï local pour chaque ligne (ici, on considère une ligne matérialisée par le trait noir), tout d'abord en regardant, pour chaque point de celle-ci, le point de F le plus proche dans l'hyperplan orthogonal (en dimension 2, ces hyperplans sont juste des droites verticales), ce qui donne l'image en haut à droite, puis ensuite en réduisant cet ensemble de points pour ne considérer que les points les plus proches des points de la ligne (image en bas). On constatera que les points de la ligne sont bien associés à la même cellule de Voronoï que sur l'image de départ.

Il est à noter que le calcul des distances se rapproche du domaine de la morphologie mathématique (Serra, 1982), la distance étant une information primordiale de la morphologie—pour la squelettisation par exemple. De fait, des méthodes de calcul de

distances dérivées de la morphologie existent aussi (Huang & Mitchell, 1994) (Vincent, 1991). On observe que la diversité des méthodes de calcul de la distance euclidienne, qu'elles soient approchées ou réelles, permet sans aucun doute de trouver des algorithmes performants dans toute situation.

En premier lieu, on commence donc par calculer l'image $\partial\mathbf{K}$ résultat de la carte de distance appliquée à $\partial\mathbf{J}$ grâce à la méthode de Danielsson (Danielsson, 1980).

Puis, à partir de cette image $\partial\mathbf{K}$, on détermine les maximums locaux que l'on note $\mathcal{C} = (c_k)_{k \leq N}$. On note $A_{u,v}$ l'ensemble défini par :

$$A_{u,v} = \{s \in E : d(s, (c_u c_v)) \leq 1\}$$

On se donne alors une matrice \mathbf{H} de taille $N \times N$, telle que :

$$H_{u,v} = \frac{\text{card}(A_{u,v} \cap \mathbf{J}^{-1}(1))}{\text{card}A_{u,v}}$$

Cette matrice est la matrice de correspondance entre les points considérés, la valeur en l'indice (u, v) répertorie le ratio de points sur le trajet de la droite $(c_u c_v)$ qui appartiennent à $\mathbf{J}^{-1}(1)$ c'est-à-dire le ratio de points appartenant à une des fibres de l'image (voir Figure 34).

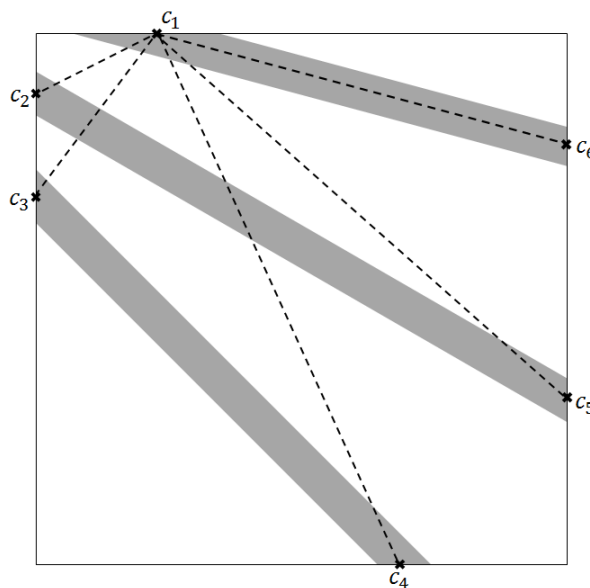


Figure 34 : Schéma décrivant le fonctionnement de l'algorithme de détection. Celui-ci regarde l'ensemble des doublets possibles et constate s'ils correspondent à une fibre. Ici, $(c_1 c_6)$ y correspond.

On note ensuite G l'ensemble des droites suivant :

$$G = \{(c_u c_v), (u, v) \in \llbracket 1 ; N \rrbracket^2 \text{ tel que } H_{u,v} = 1\}$$

D'après les hypothèses du point 2.2, les points $(c_k)_{k \leq N}$ sont les intersections des génératrices des cylindres de l'image avec les bords de celle-ci. De fait, si $(c_u c_v)$ est une génératrice d'un cylindre de l'image, on a $H_{u,v} = 1$. Si $(c_u c_v)$ n'est pas une génératrice, alors les deux points appartiennent à deux cylindres différents, et, si $J^{-1}(1)$ ne correspond bien qu'à l'ensemble des cylindres, alors il existe un point qui n'appartient pas à $J^{-1}(1)$ sur le tracé, et donc $H_{u,v} < 1$.

Ainsi, G représente exactement l'ensemble des génératrices de tous les cylindres de l'image.

On a ainsi construit une méthode rapide qui détecte les cylindres d'une image 3D. Cependant, on a supposé ici que les fibres ne se touchaient pas, et que $J^{-1}(1)$ correspondaient exactement à l'ensemble des cylindres de l'image. Or, ceci est impossible à vérifier sur les images tomographiques fournies. Il est donc nécessaire d'adapter la méthode à la réalité de terrain.

Dans les faits, la proximité de niveau de gris entre l'intérieur des fibres et la résine fait que la segmentation ne peut raisonnablement pas séparer parfaitement les deux phases, et certains voxels qui devraient être associés aux fibres se retrouvent étiquetés dans la résine, ce qui donne donc des structures de cylindres poreux, voire sectionnés à certains endroits.

Dans un tel cas, on se donne d'abord une valeur de tolérance $0 \leq \tau < 1$. Et on modifie l'ensemble G de la façon suivante :

$$G = \{(c_u c_v), (u, v) \in N^2 \text{ tel que } H_{u,v} \geq 1 - \tau\}$$

La tolérance permet d'accepter des structures légèrement creuses. La valeur est naturellement à choisir avec précaution, afin de ne pas accepter des droites qui ne seraient pas des génératrices.

Dans ce contexte, on appellera droite parfaite une droite vérifiant $H_{u,v} = 1$.

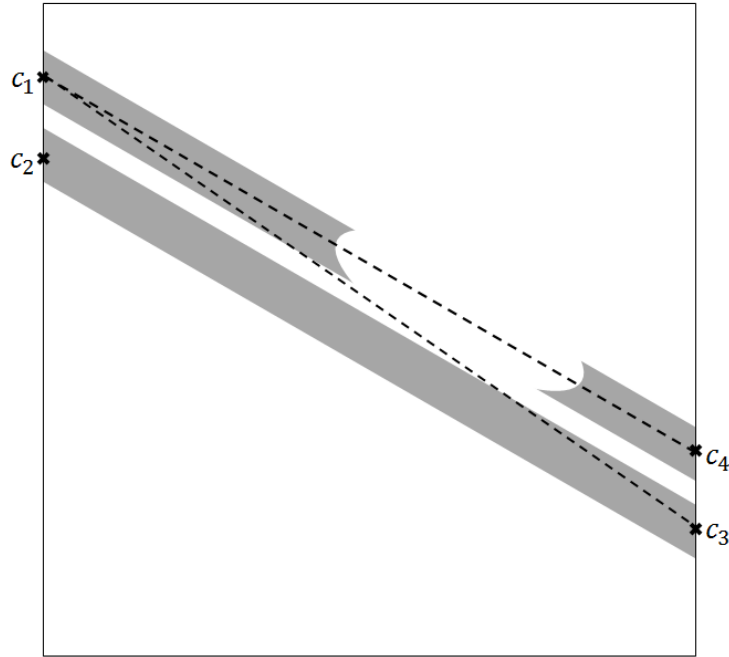


Figure 35 : Dans le cadre d'une fibre creuse (celle de génératrice (c_1c_4)) proche d'une autre fibre, il devient difficile pour l'algorithme simple de déterminer laquelle des deux droites (c_1c_4) ou (c_1c_3) correspond à une fibre réelle.

Malheureusement, dans les faits, certaines fibres se touchent localement, ce qui met à mal notre deuxième hypothèse, et par conséquent, si on augmente la tolérance suffisamment pour accepter les fibres creuses, il est très probable que des droites non génératrices soient acceptées aussi ailleurs sur l'image (Figure 35).

La méthode en une passe montre ici ses limites dans un cadre réel, et il devient donc nécessaire de considérer un algorithme itératif.

Malgré la présence de zones de contacts localisées entre certaines fibres, il reste très improbable d'obtenir une jonction suffisamment forte pour qu'une droite non-génératrice liant deux fibres proches rentre dans la seconde fibre sans avoir jamais quitté la première fibre auparavant, c'est-à-dire que la droite passe toujours localement dans $J^{-1}(0)$ entre les deux fibres.

De ce fait, même si deux fibres proches de génératrices respectives $(c_u c_v)$ et $(c_w c_t)$ sont creuses et se touchent localement, il est quasi-certain que les coefficients de H associés aux droites croisées $H_{u,w}$ et $H_{t,v}$ soient plus petits qu'au moins l'un des deux coefficients $H_{u,v}$ et $H_{w,t}$. Il serait donc possible de ne considérer que la droite la plus probable pour

chaque point par une comparaison des valeurs de H , ce qui empêcherait les doublons et de fait, empêcherait les droites non-génératrices entre deux fibres proches d'être détectées.

Toutefois, cette condition n'est encore pas suffisante. En effet, dans le cadre de deux fibres proches, une droite croisée non-génératrice passera environ 50 % de sa longueur dans la première fibre et autant dans la seconde. De fait, les porosités de la fibre se répercutent facilement sur la droite non-génératrice, qui est donc exclue de fait. Cependant, si on considère deux fibres assez creuses éloignées, séparées par un ensemble de fibres denses, un problème survient de nouveau (Figure 36). Une droite non-génératrice reliant ces deux fibres n'est que peu impactée par la porosité, ne passant que peu dans chacune d'entre elles. Pour la discriminer, il est donc nécessaire de voir que cette droite traverse d'autres fibres. De fait, il est nécessaire de modifier itérativement H en fonction des droites détectées précédemment pour empêcher le croisement d'autres fibres.

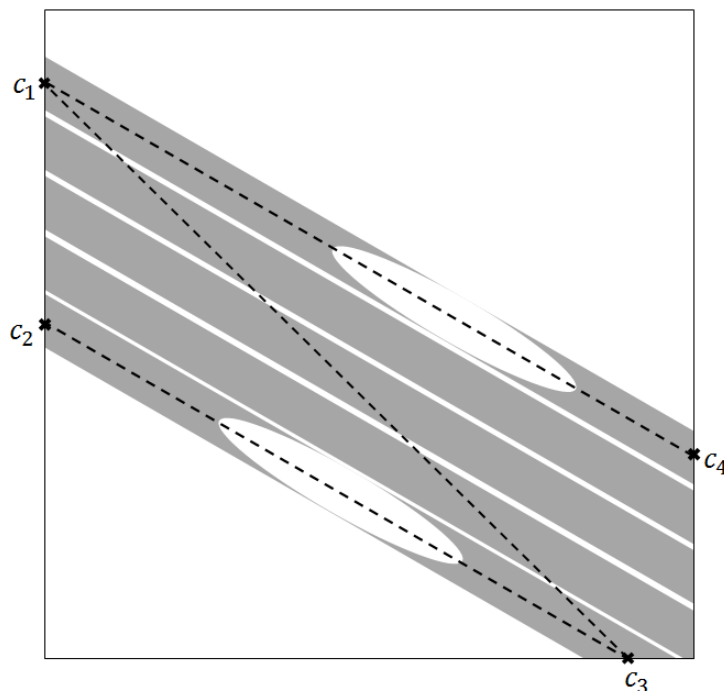


Figure 36 : La détection de la fibre maximale pour chaque centre ne suffit pas. Ici, des fibres remplissent le milieu entre c_1 et c_3 . De fait, au yeux de l'algorithme simple, (c_1c_3) est une fibre plus probable que (c_1c_4) .

Le fonctionnement de base de l'algorithme itératif résolvant le problème est le suivant pour l'itération n connaissant les résultats de $n - 1$:

1. Calcul de la carte de distance sur les faces de l'image ∂J^{n-1} .
2. Détermination des maximums locaux de cette carte de distance.

3. Calcul de la présence d'une droite ou non entre les paires de maximums locaux
4. Recherche de la droite la plus probable, i.e. celle réalisant le maximum de \mathbf{H}^n .
5. Si le maximum de \mathbf{H}^n est supérieur à $1 - \tau$, on pose \mathbf{J}^n comme \mathbf{J}^{n-1} à laquelle on supprime le cylindre déduit de cette droite, puis retour à l'étape 1. Sinon, l'algorithme s'arrête.

L'algorithme construit de cette manière supprime au fur et à mesure les fibres déjà considérées et par extension leurs points associés sur la carte de distance. De cette manière, on a donc corrigé les erreurs potentielles de l'algorithme en une seule passe (Figure 37).

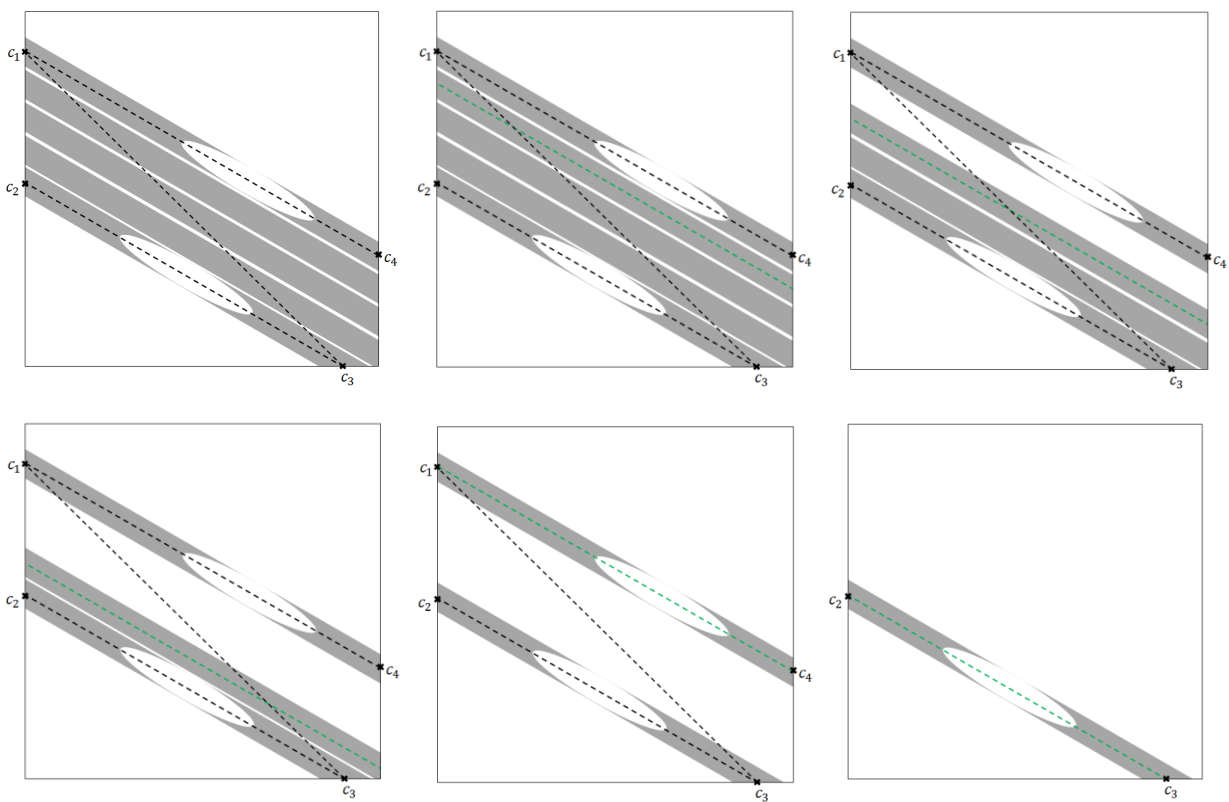


Figure 37 : Détection itérative avec suppression. Ici, avec une détection en une passe, une fausse droite serait détectée. Dans le cadre de la détection itérative, une meilleure droite est détectée (haut, centre), puis supprimée (haut droite), et ainsi de suite. A la fin, la fausse droite devient invraisemblable (bas, centre).

Cependant, d'un point de vue de la complexité, la méthode n'est pas très performante. Dans l'algorithme itératif construit ici, la matrice \mathbf{H} doit être recalculée après chaque fibre détectée. Le nombre de fibre évoluant en $O(L^2)$, et le calcul de \mathbf{H} étant en $O(L^5)$, on obtient donc ici un algorithme en $O(L^7)$, ce qui est pire que les autres méthodes explicitées auparavant.

La terminaison de l'algorithme est aisée à démontrer, le nombre de voxels considérés $(\text{card } E(\mathbf{J}_+^n))_n$ étant une suite strictement décroissante de \mathbb{N} si la tolérance est strictement inférieure à 1.

En pratique, on cherche bien sûr un moyen de ne pas avoir à recalculer la matrice \mathbf{H} après chaque itération. Pour cela, on va compter sur la qualité de l'image et de la segmentation. En effet, après observation des images, on peut admettre que, bien que certaines fibres soient poreuses, celles-ci restent toujours d'un seul tenant, c'est-à-dire que les deux points d'intersection de la génératrice avec les bords sont reliés de manière connexe dans $E(\mathbf{J}_+)$.

Par conséquent, on va se servir de la connexité dans \mathbb{N}^3 pour améliorer l'algorithme. On commence par rappeler les définitions et propriétés importantes sur la connexité :

Définition : Soit (E, \mathcal{U}) un espace topologique. On dit que E est connexe si les seules parties de E simultanément ouvertes et fermées sont E et \emptyset . Une partie F de E est connexe si l'espace topologique induit (F, \mathcal{U}^F) l'est aussi.

Proposition : Une partie F de (E, \mathcal{U}) est connexe si et seulement si pour toute paire d'ouverts disjoints U et V telle que $F \subset (U \cup V)$, on a $F \subset U$ ou $F \subset V$.

Par la suite, on utilisera plutôt la notion de connexité par arcs, plus intuitive que la notion de connexité simple définie ci-dessus.

Définition : Soit $E \subset \mathbb{R}^n$, et $F \subset E$ une partie quelconque de E . On dit que F est connexe par arcs si et seulement si, pour tout doublet $(a, b) \in F^2$, il existe une fonction $\gamma : [0 ; 1] \rightarrow F$ continue telle que

$$\gamma(0) = a, \gamma(1) = b \text{ et } \forall t \in [0 ; 1], \gamma(t) \in F$$

c'est-à-dire s'il existe un chemin continu de F conduisant de a à b .

Propriété : Un espace connexe par arcs est connexe.

On s'intéressera maintenant à étendre la connexité par arcs (que l'on appellera connexité par abus de langage) sur un espace borné discret $E \subset \mathbb{N}^n$. On se donne un système de voisinage \mathcal{N} symétrique, i.e. si $x \in \mathcal{N}(y)$, alors $y \in \mathcal{N}(x)$.

Définition : Un chemin de $E \subset \mathbb{N}^n$ est une suite finie $(a_i)_{0 \leq i < k}$ d'éléments de E tel que $\forall i \in [0 : k - 2], a_{i+1} \in \mathcal{N}(a_i)$.

Ainsi, un chemin sur un espace borné discret est une suite finie de voisins deux à deux.

Définition : Un sous-ensemble $F \subset E$ est dit connexe pour le système de voisinage \mathcal{N} si et seulement si pour tout doublet $(x, y) \in F^2$, il existe un chemin $(a_i)_{0 \leq i < k}$ de F tel que $a_0 = x$ et $a_{k-1} = y$.

Dans notre cas, on se placera dans \mathbb{N}^3 et on considèrera trois systèmes de voisinages possibles, le 6-voisinage, le 18-voisinage et le 26-voisinage.

Définition : Soit $a \in E$. On appelle composante connexe de a le plus grand connexe de E contenant a .

On constatera qu'appartenir à une même composante connexe est une relation d'équivalence. On peut ainsi définir des classes d'équivalences sur E qui correspondent aux composantes connexes.

La méthode la plus simple conceptuellement pour calculer les composantes connexes d'un ensemble $E \subset \mathbb{N}^3$ est :

1. Prendre un point $a \in E$ non considéré.
2. Calculer sa composante connexe. Tous les points de la composante connexe sont alors considérés.
3. S'il reste encore des points de E non considérés, repasser à l'étape 1.

Le calcul de la composante connexe d'un point $a \in E$, pour sa part, s'effectue alors comme suit :

1. On considère l'ensemble $C = \{a\}$.
2. On note $D = (\bigcup_{x \in C} \mathcal{N}_E(x)) \setminus C$, c'est-à-dire que D est l'ensemble des points de E voisins d'un point de C .
3. Si $D \neq \emptyset$, alors on considère l'ensemble $C := C \cup D$, et on repasse à l'étape 2.
4. Sinon, on arrête ici. C est alors la composante connexe de a .

Il est évident par construction que l'ensemble C déterminé ici est connexe. De même, il est le plus grand connexe, car sinon, on peut noter C_0 ce plus grand connexe, on a donc $C \subset C_0$. Il existe donc un chemin reliant un point de C à un point de $C_0 \setminus C$, et par conséquent il existe un point de C voisin d'un point de $C_0 \setminus C$ sur le chemin, donc ce point aurait dû être considéré, sauf si $C_0 \setminus C = \emptyset$.

Cette méthode permet donc de calculer les composantes connexes de l'image. Cependant, sa complexité n'est pas très intéressante car elle demande une propagation de voisinage dans tous les sens, c'est-à-dire une recherche de voisinage pour chaque itération.

En pratique, les différentes techniques utilisées pour la détection de composantes connexes à partir d'un ensemble $F \subset E = \prod_{i < 3} [0 : L_i]$ sont :

- Les algorithmes à passes multiples (Haralick, 1981), qui propagent les informations de voisinage par des passages multiples de l'image dans tous les sens séquentiellement jusqu'à obtenir un état d'équilibre final correspondant aux composantes connexes. La rapidité de ces méthodes dépend très fortement de la forme de l'ensemble considéré. La méthode naïve présentée ci-dessus s'apparente aux méthodes à passe multiples.
- Les algorithmes à double passe (Rosenfeld & Pfaltz, 1966) (Rosenfeld, 1970) (Lumia, 1983) (Shirai, 1987) (He, et al., 2009), qui commencent par effectuer un passage simple calculant des groupes connexes non maximaux, puis un deuxième passage qui résout les équivalences entre groupes et simplifie le résultat jusqu'aux composantes connexes.
- Les algorithmes de détections de contours (Rosenfeld, 1970) (Chang, et al., 2004), qui, au lieu de rechercher l'ensemble de chaque composante connexe, en recherche les frontières.

Ici, on utilisera des algorithmes à double passe, en particulier celui développé par He et al. (He, et al., 2009). La complexité de cette méthode variant peu selon les images, la robustesse de celle-ci nous assure de majorer la complexité dans tous les cas par une valeur relativement faible.

Ainsi, si l'on modifie le calcul de la matrice H de la manière suivante :

$$H_{u,v} = \begin{cases} \frac{\text{card}(A_{u,v} \cap E(\mathbf{J}_+))}{\text{card}A_{u,v}} & \text{si } c_u \text{ et } c_v \text{ sont reliés de manière connexe dans } E(\mathbf{J}_+) \\ 0 & \text{sinon} \end{cases}$$

on ne perd aucune information importante car si $(c_u c_v)$ est la génératrice, alors les deux points sont reliés de manière connexe d'après le présupposé, et donc $H_{u,v}$ ne change pas.

On notera $(\mathcal{M}_k)_{k \leq K}$ les composantes connexes de $E(\mathbf{J}_+)$, et, pour tout $u \leq N$, on notera $\mathcal{M}(u)$ la composante connexe contenant c_u . On peut donc réécrire :

$$H_{u,v} = \frac{\text{card}(A_{u,v} \cap E(\mathbf{J}_+))}{\text{card}A_{u,v}} \cdot \delta(\mathcal{M}(u), \mathcal{M}(v))$$

De plus, la fibre étant « d'un seul tenant », elle fait donc partie de la même composante connexe. Ainsi, on a :

$$H_{u,v} = \frac{\text{card}(A_{u,v} \cap \mathcal{M}(u))}{\text{card}A_{u,v}} \cdot \delta(\mathcal{M}(u), \mathcal{M}(v))$$

Il est à noter que le nombre de points de $E(\mathbf{J}_+^n)$ étant décroissant au cours des itérations, on peut affirmer que :

$$\forall n_0 \in \mathbb{N}, \forall n > n_0, \forall (u, v) \in [1 : N]^2, \quad H_{u,v}^n \leq H_{u,v}^{n_0}$$

De fait, si $H_{u,v}^{n_0} < 1 - \tau$, alors il est inutile de considérer et de recalculer $H_{u,v}^n$ pour tout $n > n_0$. Ainsi, étant donné que $H_{u,v}^n$ est non nul uniquement si $\mathcal{M}^n(u) = \mathcal{M}^n(v)$, on ne doit donc considérer que les doublets $(c_u c_v)$ reliés de manière connexe et pour lesquels $H_{u,v}^{n-1} \geq 1 - \tau$. La complexité du calcul de \mathbf{H} vérifie par conséquent (avec $\mathcal{C} = (c_k)_{k \leq N}$) :

$$C_{\text{calcul de } \mathbf{H}} = O\left(L \sum_{k \leq K} \text{card}(\mathcal{C} \cap \mathcal{M}_k)^2\right)$$

En théorie, le cardinal de $\mathcal{C} \cap \mathcal{M}_k$ est une donnée dépendant de la structure de l'image, constante avec L , et K , par contre, dépend quadratiquement de L tout comme le nombre de fils. Ainsi, on a finalement :

$$C_{\text{calcul de } \mathbf{H}} = O(L^3)$$

La constante de temps associée est cependant potentiellement importante, la détermination théorique de $\mathbb{E}(\text{card}(\mathcal{C} \cap \mathcal{M}_k^n))$ est effectuée dans l'Annexe 1. Dans le cas parfait où toutes les fibres sont isolées, cette constante est de 1. Dans le pire des cas, elle est reliée au nombre de fibres par fil.

On remarquera de plus que dans le cadre de l'algorithme itératif, les seuls points supprimés sont ceux issus du cylindre déduit. De fait, ils font donc tous partie d'une unique composante connexe, le reste ne changeant pas. Ainsi, si on détecte, à l'itération $n - 1$, une droite $(c_u c_v)$ donnée, alors on a pour tout $(s, t) \in [1 : N]^2$ tel que $\mathcal{M}^{n-1}(s) \neq \mathcal{M}^{n-1}(u)$ ou $\mathcal{M}^{n-1}(t) \neq \mathcal{M}^{n-1}(u)$:

$$H_{s,t}^n = H_{s,t}^{n-1}$$

Ainsi, lorsque l'on passe de l'itération $n - 1$ à l'itération n , il n'est nécessaire de recalculer \mathbf{H}^n que sur l'ensemble de la composante connexe de la dernière fibre détectée, ainsi le recalcul de \mathbf{H}^n a une complexité de :

$$C_{\text{recalcul de } \mathbf{H}} = O(L)$$

L'algorithme qu'on peut alors construire est le suivant :

- A l'itération 1 :
 1. Calcul de la carte de distance sur les faces de l'image $\partial \mathbf{J}^0$.
 2. Détermination des maximums locaux de cette carte de distance.
 3. Calcul des composantes connexes.
 4. Calcul de \mathbf{H}^1 .
 5. Recherche de la droite la plus probable, i.e. celle réalisant le maximum de \mathbf{H}^1 .
 6. Si le maximum de \mathbf{H}^1 est supérieur à $1 - \tau$, on pose \mathbf{J}^1 comme \mathbf{J}^0 à laquelle on supprime le cylindre déduit de cette droite, puis on rentre dans la boucle itérative. Sinon, l'algorithme s'arrête.
- A toutes les autres itérations :
 1. Calcul des composantes connexes de \mathbf{J}^{n-1} à partir de celles de \mathbf{J}^{n-2} . Elles restent toutes identiques exceptée celle dont est issue la droite considérée à la dernière itération.

2. Calcul de \mathbf{H}^n en fonction de la droite détectée à l'itération $n - 1$ et de \mathbf{H}^{n-1} .
3. Recherche de la droite la plus probable, i.e. celle réalisant le maximum de \mathbf{H}^n .
4. Si le maximum de \mathbf{H}^n est supérieur à $1 - \tau$, on pose \mathbf{J}^n comme \mathbf{J}^{n-1} à laquelle on supprime le cylindre déduit de cette droite, puis retour à l'étape 1. Sinon, l'algorithme s'arrête.

On s'intéresse donc à la complexité de l'algorithme de calcul des composantes connexes utilisé ici (He, et al., 2009). Pour une image de taille L^3 où tout est à calculer, la complexité dans le pire des cas vérifie :

$$C_{\text{connexe}}^{\max} = O(L^3)$$

Cependant, dans notre cas, seulement un recalcul sur une seule des composantes connexes est nécessaire. La composante connexe est un ensemble de fibres dont le nombre est fixé. Il n'y a alors que $O(L)$ points à reconsidérer. La complexité du recalcul est donc elle aussi en $O(L)$ dans le pire des cas.

On a donc :

- Une première itération où tout est à calculer. On a donc un $O(L^3)$ sur la détermination des composantes connexes, sur le calcul de la carte de distance, et aussi sur le calcul de \mathbf{H} .
- Une succession de M itérations, avec $M = O(L^2)$, pour lesquelles on se contente de mettre à jour les données des itérations précédentes. On a une complexité en $O(L)$ pour la mise à jour des composantes connexes comme pour le calcul de \mathbf{H} .

Finalement, on a construit un algorithme géométrique de détection de fibres en complexité linéaire avec la taille de l'image, i.e. en $O(L^3)$.

Ceci est tout de même à nuancer pour deux raisons :

- Malgré le fait que les calculs proprement dits ne se font que sur une partie de l'image de cardinal en $O(L)$, le recalcul des composantes connexes demande au moins un parcours complet de celle-ci à moins de réussir à contraindre a priori la

zone de chaque composante connexe. Ainsi, on pourrait aussi considérer l'algorithme comme un $O(L^5)$ avec une très petite constante de temps.

- Dans le cadre des images étudiées ici, en particulier celles à haute résolution, la taille des images n'est pas toujours assez importante pour que les fils soient entièrement compris dans celles-ci. On observe donc un comportement où le nombre de composantes connexes n'évolue pas avec la taille, alors que le nombre de fibres par composante évolue encore en $O(L^2)$. On observe donc en pratique un comportement en $O(L^5)$ jusqu'à ce que le volume considéré soit représentatif du matériau, à partir duquel on observe un comportement $O(L^3)$ avec grande constante de temps. En particulier, sur les images à résolution de $0,7 \mu\text{m}/\text{vx}$, on ne possède pas d'images de taille assez importante pour voir apparaître le comportement en $O(L^3)$.

D'autres améliorations de la méthode sont encore possibles : on utilisera le fait que les fils sont constitués de fibres d'orientation similaire pour encore réduire le champ de recherche de l'algorithme. Pour cela, on considèrera d'abord que :

1. Tant que des fibres vérifient $H_{u,v}^n = 1$, on ne contraindra pas plus la détection de celles-ci. On notera $(d_v)_v$ l'ensemble des droites détectées jusque-là.
2. Une fois que cette propriété n'est plus vérifiée, on supposera qu'il existe au moins une fibre déjà détectée pour chaque fil. Cette supposition sera validée par l'expérience.
3. On se donne un angle α de déviation maximale, et, à partir du moment où la propriété 1 n'est plus vérifiée, chaque droite $(c_u c_v)$ se verra comparée à $(d_v)_v$. Si $\forall v, ((c_u c_v), d_v) > \alpha$, alors on pose $H_{u,v}^n = 0$.

Cette méthode permet de nouveau un gain de temps conséquent dès que les droites parfaites ont toutes été détectées, en éliminant un grand nombre de doublets possibles. On constate que dans ce cas, le nombre de doublets à vérifier pour chaque point est directement réduit par les orientations (normalement, le nombre d'orientations est égal au nombre de fils, c'est-à-dire en $O(L^2)$), mais la composante connexe ne propose cependant pas de doublets dans toutes ces directions mais dans un nombre potentiellement réduit d'entre elles (Figure 38). En particulier, dans le cadre des images à haute résolution explicite

précédemment, le nombre de fils se comporte en $O(1)$, ainsi le nombre de doublets à vérifier pour chaque point devient alors un $O(1)$ aussi grâce aux contraintes d'orientations. De fait, une fois passé la détection des fibres « parfaites », les connaissances sur l'image nous permettent d'effectuer la suite de l'algorithme en une complexité de l'ordre de $O(L^3)$.

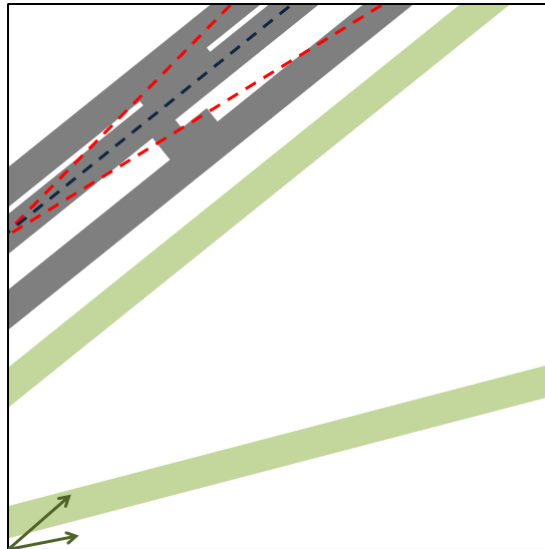


Figure 38 : Figure explicative de la restriction en orientation. Ici, deux droites parfaites (en vert) ont été détectées. La contrainte oblige alors toutes les droites suivantes à s'approcher de leurs deux orientations (représentées par les flèches vertes en bas à gauche). Ici, par exemple, malgré la connexité, les droites rouges sont directement exclues car trop éloignées des orientations acceptables, à la différence de la droite bleue.

Il est cependant à noter que ce calcul de déviation doit se faire sur toutes les droites acceptables dès lors qu'il n'y a plus de droites parfaites. Ainsi, une itération « lourde » est nécessaire pour faire le passage du cadre « parfait » au cadre « normal ».

La contrainte sur les orientations nous permet donc, de manière non supervisée, de réduire la complexité et le temps de calcul d'une partie de l'algorithme. Cependant, dans sa globalité, celui-ci conserve la même complexité algorithmique.

Dans un aspect d'amélioration purement temporel, il est aussi possible de faire progresser la tolérance τ au cours de l'exécution jusqu'à la tolérance maximale τ_{max} considérée. Pour ce faire, on prend une valeur fixée et on ne la modifie que lorsqu'il n'y a plus de doublet admissible. Ceci nécessite un recalcul lorsque l'on actualise τ , mais permet de limiter le nombre de doublets possibles simultanément. Il est possible, en fonction du temps moyen que prend une itération sans recalcul et celui avec, d'optimiser le choix du nouveau τ pour limiter au maximum la durée de l'algorithme.

Dans les faits, on recalculera l'ensemble des éléments considérés lors de ces itérations, ce qui permettra à l'algorithme de s'actualiser en profondeur, la suppression des cylindres permettant de modifier les faces de l'image, et pouvant donc faire apparaître des nouveaux centres à considérer.

L'algorithme développé ici peut être modifié pour améliorer certains points en perdant sur d'autres ensuite. Une méthode rapide a été développée, procédant au calcul en un temps très court en échange de légères erreurs et pertes sur le résultat de la méthode. Cependant, si le but visé est la recherche de fils par exemple, pour laquelle la quantité de fibres détectées à l'intérieur d'un fil importe peu tant que l'enveloppe convexe reste la même, alors les pertes seront négligeables devant le gain de temps. Cette modification s'effectue sur la détermination des centres sur les faces. Dans le cas classique, cette détermination se fait grâce à une carte de distance, et chaque maximum correspond alors à un centre. Cependant, le nombre de centres se révèle alors relativement élevé, ce qui nuit à la taille de la matrice H et par conséquent au temps de calcul de cette matrice, facteur limitant de l'algorithme.

Une autre idée serait donc d'appliquer sur l'image des faces l'algorithme de détermination des régions connexes, et de considérer ensuite les centres de gravité de chaque composante connexe. Ce changement occasionne quelques différences avec la précédente méthode :

- Dans le cas de fibres pleines et isolées, le centre considéré est le même que pour la méthode de carte de distance (Figure 39).

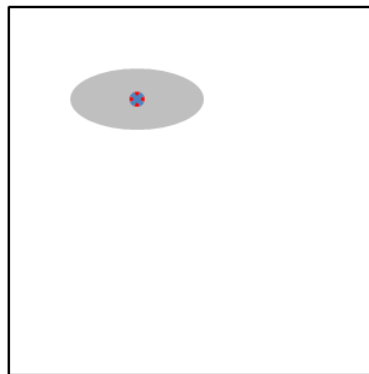


Figure 39 : Coupe d'une face intersectée par une fibre. Dans ce cas, les centres calculés respectivement par la carte de distance et par la méthode de composantes connexes sont identiques.

- Plus de robustesse face aux fibres creuses sur les faces de l'image (Figure 40). Dans ce cadre-là, les centres peuvent tout de même être proches d'un point de la génératrice, même si ce point n'est pas inclus dans la bonne région (i.e. si le point se trouve dans la classe associée à la résine).

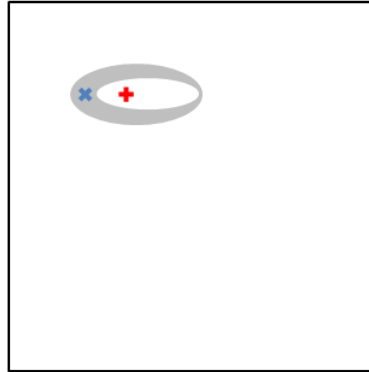


Figure 40 : Dans le cas d'une fibre creuse à son intersection avec la face, le centre de la carte de distance (en bleu) est moins représentatif de la fibre que celui calculé par les composantes connexes (en rouge).

- Moins de robustesse dans le cas de fibres mitoyennes (Figure 41). En effet, si les fibres se touchent sur les faces de l'image, alors elles ne se verront associer qu'un centre qui ne coïncide avec aucun des vrais centres des fibres. Même si le recalcul au fil des itérations va ensuite faire apparaître les autres fibres, des décalages d'angles seront induits. Dans le cas où uniquement deux fibres se touchent, ce décalage n'est pas très important (Figure 42 et Figure 43), mais si la composante connexe en contient un grand nombre, le résultat risque de se révéler très éloigné de la réalité.

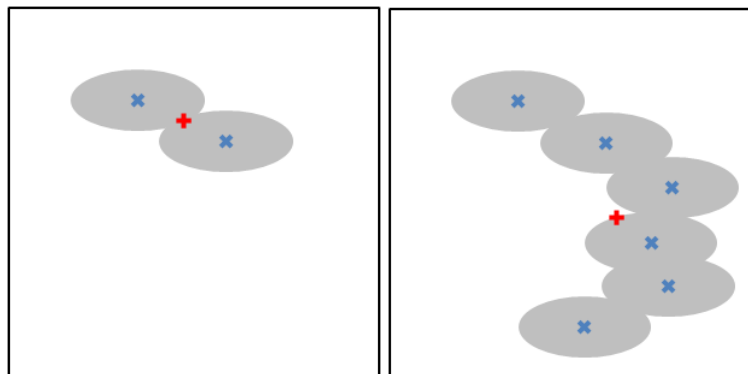


Figure 41 : Dans le cas de plusieurs fibres voisines sur la face, la carte de distance (en bleu) renvoie le bon nombre de centres et ceux-ci sont très proches de la génératrice, à l'opposé de la méthode des composantes connexes qui ne renvoie qu'un seul centre aberrant (en rouge).

Heureusement, le cas de très nombreuses fibres qui seraient connectées sur une face est improbable pour les images à haute résolution, ce qui rend la méthode viable. Par

contre, sur celles à basse résolution, les fibres sont beaucoup trop proches, et la méthode est inutilisable.

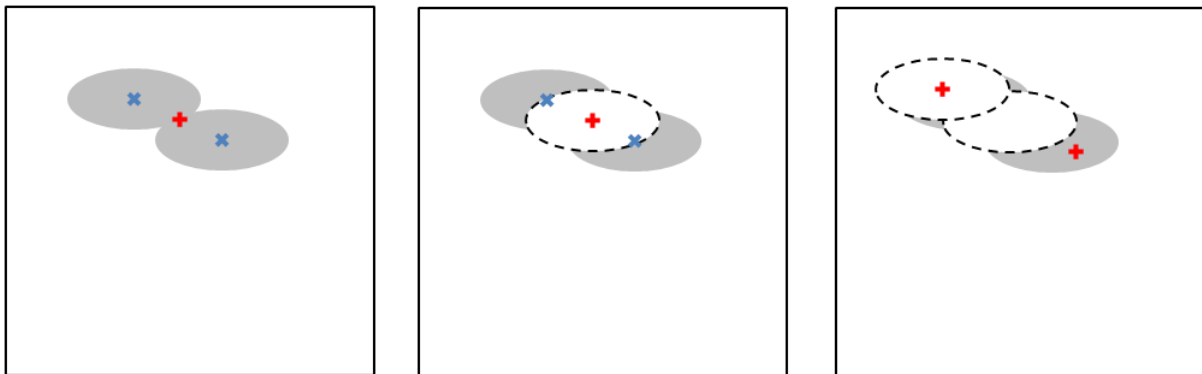


Figure 42 : Schéma de l'erreur de détection de l'algorithme de composantes connexes pour deux fibres voisines. Les deux fibres détectées sont finalement décalées.

Tout l'intérêt de la méthode porte sur la réduction du nombre de points à considérer, en particulier parce que la forme légèrement déformée de certaines fibres cause des problèmes de maximums multiples de la carte de distance à l'intérieur des fibres, ce qui surcharge la matrice H .

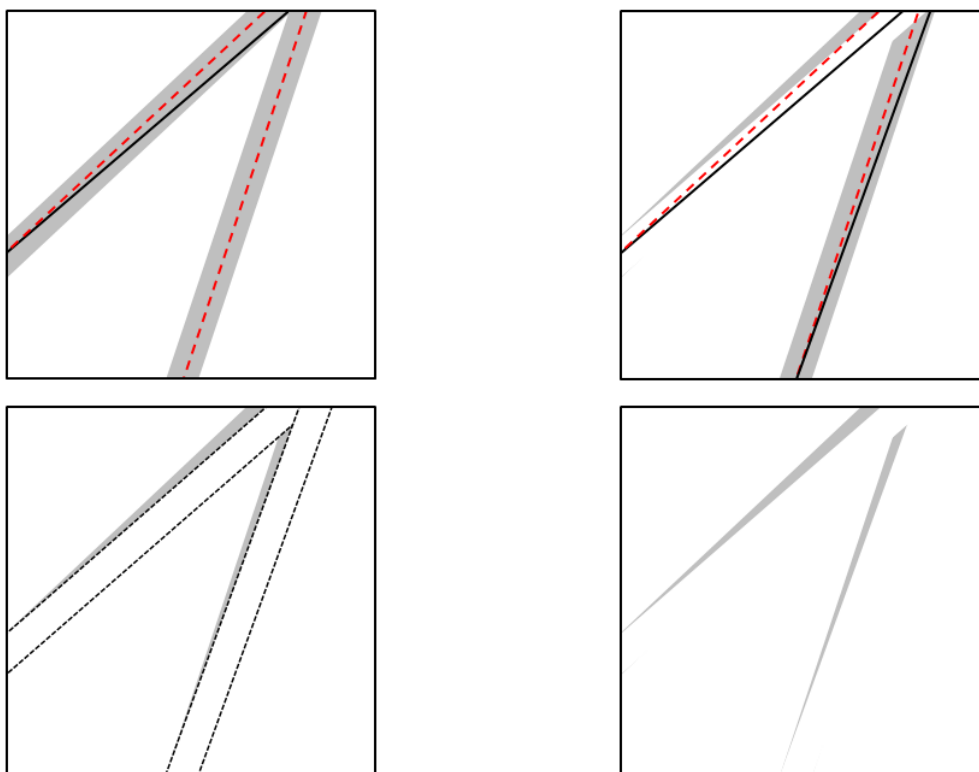


Figure 43 : Vue de coupe des erreurs induites par la détection de deux fibres voisines avec l'algorithme des composantes connexes. Après la détection des deux fibres décalées (en bas à gauche), il reste finalement un résidu de détection (en bas à droite).

Les erreurs maximales d'angles induites par la méthode rapide ont lieu pour les droites les plus courtes, qui sont déjà les plus sensibles aux erreurs de par leurs longueurs proches du rayon des fibres. Dans n'importe quel cas, on aura tendance à ne pas considérer les lignes dont la longueur est inférieure à au moins kr avec r le rayon des fibres et $k \geq 4$ une constante, afin d'obtenir un rapport supérieur à 2 entre diamètre et longueur. On considèrera, pour les images étudiées ici, des valeurs de k de 4 ou de 8. Dans un tel cas, l'écart maximal en angle est de $\alpha = \text{atan}\left(\frac{1}{k}\right)$. Dans le cas où $k = 4$, on a donc $\alpha \sim \frac{\pi}{12}$, et pour $k = 8$, on a donc $\alpha \sim \frac{\pi}{25}$ (Figure 44).

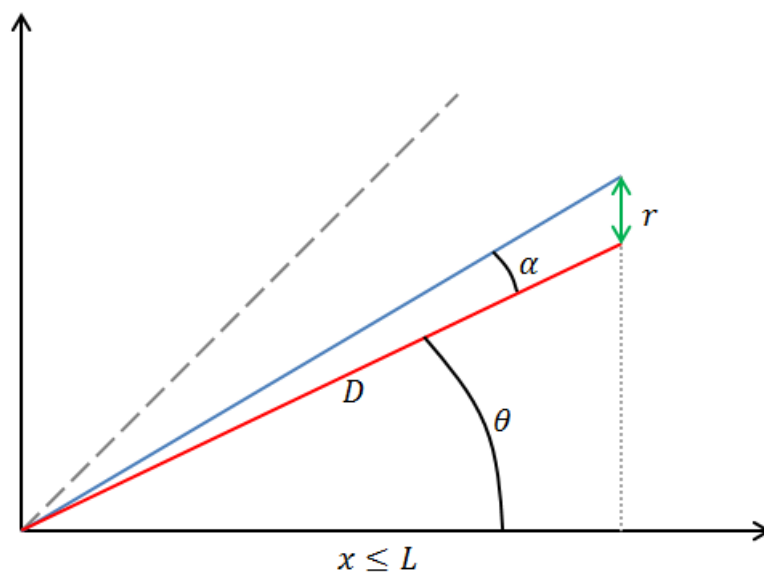


Figure 44 : Schéma explicatif des erreurs d'orientations dues à la méthode des composantes connexes.

Comme l'algorithme se sert des droites parfaites pour contraindre la détection et gagner du temps, il est par conséquent nécessaire que la tolérance sur la direction des droites soit suffisante pour accepter les droites déviées par la méthode.

2.4.3. Association des fibres

Les droites détectées sont supposées être représentées par le jeu de paramètres $(p_1, p_2, (\theta, \phi))$ représentant les deux points d'intersection de la fibre avec les bords de l'image, et les deux angles d'Euler qui peuvent en être déduits. Une classification par K-means est effectuée sur les angles (θ, ϕ) . On présente donc ici les méthodes de classification par K-means.

Avant, il est utile de noter que la classification, ou *clustering* est un problème intrinsèquement subjectif, l'application finale utilisant la classification modelant fortement, selon ses besoins, les méthodes de classifications elles-même.

Ainsi, le problème de classification est mal posé (Figure 45 et Figure 46), et relève complètement de ce que l'on considère comme une classe, et de la notion de distance entre deux éléments caractérisant leur proximité. Les différents défis de la classification ont été énoncés par Jain et Dubes (Jaine & Dubes, 1998), puis repris par Jain (Jain, 2010). Ces défis portent sur la méthode de classification, la représentation des données (et donc le choix des données significatives) ou le choix du nombre de classes, entre autres.

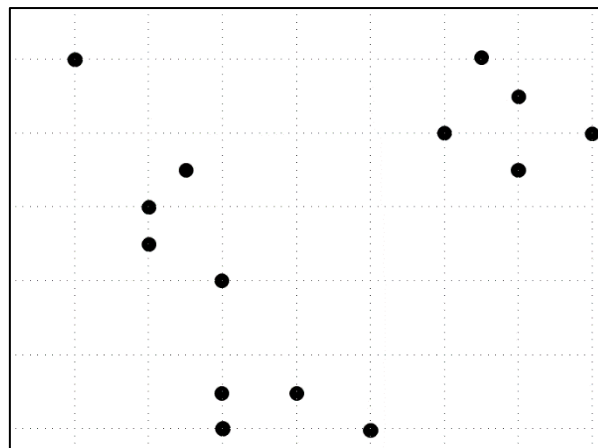


Figure 45 : Image exemple pour la classification par K-Means. Chaque point noir correspond à un point à classifier.

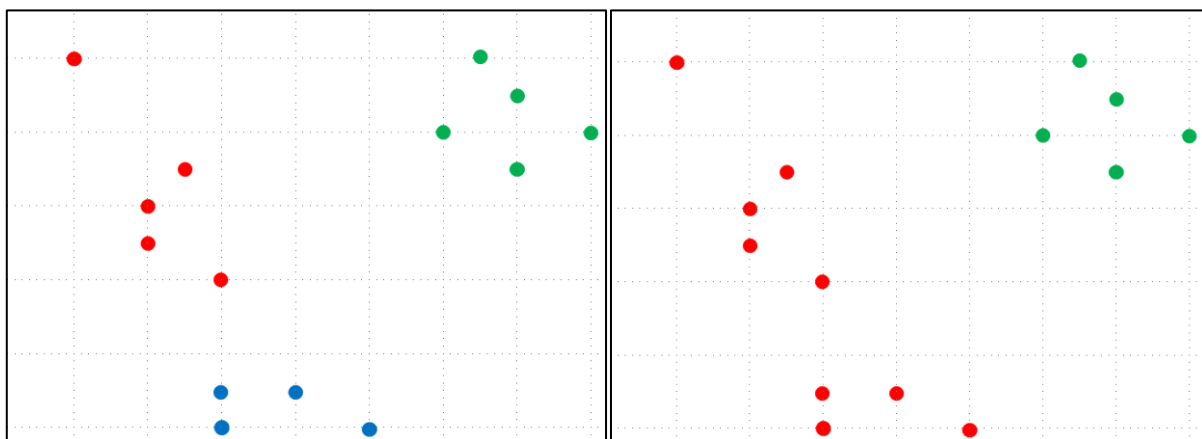


Figure 46 : Deux classifications possibles. Le problème est mal posé, il n'est pas possible de déterminer selon des critères objectifs laquelle des deux est meilleure.

Un des algorithmes les plus anciens, et pourtant l'un des plus courants, est l'algorithme des K-means. Cet algorithme, proposé par Lloyd en 1957 mais seulement publié dans sa version originale bien plus tard (Lloyd, 1982) et étendu par MacQueen (MacQueen,

1967), repose sur la minimisation de la somme des distances à des points centraux de chaque cluster.

Soit $\{x_i\}_{i \in [1 : N]} \subset \mathbb{R}^n$ un ensemble fini de points de \mathbb{R}^n que l'on désire partitionner en K classes $\{c_j\}_{j \in [1 : K]}$. L'algorithme des K-means a pour but de minimiser le carré de la distance entre les points appartenant à chaque classe et le centre de ces points. Ainsi, si on note

$$m_j = \frac{1}{\text{card}(c_j)} \sum_{x_i \in c_j} x_i$$

alors le but du K-means est de minimiser la fonctionnelle suivante :

$$\Gamma(\{c_j\}_{j \in [1 : K]}) = \sum_{j=1}^K \sum_{x_i \in c_j} \|x_i - m_j\|^2$$

Il est évident que pour minimiser cette fonctionnelle, chaque point doit être associé au cluster dont le centre est le plus proche. Ainsi, chaque point appartient à la cellule de Voronoï du centre de son cluster (Figure 47). Malheureusement, les centres dépendent des points associés et l'association des points dépend elle-même des centres. Ce problème de minimisation est donc, de manière prévisible, un problème NP-complet (Drineas, et al., 1999).

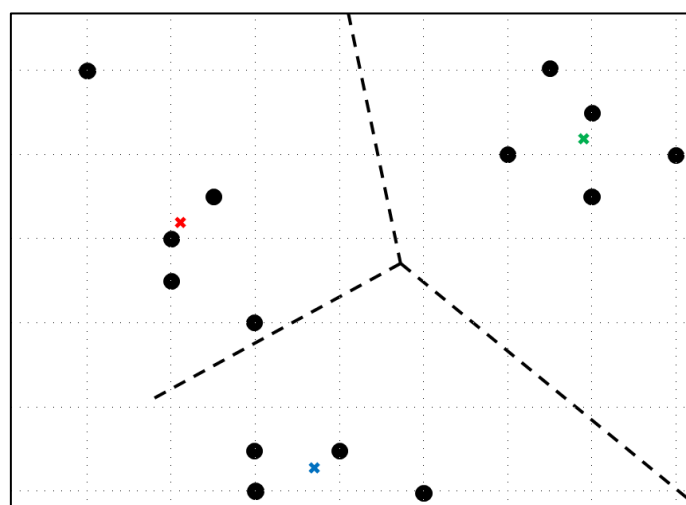


Figure 47 : Classification par K-Means donnant le résultat de la Figure 46 à gauche. Les croix correspondent aux centres des classes.

De manière pragmatique, il n'est donc pas possible de calculer à coup sûr la solution minimale en un temps raisonnable, et la façon dont l'algorithme des K-means traite les données ne permet que d'obtenir une solution correspondant à un minimum local de la fonctionnelle considérée. Cette méthode utilise un principe de convergence simple (Jain, 2010), en partant d'un clustering initial $\{c_j^0\}_{j \in [1:K]}$:

- Calculer le point moyen pour chaque cluster m_j .
- Recalculer les clusters en mettant chaque point dans la classe dont le centre est le plus proche.
- Répéter ces deux phases tant que les points moyens sont modifiés.

Les clusters finissent alors par se retrouver dans une configuration consistant en un minimum local de la fonction d'écart quadratique. Cette configuration finale dépend donc beaucoup du choix du clustering initial $\{c_j^0\}_{j \in [1:K]}$ (Figure 48). Une discussion sur le choix des clusters initiaux a été effectuée par Bradley et Fayyad (Bradley & Fayyad, 1998), qui proposent une solution impliquant du *clustering de cluster*, fondée sur une méthode de noyaux (la méthode des noyaux pouvant être utilisée en tant que telle pour la classification, voir (Ben-Hur, et al., 2001) (Camastra & Verri, 2005)).

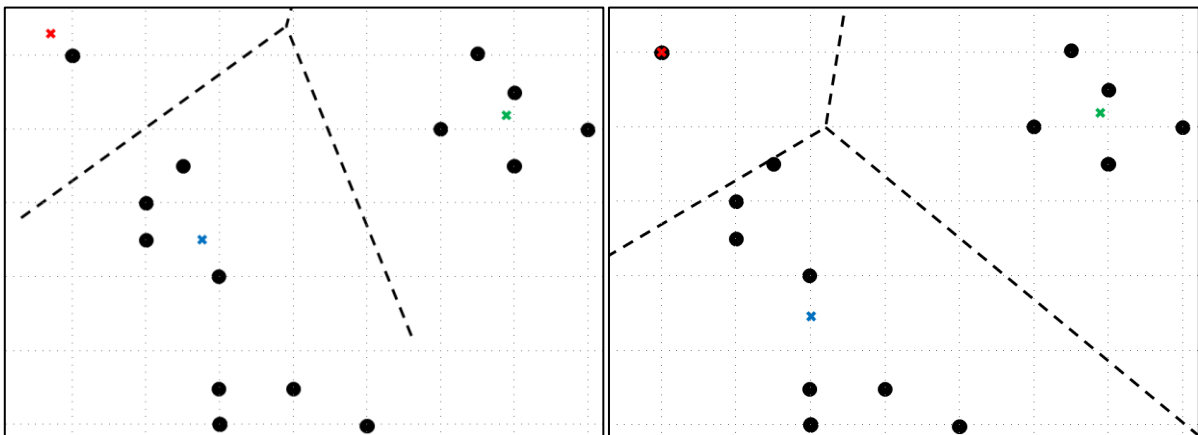


Figure 48 : Autre classification possible par méthode des K-Means. Si l'on prend la répartition initiale des centres comme sur l'image de gauche, le résultat final est celui représenté à droite, qui ne correspond pas au même minimum que sur la Figure 47.

On commencera par définir sur l'ensemble de départ une densité de probabilité d'avoir un point x faisant partie des données à classifier. Cette densité est estimée en fonction des données fournies (Scott, 1992), puis on recherchera les maximums locaux de la

fonction densité, qui donneront alors les centres initiaux pour le clustering. Cependant, la recherche des maximums comme le calcul de densité sont des opérations longues, rendant la méthode peu attractive en elle-même sur des jeux de données importants.

Bradley et Fayyad (Bradley & Fayyad, 1998) proposent donc d'effectuer cette initialisation sur des sous-ensembles de l'ensemble de données à classifier, afin de réduire le temps de calcul. Cependant, les points des extrémités des clusters peuvent alors avoir, selon le tirage des points, une importance décuplée, qui mène alors à des mauvaises initialisations. Une correction consiste donc à effectuer de nombreuses initialisations sur des sous-ensembles, puis de classifier l'ensemble des initialisations (cette classification étant plus simple que celle des données totales, sinon l'intérêt de la méthode serait nul) afin de trouver une valeur de départ valide.

Un deuxième paramètre important à fixer est le nombre de clusters K . Le principe d'initialisation par les noyaux peut déterminer de lui-même le nombre de clusters suivant le nombre de maximums locaux de la fonction de densité (et leur valeur afin de contrôler ce nombre), d'autres méthodes d'estimations existant aussi (Tibshirani, et al., 2001).

L'algorithme des K-means, par ses résultats robustes, sa simplicité et son côté assez général, sera utilisé par la suite, d'autant plus que des méthodes efficaces ont depuis été présentées, utilisant une structure d'arbre par exemple (Kanungo, et al., 2002). D'autres méthodes de classification et une discussion peuvent être trouvées dans (Jain, 2010).

Dans notre cas où la classification s'effectue en (θ, ϕ) , une distance euclidienne classique ne rendrait pas bien compte de l'aspect tridimensionnel des droites. En effet, comme les angles servent à repérer une orientation, c'est-à-dire un point sur la sphère unité, une classification avec une distance 2D classique revient à l'effectuer sur une projection cartographique des points représentant les orientations.

Une des premières idées est de garder la distance euclidienne, mais de passer en trois dimensions en ne représentant plus une droite par ses angles (θ, ϕ) , mais par le point associé sur la sphère unité $(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$. La distance entre deux orientations est alors la longueur de la corde reliant leurs représentations sur la sphère unité. Une autre idée est de considérer la distance du grand cercle, qui correspond à la

longueur du chemin le plus court de la sphère unité reliant les deux points. Cette distance correspond à l'angle séparant les deux points sur le cercle unité les contenant, et est égale à, pour deux points de coordonnées polaires $(1, \theta, \phi)$ et $(1, \theta', \phi')$:

$$d[(\theta, \phi)(\theta', \phi')] = 2 \operatorname{asin} \sqrt{\sin^2 \left(\frac{\theta - \theta'}{2} \right) + \sin \theta \sin \theta' \sin^2 \left(\frac{\phi - \phi'}{2} \right)}$$

La distance du grand cercle est celle qui représente le mieux notre problème, on la considèrera donc par la suite.

La méthode des K-means étant maintenant définie, il est nécessaire de s'occuper de l'initialisation de la classification, c'est-à-dire déterminer le nombre de classes, d'une part, et un centre originel c_k^0 associé à chaque cluster k . Pour cela, on utilise la méthode des noyaux, qui consiste à associer à chaque droite détectée non pas un point unique de la sphère unité de coordonnées $(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$, mais une densité de probabilité sur la sphère centrée en ce point. On choisit une fonction $f : [0 : \pi] \rightarrow [0 : 1]$ intégrable et décroissante telle que $f(\pi) = 0$.

On note $(\theta_k, \phi_k)_{k \in [1 : K]}$ l'ensemble des angles de chaque droite détectée. On construit la fonction $\Psi((\theta_k, \phi_k)_{k \in [1 : K]})(\alpha, \beta) : [0 : \pi] \times [0 : 2\pi[\rightarrow [0 : 1]$ définie par :

$$\Psi((\theta_k, \phi_k)_{k \in [1 : K]})(\alpha, \beta) = \frac{1}{K} \sum_{k=1}^K f(d[(\alpha, \beta)(\theta_k, \phi_k)])$$

Pour l'initialisation des classes, on recherche alors les maximums locaux de la fonction $\Psi((\theta_k, \phi_k)_{k \in [1 : K]})$. Cette fonction, par l'effet d'étalement dû à la fonction f , possède des maximums dans les régions où la densité de points (θ_k, ϕ_k) est plus importante. Ce calcul de densité dépend beaucoup de la fonction f qui représente un étalement plus ou moins fort. Par exemple, si on prend pour f l'impulsion de Dirac δ , alors chaque (θ_k, ϕ_k) correspond à un maximum local.

De plus, suite au choix de f , il est aussi nécessaire de définir un seuil minimum à partir duquel un maximum local sera considéré. En effet, si l'on a un maximum local qui correspond à un unique point isolé, il est probablement peu pertinent de créer une classe pour un point seul. Heureusement, dans notre cas, la détection des droites est contrainte en

angle. Par conséquent, on sait que toutes les droites non parfaites détectées se trouvent proches d'une droite parfaite. Ainsi, si toutes les droites parfaites d'un même fil ont même orientation, que chaque fil possède au moins une droite parfaite et un nombre conséquent de droites détectées au total, on a alors un a priori assez fort sur une fonction f qui permettrait d'éviter tout isolement.

Cependant, dans certains cas, il est possible d'avoir des fils spatialement proches (leur séparation doit donc être effectuée en angle) et suffisamment proches en angles pour ne pas être séparables si l'on prend une fonction f qui assure que chaque point n'est pas isolé. Il est nécessaire dans de tels cas de prendre une fonction moins étalée, ce qui pose de nouveau le problème des points isolés. En général, un seuil minimal sera donc tout de même défini.

Algorithmiquement, on connaît donc au départ l'ensemble $(\theta_k, \phi_k)_{k \in [1:K]}$. Pour initialiser la méthode des K-means, on va donc se donner une discrétisation de l'ensemble $[0; 1]^3$ en une image I de $(2n + 1)^3$ éléments. Cette image est initialisée à 0 au départ. On effectue alors : $\forall k \in [1:K], \forall (u, v, w) \in [0:2n]^3$ avec $\left(\frac{n-1}{n}\right)^2 < \frac{(u-n)^2 + (v-n)^2 + (w-n)^2}{n^2} \leq 1$,

$$I(u, v, w) = I(u, v, w) + f(d[(\alpha, \beta)(\theta_k, \phi_k)])$$

avec $(u, v, w) = (r \sin \alpha \cos \beta, r \sin \alpha \sin \beta, r \cos \alpha)$ et d la distance du grand cercle.

On recherche ensuite les maximums locaux de l'image I . Sur une image discrète, on définira un maximum local de la manière suivante :

$I(s)$ est un maximum local si et seulement si il n'existe pas de chemin continu $(e_i)_{i < N}$ tel que $e_0 = s, I(e_{N-1}) > I(s)$ et $\forall 0 < i < N - 1, I(e_i) = I(s)$.

Ainsi, un voxel est un maximum local s'il appartient à un ensemble connexe de voxels de même valeurs n'ayant aucun voisin plus grand. De cette manière, les voxels des « paliers » sont donc exclus des maximums locaux.

La détermination de cet ensemble de maximums est une opération de complexité polynômiale. Pour l'effectuer, on commencera par une première passe où on éliminera tous les voxels ayant un voisin strictement plus grand qu'eux. Ensuite, on effectuera de multiples passes où pour chacune, on éliminera les voxels ayant un voisin de valeur égale déjà éliminé.

Le nombre de passes nécessaires est donc majoré par la distance maximale entre deux points d'un palier de valeur.

Finalement, on obtient un ensemble de paliers de valeurs maximales. Or, considérer chaque point du palier comme un point d'initialisation potentiel est totalement contre-productif. On associera plutôt un point à chaque palier, ce qui ne posera de problèmes que si deux points d'initialisation recherchés se trouvent dans le même palier, cas somme toute très improbable (si cela arrivait, on aurait un seul point pour deux clusters). On effectue donc un calcul de barycentre sur chaque palier maximal. Les points issus de ce calcul seront considérés comme les initialiseurs de la méthode des K-means.

Via la méthode des K-means, on obtient alors un clustering cohérent en termes d'angles. Cependant, le suivi des fils requiert aussi un suivi de la localisation des fibres. En effet, il est possible d'obtenir, sur une seule image, plusieurs fils distincts présentant une orientation très similaire. De plus, la détection n'étant pas toujours complètement parfaite, il est possible d'obtenir des fibres ne suivant pas du tout l'orientation des fils auxquels elles appartiennent. La classification spatiale va donc nous permettre d'obtenir un regroupement supplémentaire et différent, que l'on recoupera ensuite avec le clustering en angle pour détecter les fils et éliminer les droites aberrantes détectées.

La manière de procéder sur les informations spatiales ne peut s'apparenter en elle-même à une classification, mais plutôt à une représentation des liens entre les fibres. En effet, la méthode présentée consiste en un maillage de l'ensemble des points d'intersection des fibres avec les bords de l'image, qui permet ensuite de définir un système de « fibres voisines » dans le cas où le segment reliant deux de leur points d'intersection est aussi une arête du maillage.

Le maillage doit bien entendu être fait d'une manière respectant le plus possible des critères de distances et de formes. En particulier, on cherchera un comportement isotrope du mailleur, c'est-à-dire que les triangles formés autour d'un point ne doivent pas être trop dépendants de ce qui se passe « loin » de lui, et de fait, on cherchera à privilégier des triangles les « moins plats » possibles. Pour ce faire, on utilise la méthode de la triangulation de Delaunay, que l'on présente ci-dessous :

Définition : Soit un ensemble $E \subset \mathbb{R}^n$ convexe borné et F un ensemble fini de points de E . Soit $y \in F$. On appelle **cellule de Voronoï** de y l'ensemble

$$\mathcal{V}(y) = \{x \in E \text{ tel que } \forall z \in F, \|x - y\| \leq \|x - z\|\}$$

La cellule de Voronoï de y est donc l'ensemble des points de E plus proches de y que de tout autre point de F (Voronoi, 1908).

Si pour chaque doublet $(y, z) \in F^2$, on note $E(y/z)$ et $E(z/y)$ les deux demi-espaces séparés par l'hyperplan médian de (y, z) contenant respectivement y et z , alors on peut caractériser la cellule de Voronoï de y comme :

$$\mathcal{V}(y) = \bigcap_{z \in F, z \neq y} E(y/z)$$

De fait, une cellule de Voronoï est toujours convexe comme intersection finie d'ensembles convexes.

On dira aussi que deux cellules de Voronoï $\mathcal{V}(y)$ et $\mathcal{V}(z)$ sont voisines si et seulement si

$$\mathcal{V}(y) \cap \mathcal{V}(z) \neq \emptyset$$

i.e. qu'il existe $x \in E$ tel que $\|x - y\| = \|x - z\|$ et que

$$\forall w \in F \setminus \{y, z\}, \|x - w\| \geq \|x - y\|$$

On appellera diagramme de Voronoï l'ensemble des cellules de Voronoï de F (Figure 49). Un tel diagramme est très utile dans les calculs de distance, comme cela sera montré dans la partie suivante, car l'appartenance d'un point x à une cellule de Voronoï permet de déterminer directement de quel point $y \in F$ il est le plus proche.

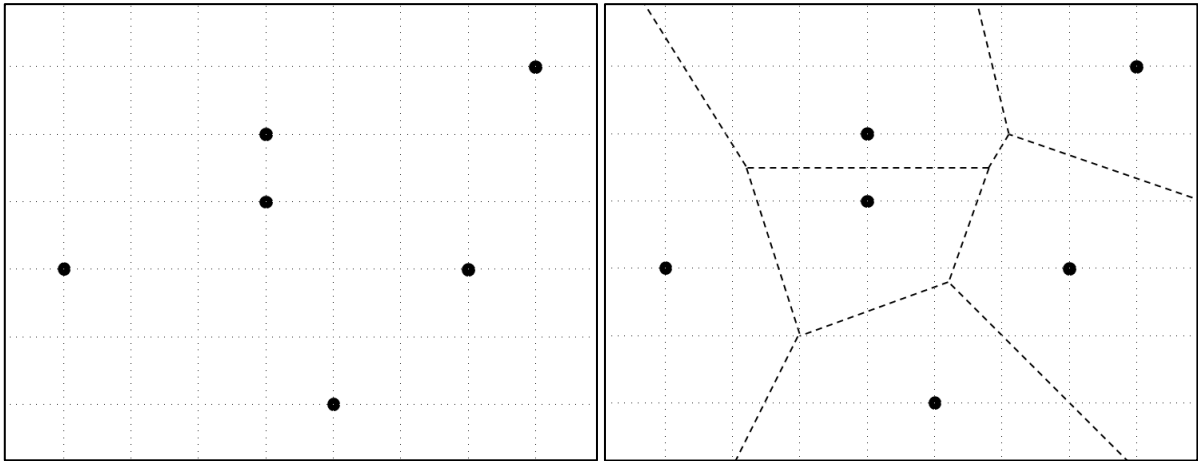


Figure 49 : Illustration du diagramme de Voronoï. A gauche, l'image de départ considérée, F correspondant à l'ensemble des points noirs. A droite, les arêtes des cellules de Voronoï sont représentées par les traits pointillés noirs.

On se placera maintenant dans le cas $E \subset \mathbb{R}^2$.

On constatera que si un point x appartient à trois cellules de Voronoï $\mathcal{V}(w)$, $\mathcal{V}(y)$ et $\mathcal{V}(z)$, alors ce point est le centre du cercle circonscrit au triangle $(w ; y ; z)$. De plus, aucun autre point de F n'appartient à l'intérieur du cercle circonscrit (sinon ce point serait plus proche de x , qui n'appartiendrait donc pas aux cellules de Voronoï $\mathcal{V}(w)$, $\mathcal{V}(y)$ et $\mathcal{V}(z)$). De même, réciproquement, si aucun point de F n'appartient au cercle circonscrit au triangle $(w ; y ; z)$, alors le centre du cercle est l'intersection des trois cellules de Voronoï $\mathcal{V}(w)$, $\mathcal{V}(y)$ et $\mathcal{V}(z)$.

Définition : Si $\mathcal{V}(w)$, $\mathcal{V}(y)$ et $\mathcal{V}(z)$ partagent un point commun, alors on dira que le triangle $(w ; y ; z)$ est un **triangle de Delaunay** sur F .

Proposition : Si $(w ; y ; z)$ est un triangle de Delaunay et qu'on note $\mathcal{C}(w ; y ; z)$ le cercle circonscrit au triangle, alors

$$F \cap \overset{\circ}{\mathcal{C}}(w ; y ; z) = \emptyset$$

($\overset{\circ}{A}$ désignant l'intérieur de l'ensemble A)

Delaunay définit les tétraèdres éponymes dans le cadre général $E \subset \mathbb{R}^n$ (Delaunay, 1934). Ce qui est défini en dimension deux s'étend aussi à toute dimension. Nous ne nous intéresserons ici qu'au cas planaire, suffisant dans notre cas.

Définition : On appellera enveloppe convexe de F le plus petit convexe de E contenant F .

Des méthodes algorithmiques de calcul de l'enveloppe convexe sur un ensemble fini de points du plan ont été proposées par Graham (Graham, 1972) et Jarvis (Jarvis, 1973).

On utilisera aussi le théorème suivant :

Théorème de Carathéodory (Carathéodory, 1907) : Dans un espace affine de dimension n , l'enveloppe convexe d'un sous-ensemble F est l'ensemble des barycentres à coefficients positifs ou nuls des familles de $n + 1$ points de F .

De là, on peut déduire que l'enveloppe convexe d'un nombre fini F de points est un polygone dont les sommets sont des points de F .

On supposera par la suite qu'il n'existe pas trois points de F colinéaires, ni quatre points cocycliques. Dans ce cadre, il n'existe pas de points « quadruples » dans le diagramme de Voronoï, et deux cellules sont donc reliées par une arête de Delaunay si une arête de Voronoï les sépare.

Définition : On appellera **triangulation de Delaunay** l'ensemble des triangles de Delaunay de F (Figure 50).

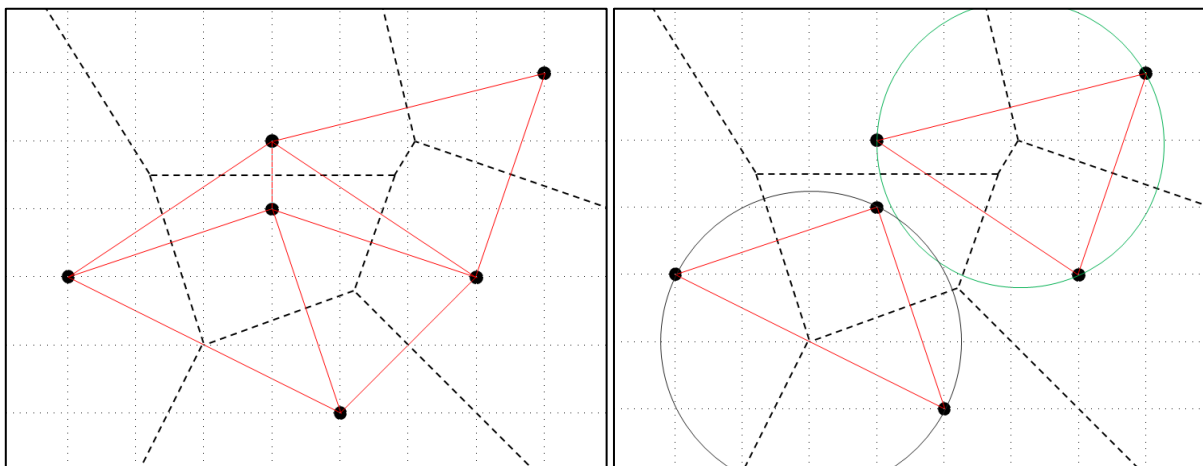


Figure 50 : A gauche, triangulation de Delaunay obtenue pour l'image considérée à la Figure 49. A droite, illustration sur deux des triangles de Delaunay de la propriété concernant les cercles circonscrits. On constate bien qu'aucun point de F n'appartient au cercle circonscrit au triangle de Delaunay.

Proposition : Un point quelconque de l'intérieur d'un triangle de Delaunay appartient à un seul triangle de Delaunay.

En effet, si un point appartient à l'intérieur de deux triangles de Delaunay différents, alors les deux triangles s'intersectent au moins sur une arête (sinon, un des deux triangles serait inscrit dans l'autre, ce qui contredit la propriété sur l'absence d'autres points dans le cercle circonscrit). Or, une arête de Delaunay relie deux points dont les cellules de Voronoï se touchent, et par conséquent, tout point de cette arête est plus proche d'un des deux points de l'arête que de n'importe quel autre point de F . Comme deux arêtes des deux triangles s'intersectent, on a un point qui appartient à une arête de Delaunay tout en appartenant à une autre cellule de Voronoï, ce qui est exclu.

Proposition : La triangulation de Delaunay de F recouvre exactement l'enveloppe convexe de F (Figure 51).

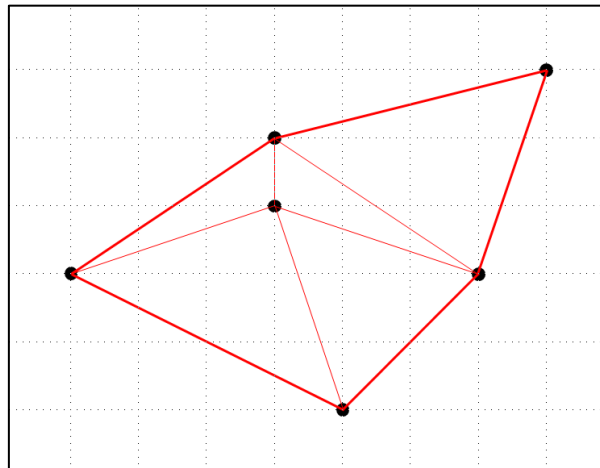


Figure 51 : Triangulation de Delaunay et enveloppe convexe (représentée par les traits épais). On constate le recouvrement de l'enveloppe par la triangulation.

L'inclusion de la triangulation dans l'enveloppe convexe est simple. En effet, si un point x appartient à la triangulation, alors il appartient à un triangle $(a ; b ; c)$ de points de F . Ainsi, x est barycentre à coefficients positifs ou nuls de F , donc x appartient à l'enveloppe convexe de F . D'un autre côté, si x appartient à l'enveloppe convexe de F , alors x appartient à un triangle formé par trois points de F d'après le théorème de Carathéodory. On va montrer par récurrence :

Si $x \in (a ; b ; c)$ tel que $\text{card}(F \cap \mathcal{C}(a ; b ; c)) = n$, alors x appartient à un triangle de Delaunay de F .

- Si $n = 0$, alors $(a ; b ; c)$ est un triangle de Delaunay de F , donc on a bien le résultat.

- Supposons le résultat vrai pour tout $k \leq n$ fixé. Soit $x \in (a; b; c)$ tel que $\text{card}(F \cap \mathcal{C}(a; b; c)) = n + 1$. On prend $d \in F \cap \mathcal{C}(a; b; c)$ quelconque. Quitte à modifier l'ordre des points, on posera $x \in (a; b; d)$. On a alors $x \in (a; b; d)$ tel que $\text{card}(F \cap \mathcal{C}(a; b; d)) = k \leq n$. Par récurrence, on a donc que x appartient à un triangle de Delaunay.

La triangulation de Delaunay est donc une méthode intéressante de triangulation de l'enveloppe convexe d'un ensemble de points. La propriété des triangles de Delaunay concernant les cercles circonscrits assure que les triangles ne seront de forme étirée que si cela est nécessaire. La triangulation de Delaunay assure aussi une certaine continuité dans la taille des triangles.

Algorithmiquement, la borne inférieure de la complexité temporelle du calcul d'un diagramme de Voronoï est $O(n \ln(n))$. Parmi les algorithmes développés, on notera les méthodes incrémentales (Green & Sibson, 1977) (Ohya, et al., 1984), construisant le diagramme en ajoutant les points de F un à un. Les méthodes incrémentales ont une complexité dans le pire des cas en $O(n^2)$, mais la constante associée est faible, ce qui rend ces algorithmes rapides dans beaucoup de cas classiques. Une autre approche classique est une méthode fondée sur le *divide-and-conquer*, calculant des diagrammes de Voronoï locaux qui sont fusionnés ensuite (Shamos & Hoey, 1975). Cette approche présente une meilleure complexité temporelle, de l'ordre de $O(n \ln(n))$, mais la réalisation de la phase de fusion est délicate. La méthode proposée par Fortune (Fortune, 1987), finalement, est aussi de complexité temporelle en $O(n \ln(n))$, mais ne fonctionne pas sur une approche de *divide-and-conquer*.

En ce qui concerne la triangulation de Delaunay, elle est linéairement liée au diagramme de Voronoï. La meilleure complexité qu'on puisse obtenir pour la détermination de celle-ci est donc aussi en $O(n \ln(n))$. Deux algorithmes permettant de déterminer la triangulation de Delaunay sont présentés par Lee et Schachter (Lee & Schachter, 1980), semblables aux algorithmes présentés pour le diagramme de Voronoï, c'est-à-dire une approche *divide-and-conquer* en $O(n \ln(n))$, et une approche itérative en $O(n^2)$ avec une faible constante de temps.

Une triangulation de Delaunay des intersections de fibres sur les bords de l'image nous permet donc d'obtenir, pour chaque fibre, un système de voisinage propre. La donnée du voisinage permet aussi de construire une connexité sur cet ensemble comme celle utilisée sur \mathbb{N}^3 . A partir de cela, on peut alors définir un fil comme un sous-ensemble connexe maximal d'une classe issue du clustering en angle.

On connaît donc :

- Une liste des droites $D = (d_i)_i$ détectées par l'algorithme de reconnaissance, et les coordonnées des deux points d'intersections avec les bords de l'image associés $(p_i^1)_i$ et $(p_i^2)_i$.
- Pour chacune de ces droites, on connaît le cluster en angle qui lui est associé $(l_i)_i$.

A partir des $(p_i^j)_{i,j}$, on construit donc un diagramme de Delaunay. Le voisinage d'un point p_i^j quelconque, noté $\mathcal{N}(p_i^j)$, correspond à l'ensemble des $(p_u^v)_{u,v}$ tel que $(p_i^j p_u^v)$ soit l'arête d'un triangle de Delaunay. Par conséquent, on peut définir le voisinage d'une droite d_i quelconque, noté $\mathcal{N}(d_i)$, comme l'ensemble des droites $(d_u)_u$ telles qu'il existe un couple $(j, v) \in \{1; 2\}^2$ pour lequel $(p_i^j p_u^v)$ est une arête de Delaunay.

De là, on peut définir la notion de connexité sur ce système, $E \subset D$ est un connexe si pour tout doublet de droites de E , il existe un chemin de voisins les reliant, et $F \subset E$ est un connexe maximal s'il est connexe et que $\forall d \in E \setminus F, F \cup \{d\}$ n'est pas connexe.

Ainsi, on définira les ensembles $(E_k)_k$ comme l'ensemble des droites $\{d_i \text{ tel que } l_i = k\}$, et finalement, on définira les fils comme les connexes maximums des ensembles $(E_k)_k$.

A ce stade, les fibres aberrantes devraient former des fils ne comportant qu'un nombre très limité de fibres (dans la très grande majorité des cas, ce nombre est égal à un), et il devient alors possible de détecter et supprimer les fibres aberrantes par suppression des composantes connexes ne possédant que très peu d'éléments.

Il est cependant nécessaire aussi de se donner une méthode de fusion de fils dans le cas où la classification par K-Means se révélerait trop détaillée. Auquel cas, on se donnera une différence d'orientation maximale α , et une valeur seuil $1 > \varepsilon > 0$ petite. Pour chaque

doublet de fils (f_i, f_j) , on notera $\kappa_{i,j}$ le nombre d'arêtes de Delaunay reliant des fibres de chacun des fils. On notera alors

$$\lambda_{i,j} = \frac{\kappa_{i,j}}{\max(\sum_k \kappa_{i,k}, \sum_k \kappa_{j,k})}$$

Finalement, deux clusters (f_i, f_j) seront fusionnés en un seul fil si et seulement si :

- L'angle $\delta_{i,j}$ séparant les deux orientations moyennes vérifie $\delta_{i,j} < \alpha$
- Le contact entre les deux clusters est significatif, i.e. $\lambda_{i,j} > \varepsilon$.

On a cependant occulté ici un problème concernant la réalisation de la triangulation de Delaunay. En effet, les points considérés sont tous sur les faces d'un parallélépipède, et on cherche à fabriquer un maillage 2D sur cette surface. Dans ce cas, il est impossible de trouver une fonction de distance bien définie. On peut, pour résoudre ce problème, soit totalement ignorer la triangulation aux arêtes (on obtient alors des maillages distincts sur chaque face), soit considérer le patron du parallélépipède posé à plat et centré sur la face considérée (ce qui peut alors poser des problèmes d'unicité de la triangulation dans les coins), soit projeter toutes les fibres sur le plan contenant la face considérée. Cette dernière possibilité est de loin la meilleure étant donné qu'elle représente fidèlement la distance entre les fibres. De plus, les fibres extérieures à la face qui sont considérées pour la triangulation sont relativement proches des arêtes. De fait, leurs positions relatives ne changent que très peu en projetant sur la face considérée. Ainsi, on peut considérer que la triangulation aux arêtes tend vers l'unicité, et s'il n'y a pas unicité, alors c'est qu'une fibre s'est beaucoup déplacée relativement aux autres sur une petite distance, ce qui sous-entend que cette fibre est une erreur de détection.

Algorithmiquement, le procédé est donc le suivant :

On se donne N droites $(d_i)_{i < N}$ et K classes issues du clustering par K-means. On note $(l_i)_{i < N}$ l'ensemble des classes associées respectivement à chaque droite. On connaît aussi les dimensions de l'image de départ. Le principe est alors le suivant :

- Pour toute face F de l'image, on déduit le plan $P(F)$ contenant la face.

- On construit l'ensemble $(p^{P(F)}(d_i))_{i < N}$ des intersections de $P(F)$ avec toutes les droites $(d_i)_{i < N}$. Si une droite est parallèle au plan, alors on ne la considère pas. Pour simplifier, on considère ici que ce cas de figure n'existe pas.
 - Pour chaque $j < N$ tel que $p^{P(F)}(d_j) \in F$, on détermine $T^F(j)$ l'ensemble des sommets des triangles de Delaunay de $(p^{P(F)}(d_i))_{i < N}$ contenant $p^{P(F)}(d_j)$.
- Pour chaque $j < N$, la droite d_j intersecte deux faces de l'image notées F et G . On désigne par $\mathcal{N}(d_j)$ l'ensemble :
$$\{i < N \text{ tel que } p^{P(F)}(d_i) \in T^F(j) \text{ ou } p^{P(G)}(d_i) \in T^G(j)\}$$
- Pour chaque classe $k < K$, on note $E_k = \{d_i, i < N \text{ tel que } l_i = k\}$.
 - On recherche alors les connexes maximums de E_k pour le voisinage \mathcal{N} , que l'on notera $(C_{k,m})_{m < M(k)}$.
 - Pour tout $m < M(k)$, si $\text{card}(C_{k,m}) < \alpha$ fixé (généralement $\alpha = 2$ suffit), alors $C_{k,m}$ est considéré comme un fil aberrant et est supprimé.
 - Sinon, $C_{k,m}$ est considéré comme un fil, d'indice (k, m) .

Notons que contrairement à ce qui est présenté dans les parties précédentes, le calcul des composantes connexes des E_k ne peut pas se faire avec les mêmes algorithmes optimisés. En effet, ceux-ci sont conçus pour fonctionner avec un voisinage isotrope particulier de type 6, 18 ou 26-voisinage. Ici, le voisinage est complètement anisotrope. De fait, la manière choisie pour calculer les composantes connexes est la plus évidente mais aussi la plus lourde en calculs, i.e. la propagation des voisinages à partir de droites choisies. Heureusement, le nombre de droites étant très faible face aux nombres de points considérés dans les précédentes utilisations d'un algorithme de recherche de régions connexes, le temps de calcul de l'algorithme naïf sur les droites reste négligeable tant que la taille des images et par conséquent le nombre de droites n'explose pas.

2.5. Découpage et suivi

Malgré la recherche d'un algorithme de détection de fibres et de fils le plus performant possible, la complexité des méthodes proposées atteint toujours des limites. La méthode de détection présentée dans la partie 2.4.2 a une complexité théorique en $O(L^3)$ sur les images étudiées, mais dans les faits, le temps de calcul ne suit ce comportement que pour des trop grandes valeurs de L , ce qui nous laisse apparaître, dans le domaine d'étude, un comportement en $O(L^5)$ bien moins intéressant. Ainsi, sur les plus grandes images considérées, le temps de calcul continue, malgré les méthodes proposées, d'exploser et de rendre le calcul rapide inenvisageable.

On décide donc, pour combler cela, de regarder le comportement de l'algorithme si on le combine à une approche en *divide-and-conquer*.

On commence par supposer qu'il existe une méthode permettant de fusionner les résultats de détection de différents domaines voisins de taille L^3 en complexité $f(L)$. Si l'on suppose que l'algorithme fonctionne sur une image de taille 1^3 et que l'image de base est de taille $L^3 = 2^{n^3}$ avec $n \in \mathbb{N}$, alors la méthode *divide-and-conquer* fonctionne et sa complexité s'exprime sous la forme :

$$C = \sum_{k=0}^n 2^{k^3} f\left(\frac{L}{2^k}\right)$$

Dans notre cas, la phase de fusion correspond à une association des fils et/ou des fibres entre elles, selon leurs orientations et leurs intersections avec les plans. Ceci peut donc se faire en une complexité de $O(L^2)$ car uniquement des opérations surfaciques sont impliquées. De fait, on pose $f(L) = L^2$ et donc, on a comme complexité :

$$C = \sum_{k=0}^n 2^k L^2 = O(L^3)$$

Dans les faits, on procèdera plutôt au calcul sur toutes les sous-images d'abord, suivi d'une phase de fusion des résultats. La complexité de la partie de calcul est en $O(L^3)$, car le calcul sur chaque sous-image est de complexité constante, et le nombre de ces sous-images

est en $O(L^3)$. La partie fusion se fait aussi en complexité constante sur une sous-image, et donc en complexité $O(L^3)$ au total.

Il est aussi à noter qu'il est en réalité impossible d'utiliser l'algorithme de détection sur des images à un voxel. On a donc une taille d'image minimale L_0^3 en dessous de laquelle on ne peut plus descendre. Ceci ne fait pas varier la complexité asymptotique, mais à une incidence sur la constante associée, qui prendra en compte la complexité de l'opération sur une image de taille minimale.

Ainsi, le découpage de l'image permet donc d'obtenir une complexité se comportant de manière linéaire avec la taille de l'image, i.e. en $O(L^3)$, dès que la taille de celle-ci devient significative devant L_0^3 , alors qu'auparavant, ce comportement était atteint lorsque l'image devenait représentative du matériau pour une taille Λ^3 . Dans le cas des images à haute résolution présentée au départ, on obtient des valeurs de l'ordre de :

$$50 < L_0 < 100 \text{ et } 1000 < \Lambda < 2000$$

On gagne donc un facteur 20 environ sur la valeur de L minimale à partir de laquelle la complexité suit un comportement linéaire. Cependant, sur les images à faible résolution, on obtient alors des valeurs de l'ordre de :

$$35 < L_0 < 70 \text{ et } 400 < \Lambda < 900$$

Le facteur de gain est maintenant plus proche de 10. En effet, la variation de la résolution fait varier d'autant le volume représentatif du matériau, mais la plus grande difficulté à détecter des fibres devenues très proches et les erreurs plus grandes sur les bords de l'image nous empêchent de diminuer d'autant le volume minimal. Le rayon des fibres joue le rôle le plus important dans la détermination de L_0 , mais la densité de fibres va aussi faire varier la valeur en fonction de la facilité à détecter deux fibres proches distinctes.

Dans les cas étudiés ici, on considère généralement des images de taille minimale de 100^3 , ce qui nous permet, dans les deux résolutions, de faire apparaître directement le comportement linéaire.

Concernant la fusion des résultats, celle-ci peut-elle être effectuée simplement en $O(L^2)$? Deux réponses sont possibles selon ce que l'on cherche à fusionner :

- Si l'on cherche à fusionner les fils, c'est assez simple. En effet, les fils correspondent à des assemblages de très nombreuses fibres, il est donc aisé de constater la cohérence entre de gros ensembles. Pour se faire, on utilisera la triangulation de Delaunay définie à la partie 2.4.3. Sur chaque face, on a donc une triangulation définie à partir des points d'intersection des fibres avec celle-ci. On appellera empreinte d'un fil f sur une face, notée $\mathcal{E}(f)$, l'ensemble des points appartenant à un triangle de Delaunay dont les trois sommets sont associés à des fibres appartenant au fil f . La fusion de deux fils voisins f, g se fera donc en comparant l'aire de leur empreinte commune $\mathcal{A}(\mathcal{E}(f) \cap \mathcal{E}(g))$ avec $\min(\mathcal{A}(\mathcal{E}(f)), \mathcal{A}(\mathcal{E}(g)))$.

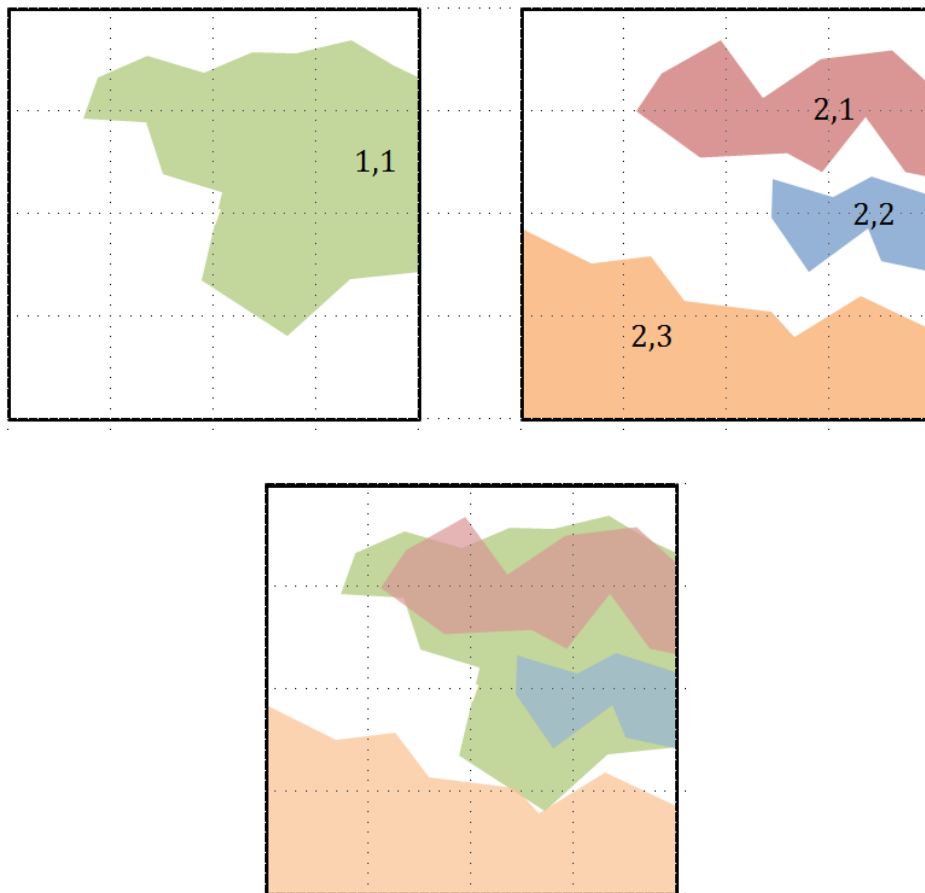


Figure 52 : Vue d'une face sur laquelle on procède à la fusion de fils. En haut à gauche, la face sur l'image 1, sur laquelle on voit l'empreinte d'un fil noté 1,1. A droite, la face sur l'image 2, sur laquelle on voit l'empreinte de trois fils 2,1, 2,2, 2,3. En bas, la superposition des empreintes. Les fils 2,1 et 2,2 seront fusionnés avec le fil 1,1, mais pas le fil 2,3.

Dans ce cas, les triangles de Delaunay ayant déjà été calculés auparavant, il ne reste plus qu'à calculer les aires et vérifier les intersections. On a bien une méthode simple en $O(L^2)$ (Figure 52).

- Si l'on cherche à fusionner les fibres, c'est plus compliqué. En effet, les fibres sont représentées par des lignes, donc des structures linéiques dont l'intersection avec les bords est réduite à un point. Auquel cas, il n'est pas possible de vérifier les empreintes. Heureusement, si l'on met un recouvrement de 1 voxel d'épaisseur dans le découpage, alors les deux faces des images voisines seront les mêmes, c'est-à-dire que dans un premier temps, les centres considérés seront aussi les mêmes. Cependant, les actualisations successives vont petit à petit mettre à mal l'égalité, car des nouveaux centres vont alors apparaître de la suppression des fibres détectées, et ceux-ci vont dépendre de la position du centre détecté (qui sera le même) mais aussi de l'orientation de la fibre (qui, elle, ne sera pas toujours identique).

On obtient alors sur chacune des deux images, un ensemble de centres considérés $(c_i^1)_{i < N}$ et $(c_i^2)_{i < M}$. A chaque centre on peut associer deux informations :

- Le centre est-il originel, c'est-à-dire détecté dès le départ de l'algorithme de détection ? Si oui, son homologue existe obligatoirement sur l'autre face. Sinon, le centre est dit ajouté, et on ne peut pas conclure directement.
- Le centre est-il membre d'une fibre à la fin de l'algorithme, ou est-il laissé seul ?

Plusieurs cas de figure sont alors possibles :

- Pour un centre originel associé à une fibre, on regarde son homologue
 - S'il est associé à une fibre, on fusionne les deux fibres.
 - Sinon, on cherche le fil le plus proche du centre non-associé, et on prolonge la fibre en suivant l'orientation de ce fil.
- Pour un centre ajouté associé à une fibre, on regarde :

- Si le centre a un homologue, on fait comme pour les points originels.
- S'il n'en a pas, on recherche un centre sur la face opposé n'ayant pas d'homologue et à distance inférieure à r (rayon des fibres) du centre considéré. S'il en existe un, on modifie les coordonnées des deux centres vers celles d'un point unique, leur barycentre, puis on effectue la fusion. S'il n'en existe pas, on prolonge la fibre en suivant l'orientation du fil le plus proche.

Dans le cadre où les fibres sont droites et la détection est performante, un centre ajouté a toujours un homologue sur l'autre face, ce qui nous permet alors de procéder de manière plus simple :

- Si deux fibres ont même point de contact, on les fusionne.
- Si une fibre est seule à son point de contact, on la prolonge.

Il est intéressant de noter que la méthode *divide-and-conquer* permet d'ajouter des fonctionnalités à l'algorithme, en autorisant le suivi de fils et de fibres courbes si, sur les images de taille minimale, ces courbes sont assimilables à des droites. Concernant les fils, ce passage à la détection de courbes est évident, la fusion ne portant que sur l'empreinte des fils et pas sur leur orientation. Dans le cas des fibres, on pourra argumenter que la fusion ne tient aussi que peu de cas de l'orientation, mais celle-ci influe sur la position et l'existence des centres ajoutés. L'évolution des centres considérés peut donc diverger sur les deux faces, ce qui complique la fusion. Pour autant, celle-ci est toujours fiable. Par contre, si on considère la détection rapide présentée en 2.4.2, alors les centres divergent beaucoup plus fortement en raison de l'imprécision de la détection. La fusion de fibres devient difficile dans de tels cas.

2.6. Tests et Résultats

Avant de tester la méthode et les algorithmes associés en condition réelles, il est nécessaire, pour valider leurs fonctionnements théoriques, de les vérifier sur une batterie d'images virtuelles parfaites, de quatre types différents.

Dans toutes les images testées générées ici, le rayon des fibres est fixé à 4 voxels. Le rayon n'influe qu'assez peu sur les tests (à un facteur multiplicatif près, on peut retrouver des images semblables avec un autre rayon), et les images étudiées ont des rayons de fibres comprises entre 2 et 6, on choisira donc 4 pour être assez proche des données réelles.

2.6.1. Image de fibres sans contacts

En premier lieu, on générera des images virtuelles représentant un ensemble de lignes aléatoires parfaites ne présentant aucun contact entre elles, i.e. la distance minimale entre fibres vérifie $\rho > 2r$ où r est le rayon des fibres (Figure 53). On se place alors dans un cadre simple où, modulo les erreurs de détection sur les bords de l'image, tout doit être retrouvé. Dans ce cadre, on effectue les tests sur différentes tailles d'image :

Numéro	1.1	1.2	1.3	1.4	1.5	1.6
Taille	100^3	200^3	300^3	400^3	500^3	600^3
Nombre de droites	18	75	168	300	468	675

Dans un tel cadre, nous n'avons aucun a priori sur la répartition des lignes et en particulier leurs orientations, mais comme les régions connexes sont réduites à une fibre, on pourrait attendre normalement un comportement en $O(L^3)$.

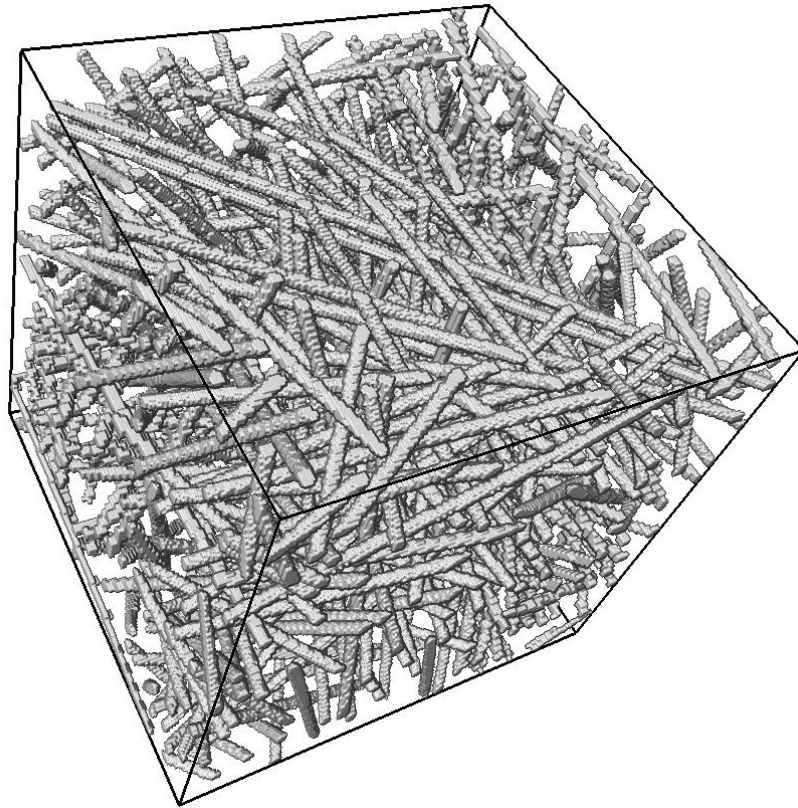
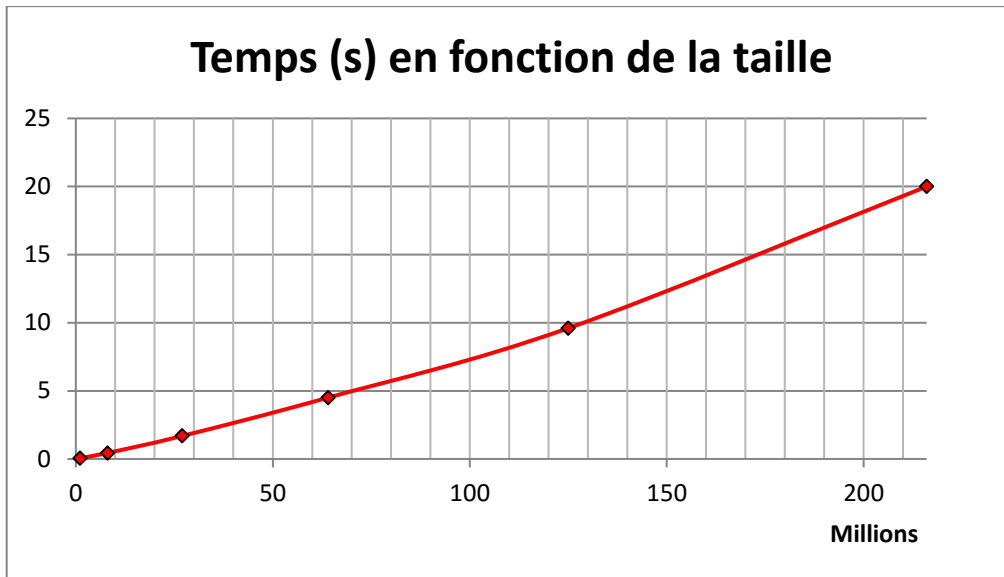


Figure 53 : Vue d'une image virtuelle générée de taille 400^3 avec 300 fibres sans contact.

Les résultats obtenus sont présentés dans le tableau ci-dessous :

Numéro	1.1	1.2	1.3	1.4	1.5	1.6
Temps (s)	0,06	0,44	1,7	4,5	9,6	20
Taux détection (%)	77	95	97	98	94	83

Si l'on trace le temps de calcul de l'algorithme en fonction de la taille totale de l'image, on obtient la courbe suivante :



Cette courbe suit une droite avec un coefficient de régression linéaire de 0,991. On peut admettre que le comportement temporel de l'algorithme en fonction de la taille de l'image est bien linéaire. Le taux de détection, pour sa part, est assez élevé pour les images de longueur 200 à 500. Les chutes de taux sur les autres valeurs sont interprétables : les fibres des bords étant non-considérées en raison du fort taux d'erreur sur celles-ci, l'image de longueur 100 rend des résultats moins bons en raison de la grande taille relative des bordures (et donc de la petite taille du domaine utile). Pour les images de longueur 600 et dans une moindre mesure 500, les erreurs sont plutôt à chercher du côté de la génération du matériau virtuel. En effet, pour suivre l'évolution naturelle de la quantité de fibres, il est nécessaire d'en générer un nombre suivant une fonction carrée. Cependant, cette évolution rapide du nombre de fibres sature rapidement le centre de l'image, ce qui fait que les fibres supplémentaires ont alors tendance à se retrouver excentrées sur les bords. Cette diminution du taux de détection est alors plutôt due à la génération qu'à l'algorithme (si on limite le nombre de droites générées à 400 sur l'image de taille 600^3 , on retrouve un taux d'environ 97 %).

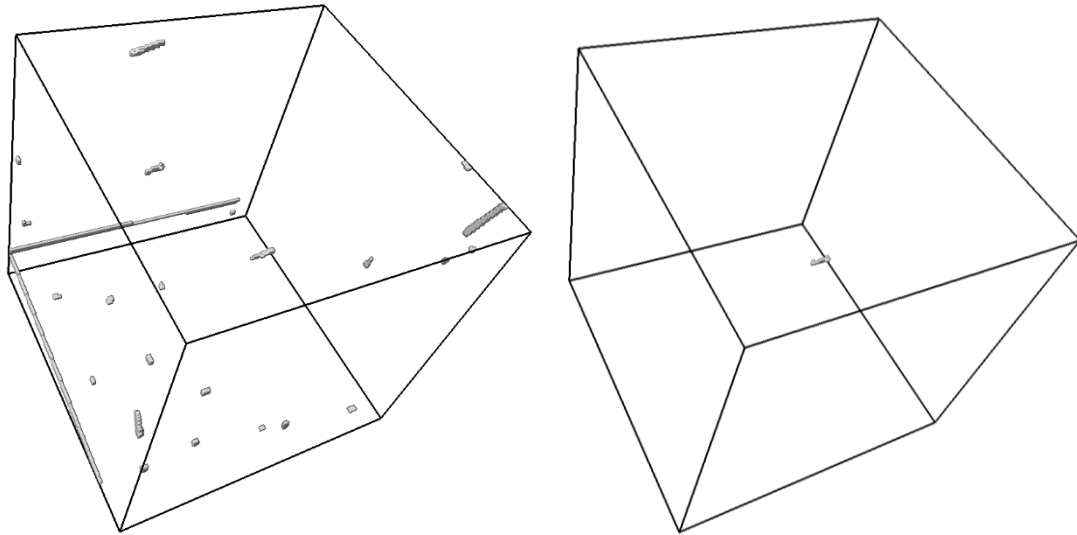


Figure 54 : A gauche, différence entre l'image de départ (voir Figure 53) et l'image résultat de l'algorithme de détection. A droite, la même image à laquelle on a retiré les bordures (de taille 390^3) et à laquelle il ne reste donc que le volume où tout est considéré.

On constate sur la Figure 54 que les différences au centre entre l'image générée et le résultat de l'algorithme de détection sont quasi-nulles.

Dans le cadre de fibres parfaites, aléatoires ne se touchant pas, on a donc un algorithme fonctionnel et rapide pour la détection de fibres.

2.6.2. Images de fibres avec contacts, excepté sur les faces

Dans un second temps, on effectue des tests sur des images où les fibres peuvent s'interpénétrer en vérifiant une distance minimale entre génératrices de fibres de $\rho < 2r$, excepté sur les bords de l'image où il n'y a aucun contact. On notera que ce type d'image n'a pas de sens réel (hormis en fabrication additive), il s'agit donc uniquement d'un test virtuel afin de vérifier l'influence de l'interpénétration seule, i.e. sans modifier les faces de l'image.

Ce que l'on peut attendre dépendra cette fois de nombreux paramètres. Plus l'image est dense et plus les interpénétrations sont fortes, plus les fibres vont être intersectées par des homologues. De fait, la détection de fibres les supprimant, les dernières fibres à détecter seront très creuses car amputées de tous les morceaux appartenant aux fibres précédentes. Ainsi, une forte tolérance devient nécessaire, ce qui est en soi potentiellement dangereux sur des images à forte densité. La méthode développée n'est probablement pas adaptée à de tels cas de figures

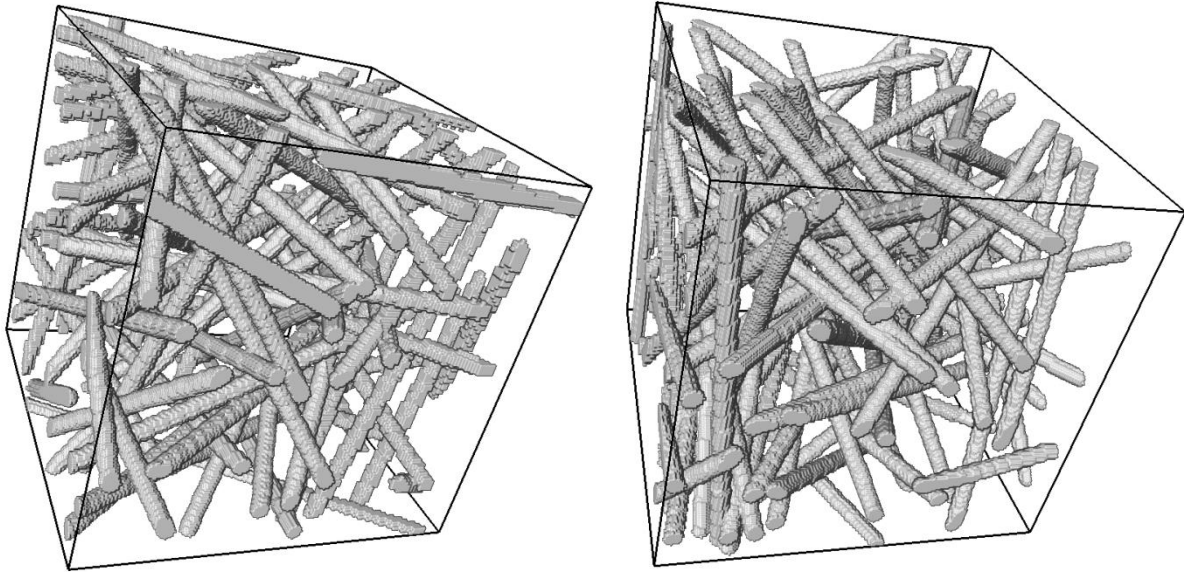


Figure 55 : Vue de deux images virtuelles générées de taille 200^3 avec 75 fibres avec interpénétration hors des faces (interpénétration admise de $r/2$ à gauche, de r à droite).

On testera les algorithmes sur plus d'images que précédemment :

Numéro	2.1	2.2	2.3	2.4	2.5	2.6
Taille	200^3	400^3	600^3	200^3	400^3	600^3
Nombre de droites	75	300	675	150	600	1350
Interpénétration	r	r	r	r	r	r

Numéro	2.7	2.8	2.9	2.10	2.11	2.12
Taille	200^3	400^3	600^3	200^3	400^3	600^3
Nombre de droites	75	300	675	150	600	1350
Interpénétration	$r/2$	$r/2$	$r/2$	$r/2$	$r/2$	$r/2$

Sur cet ensemble d'images (exemple sur la Figure 55), on obtient différentes possibilités :

- Si les images sont denses avec forte interpénétration (tests 2.4, 2.5, 2.6), alors la présence des nombreux recouvrements impose une tolérance élevée afin de tout détecter. Cependant, les valeurs de tolérance nécessaires (0,8 pour détecter quasiment tout l'intérieur, c'est-à-dire une acceptation dès que 20 % des points

sur la droite sont bons) entraînent aussi nombre de fausses détections avant même que tout ne soit retrouvé. L'algorithme développé est, il faut dire, inadapté par construction à ce genre d'exercice. En effet, dans le cas de fibres pleines s'intersectant, il suffit d'un algorithme à une passe (sans suppression de ce qui est détecté) avec une tolérance nulle pour obtenir de bons résultats.

- Si les images sont peu denses avec forte interpénétration (tests 2.1, 2.2, 2.3), la détection peut se faire avec une tolérance modérée. Les résultats sont alors bons, mais soumis au côté aléatoire de l'image, une droite pouvant par hasard être intersectée par de très nombreuses autres droites, et ainsi détruite par la détection des autres. Dans les faits, l'algorithme fonctionne convenablement dans ce cas. En théorie, celui-ci n'a pas été conçu pour les fortes interpénétrations, pour lesquelles on préférera une approche « non-destructive » (ne supprimant pas d'information au cours de l'exécution). Grâce à ces vérifications, il est tout de même agréable de voir que quelques rares interpénétrations fortes ne remettent pas en cause le fonctionnement général de l'algorithme.
- Si les images sont denses avec faible interpénétration (tests 2.10, 2.11, 2.12), la détection se fait bien avec une tolérance nulle (Figure 56). En effet, dans ce cas, les génératrices ne sont plus affectées par la suppression des fibres détectées, ce qui rend l'algorithme aussi performant en résultats que sur les images sans contact. Cependant, les composantes connexes sont cette fois plus grandes, ainsi l'algorithme est plus lent que dans le cas sans contact.

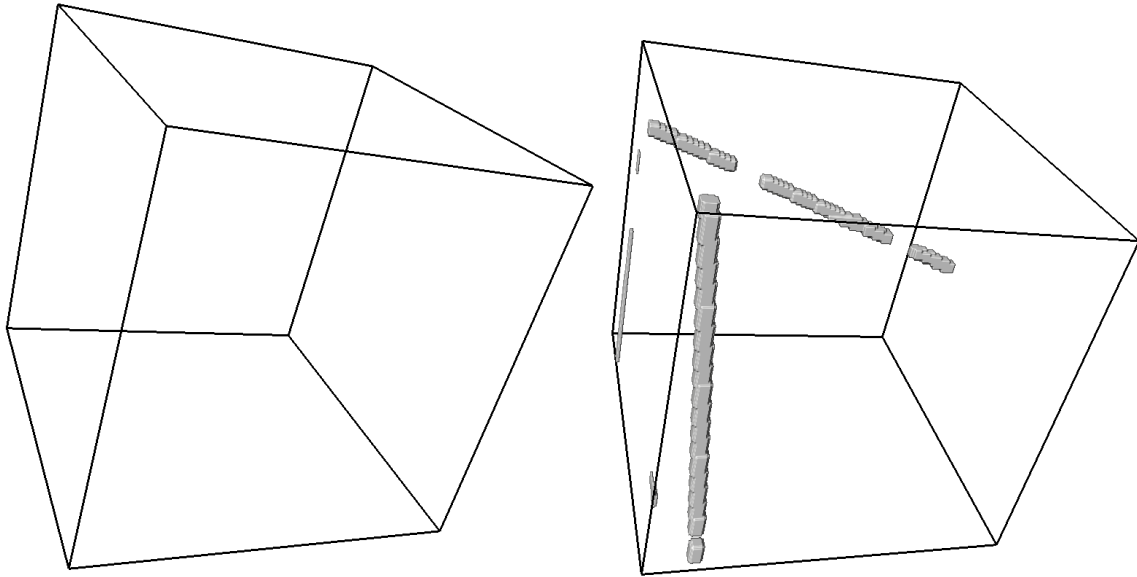


Figure 56 : Différence entre le résultat et l'origine pour les deux images de la Figure 55 (sans les bordures).

Tests	2.1 à 2.3	2.4 à 2.6	2.7 à 2.9	2.10 à 2.12
Taux détection moyen (%)	~75	~65	~95	~95

L'algorithme est donc fonctionnel pour des images avec interpénétration, tant que celles-ci ne deviennent pas trop importantes. Dans le cadre des images fibreuses réelles, ceci est parfaitement suffisant, deux fibres pouvant se toucher, voire se déformer légèrement au contact, mais l'intersection brute des fibres étant impossible (sauf fabrication additive).

2.6.3. Images de fibres avec contacts

Dans un troisième temps, des tests sont effectués sur les mêmes images que précédemment, mais en supprimant la contrainte sur les faces. Ainsi, les contacts sur les faces peuvent maintenant être les mêmes que sur le reste des images. On notera ces images tests de 3.1 à 3.12.

Dans ce cadre, on obtient des résultats très proches de ceux des images avec interpénétration autorisée sauf sur les bords. On peut en déduire que les intersections de fibres sur les bords de l'image n'ont que peu d'influence sur le fonctionnement de l'algorithme. Ceci est prévisible ; les centres détectés par la méthode de carte de distance

restent des points de la génératrice quand deux fibres s'interpénètrent tant que la distance entre les génératrices reste supérieure au rayon des cylindres. De fait, les résultats sont bien cohérents avec ce que l'on attendait.

Cependant, plus intéressant, les résultats restent semblables si on utilise la méthode rapide, fondée non pas sur une méthode de carte de distance, mais sur des régions connexes. Cette méthode a pourtant tendance à déplacer les fibres lorsque quelques-unes se touchent, et à corrompre le résultat dans le cas de contacts plus importants. Ici, contrairement à ce que l'on pourrait attendre en premier lieu, la méthode de détection rapide donne pourtant de bons résultats, qui sont en réalité explicables par la génération aléatoire du matériau. En effet, les fibres auront une tendance à traverser le milieu de l'image, qui s'en retrouvera densifié. En contrepartie, dans les régions « où il y a plus de place », c'est-à-dire les bordures, la densité sera moindre, et les contacts entre fibres aussi par conséquent.

Pour autant, ceci nous permet d'affirmer que l'algorithme développé n'est que très peu impacté par les contacts sur les faces pour la méthode classique, et ne l'est que peu aussi si les matériaux sont peu denses pour la méthode rapide.

2.6.4. Images de fibres cohérentes spatialement

Dans un dernier temps, on crée virtuellement des matériaux semblables à ceux étudiés. Pour cela, on décide d'un nombre de fils dans notre image, puis on génère autant de fibres de manière aléatoire. Par la suite, on crée de nouvelles fibres par voisinage des anciennes, en leur accordant une modification d'orientation faible et aléatoire (Figure 57).

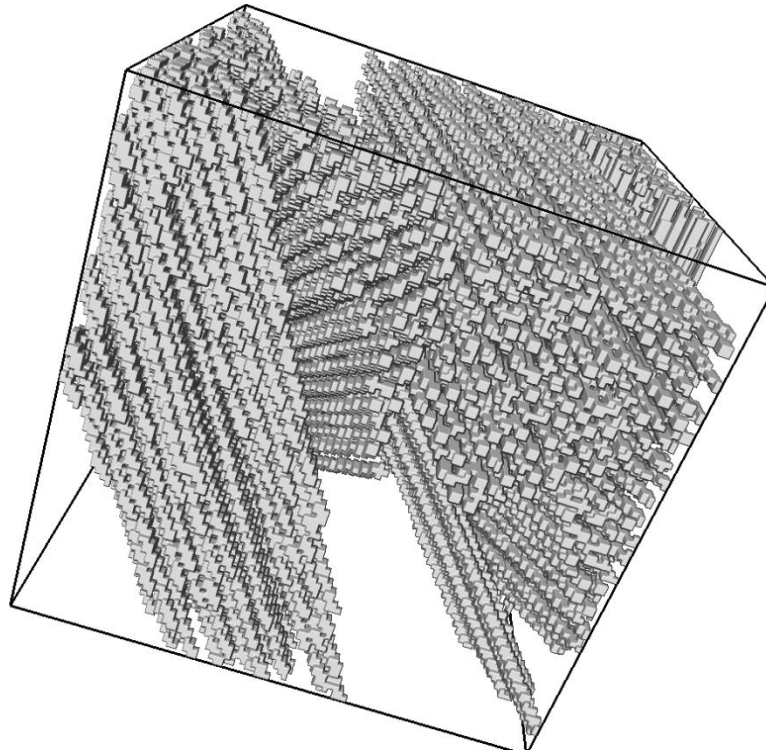


Figure 57 : Image d'un matériau virtuel utilisé, où les droites sont reliées en fils cohérents. Image de taille 400^3 , contenant 300 droites réunies en 6 fils.

Sur ces images, la détection des lignes se fait de manière semblable aux images aléatoires sans contact (en effet, les fibres étant toutes parfaitement remplies, la contrainte supplémentaire accordée par la connaissance des orientations possibles n'est jamais utilisée) et le temps d'exécution est le même pour des images avec le même nombre de fibres. Sur ces images (Figure 58), on effectue aussi la détection de fils par la méthode explicitée dans la partie 2.4.3. On constate que le *clustering* obtenu par la méthode des K-means sépare bien les différents fils entre eux, c'est-à-dire que chaque cluster ne contient que des fibres d'un seul fil. Cependant, la classification est aussi trop forte, des fils étant découpés en plusieurs classes distinctes. Ceci est imputable à deux causes en particulier :

- La première, qui ne nous dérange pas par la suite, est liée à la construction des images virtuelles. Les fibres de ces images sont créées en choisissant aléatoirement une fibre, et en essayant ensuite de lui accoler une voisine avec une différence d'orientation aléatoire. Ainsi, l'orientation des fibres suit des propriétés locales plutôt que des propriétés associées au fil entier. Une « forte » déviation sur une fibre peut entraîner une séparation perceptible du fil en deux parties proches mais néanmoins distinctes.

- La seconde, plus gênante, est liée à la difficulté de choisir une fonction adaptée dans l'initialisation de la classification grâce à la méthode des noyaux. Il est en effet ardu de déterminer à l'avance, ou algorithmiquement, une fonction adaptée à l'image étudiée.

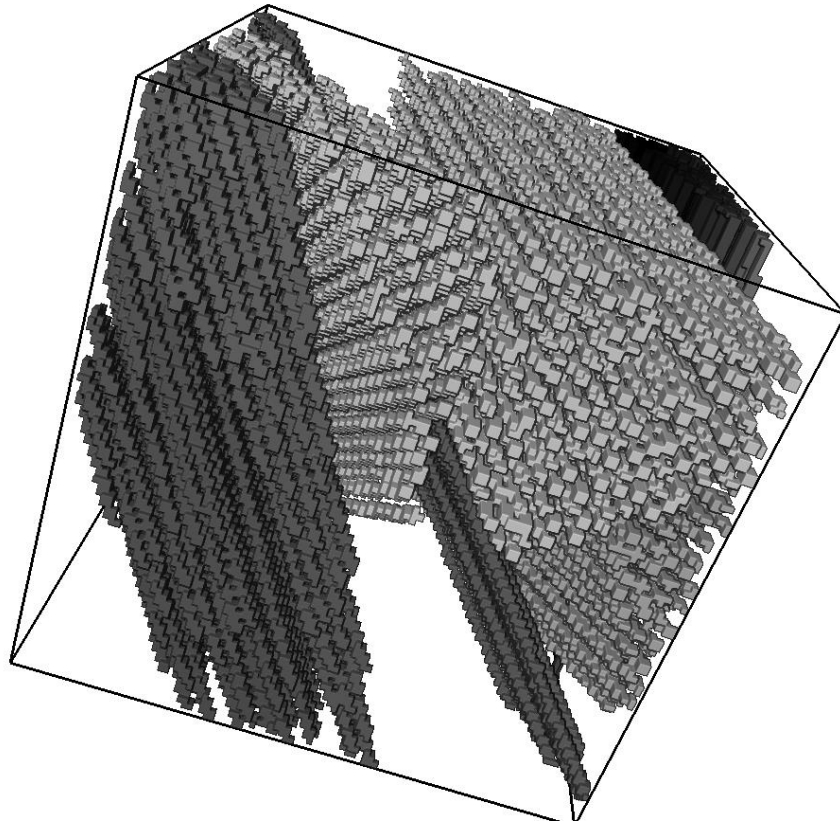


Figure 58 : Résultat de l'algorithme de détection de lignes sur l'image présentée en Figure 57.

On préfère tout de même naturellement obtenir trop de classes (i.e. trop d'information, ce que l'on peut corriger) que pas assez (absence d'information, incorrigible). La méthode de fusion de fils présentée dans la partie 2.4.3 est utilisée, et elle permet finalement la fusion des fils (Figure 59). La qualité du résultat peut être observée sur la Figure 60, qui représente l'appartenance aux fils des voxels des fibres de départ.

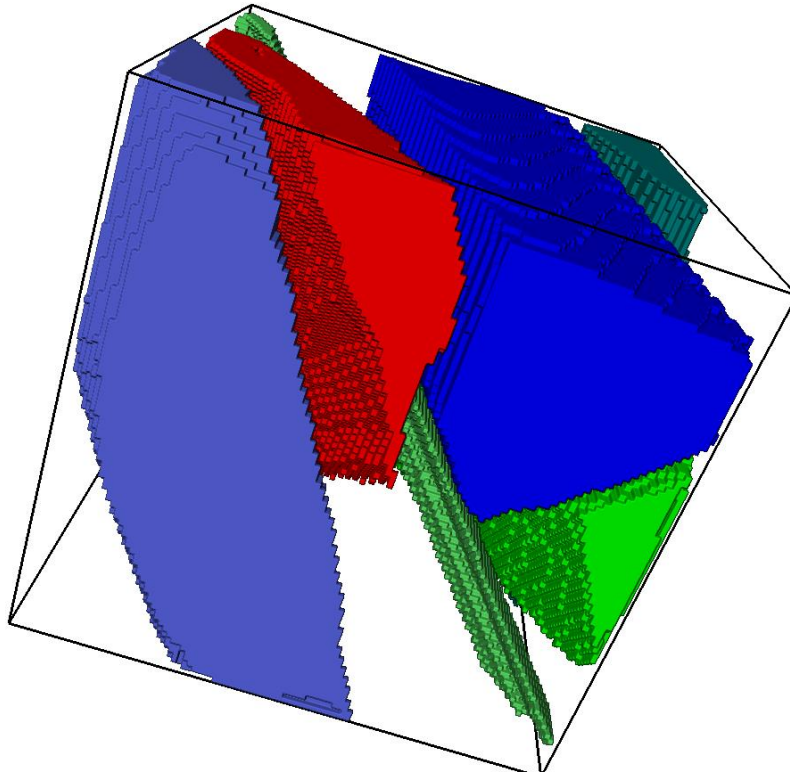


Figure 59 : Résultat de la classification en fils sur les fibres détectées présentées sur la Figure 58. Les 6 fils sont bien tous retrouvés.

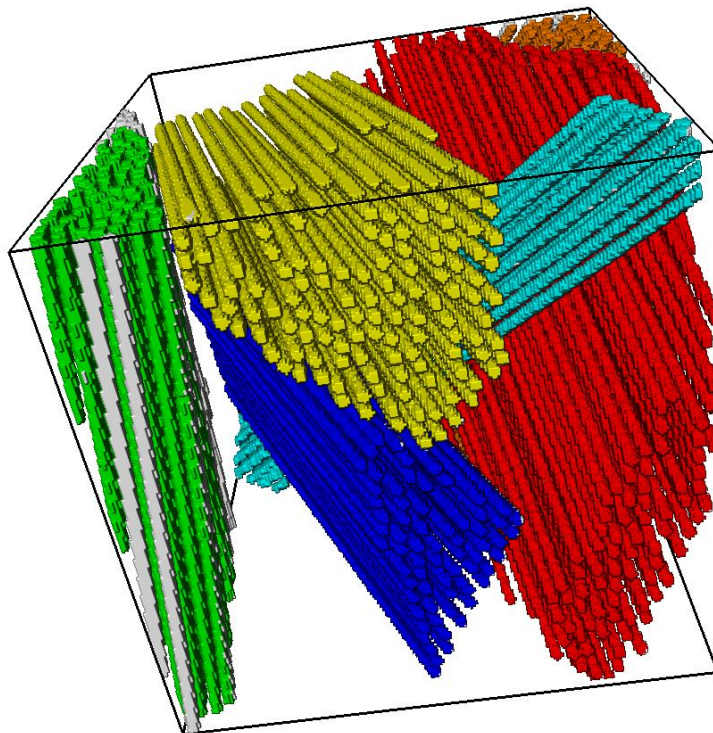


Figure 60 : Superposition des droites de départ et des fils retrouvés. Les voxels de couleur appartiennent à une droite et un fil, les voxels gris n'appartiennent pas à un fil retrouvé. On voit que près des bords, le fil peut ne pas englober toutes les fibres (fil vert à gauche)

3. Recherche de sphères : applications à des mousses de silice et polyuréthane

3.1. Introduction et propriétés des images utilisées

Un deuxième type de matériau est étudié dans ce manuscrit. Il s'agit de matériaux poreux de type mousses ou avec inclusions, présentant deux phases et des structures détectables approchant des sphères. Les structures peuvent cependant varier selon les matériaux et le type d'acquisition. Les images peuvent en effet présenter deux phases distinctes, une formant une matrice dans laquelle baigne des boules pleines de la deuxième phase, comme présenté sur les images de mousses de polyuréthane ci-dessous (Figure 61). Dans d'autres cas, une phase peut constituer l'intérieur et l'extérieur des sphères, une coquille étant constituée de la deuxième phase, comme présenté sur les images de mousses de silice ci-dessous (Figure 62 et Figure 63). À un passage de gradient près sur les images de boules pleines, les deux types d'image peuvent être représentés par des images de sphères creuses (en effet, pour une image indicatrice d'une boule, la norme du gradient numérique calculé à partir de masques communs sera non nulle uniquement sur les voxels du bord de la sphère). Ainsi, on pourra s'intéresser au problème de détection de sphères creuses plutôt qu'à celui de boules pleines dans toute la suite.

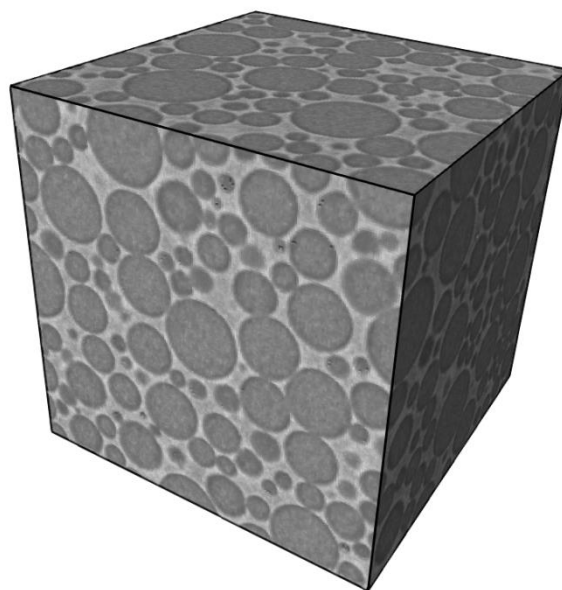


Figure 61 : Image tomographique d'une mousse de polyuréthane. Image de taille 300^3 , avec une résolution de $5 \mu\text{m}/\text{vx}$.

Les différents matériaux étudiés présentent tous des sphères de tailles diverses, suivant une certaine distribution. Contrairement aux images de fibres, les objets à rechercher ici sont de taille variable, ce qui peut influencer sur le choix et la performance des algorithmes utilisés.

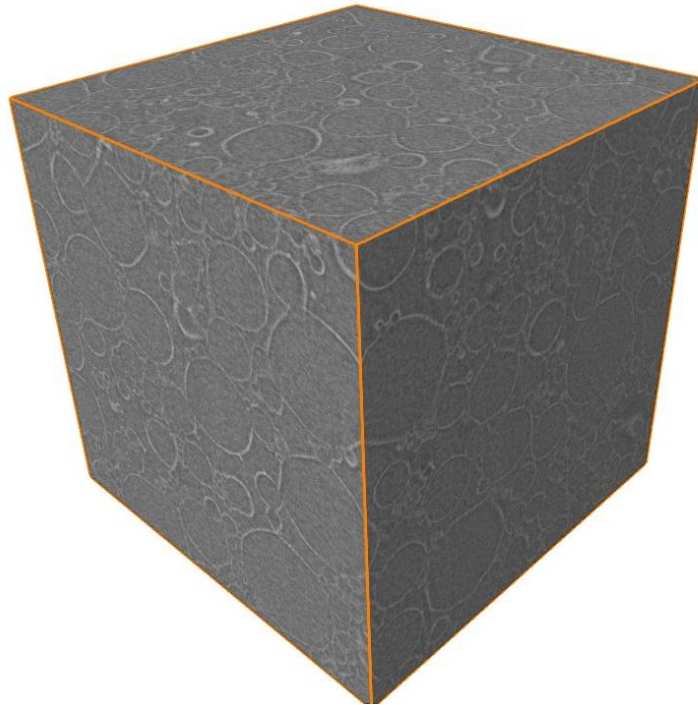


Figure 62 : Image tomographique d'une mousse de silice. Image de taille 700^3 , avec une résolution de $5 \mu\text{m}/\text{vx}$.

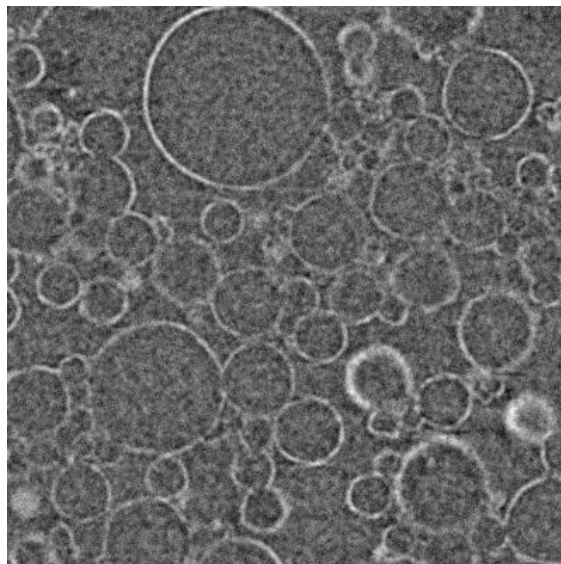


Figure 63 : Coupe de l'image de mousse de silice présentée sur la Figure 62. On constate la présence de bruit, des structures difficiles à repérer à certains endroits, et pas parfaitement sphériques.

Le but de l'étude de ces images est de détecter l'ensemble des sphères présentes dans le matériau. La connaissance de ces sphères peut ensuite nous permettre de vérifier plusieurs points sur les mousses :

- Vérification du procédé de fabrication : il s'agit de comparer les distributions attendues et réelles de taille de sphères dans le matériau afin de valider la manière de le produire. Dans le cas des matériaux sous forme de billes, la fabrication est souvent effectuée en visant une certaine distribution, mais le remplissage des interstices entraîne souvent des constructions différentes. La répartition aléatoire des sphères peut aussi être vérifiée, afin de voir si le matériau ne présente pas de direction privilégiée, par exemple.
- Simulation thermique et mécanique : grâce à la connaissance de la structure du matériau, il est possible de représenter celui-ci plus simplement (par la liste des sphères plutôt que voxel par voxel), et ainsi de le modéliser plus facilement. A partir de cette modélisation, il devient plus aisé de procéder à des calculs physiques sur de telles structures.

De fait, on cherche donc à détecter, sur ce type d'images, deux informations que sont la localisation et la taille des structures sphériques.

3.2. Contraintes et facilités accordées par le contexte et les images

De la même manière que pour les matériaux fibreux, la structure géométrique du matériau et les propriétés des images tomographiques obtenues vont de nouveau orienter le choix des méthodes utilisées pour le traitement des images, en fonction des contraintes et facilités que celles-ci accordent. Cependant, ici, différents types de matériaux seront étudiés, et, par conséquent, les inconvénients et avantages fournis par les images varient. Il est alors nécessaire de concevoir des solutions soit génériques (c'est-à-dire fonctionnant pour de larges choix d'images), soit modulables (avec un cœur commun et des parties variables selon le matériau étudié). Les solutions génériques, bien qu'apportant un confort supérieur à l'utilisateur, en raison de la démarche plus simple, donnent aussi des résultats souvent moyens. On préfère ici développer un éventail de solutions permettant de s'adapter aux avantages et inconvénients de chaque image.

On commencera par s'intéresser aux images de type mousse de polyuréthane présentées dans la partie 3.1. Ces images présentent des avantages très conséquents pour très peu d'inconvénients, ce qui en fera un bon point de départ pour la recherche d'une méthode centrale utilisable pour toute détection de sphères. Les quelques inconvénients de ces images sont :

1. La taille des images est encore une fois importante, vers le milliard de voxels. Comme dans le cas de la détection de fibres, cette taille va limiter le champ des techniques utilisables aux méthodes peu gourmandes en temps et en mémoire. On notera cependant que cet inconvénient est finalement inhérent aux images tomographiques de matériaux au sens large, ce qui fait de cette taille plus une caractéristique générale qu'un inconvénient d'un type d'image en particulier.
2. Les structures sphériques sont de petite taille. De fait, les plus petites structures présentes sont constituées de quelques dizaines de voxels tout au plus (pour des structures génériques comportant au moins un millier de voxels), ce qui peut rendre difficile leur détection précise, autant pour leur position que leur rayon. Aussi, il peut être difficile de séparer des structures de très petite taille présentes dans l'image du bruit.

3. La densité de sphères des matériaux est assez importante, et il peut être difficile de détecter des petites sphères si celles-ci sont coincées entre de grandes structures à cause d'un effet d'échelle.

Concernant les avantages, on notera :

1. Malgré la taille de l'image, le volume élémentaire représentatif des images de mousse polyuréthane étudiées ici est assez petit. Ainsi, il suffit d'étudier des sous-images de taille raisonnable pour obtenir par extrapolation le comportement et les propriétés des images plus importantes.
2. Les sphères sont toutes de taille limitée. Ainsi, la complexité de la détection d'une sphère dépendant de son rayon, celle-ci sera faible. Les méthodes utilisées pour ce type de matériau seront donc plutôt rapides.
3. La limitation de taille permet aussi de déduire une limitation de l'écart maximal de rayon, ce qui permet de restreindre au départ un des paramètres des sphères.
4. Les structures sont très fidèlement représentées par des sphères. Ainsi, les méthodes utilisées ici n'ont même pas réellement besoin d'être robustes par rapport aux déformations.
5. Les images sont remarquablement nettes. Ainsi, le bruit est quasi-inexistant, et la détection s'en retrouve donc facilitée.

L'ensemble des avantages et contraintes accordées par les images de polyuréthane augure d'une détection facile des structures sphériques, d'autant plus que l'inconvénient 1 est complètement annulé par les avantages. Auquel cas, le choix de la méthode ne pourrait se jouer que sur un critère de vitesse.

Cependant, si l'on s'intéresse maintenant aux images de mousse de silice, les caractéristiques de celles-ci ne sont pas aussi appréciables. Du côté des inconvénients, on en recense beaucoup plus :

1. La taille des images est toujours importante, vers le milliard de voxels. Cependant, cette fois, le volume représentatif de l'image est beaucoup plus important que pour les mousses de polyuréthane, ce qui empêche de corriger ce problème par la considération d'une sous-image de petite taille.

2. Les structures sphériques sont de tailles très diverses. Ceci cause de nombreux problèmes, autant temporellement, où la détection de grandes sphères ne sera pas rapide, que pour les résultats, étant donné la difficulté à détecter de très petites sphères.
3. La densité de sphères des matériaux est assez importante. Il peut être difficile de détecter des petites sphères si celles-ci côtoient des très grandes structures.
4. La sphéricité des structures est toute relative. Dans de nombreux cas, ces formes s'approchent plutôt de polyèdres à multiples côtés. La non-sphéricité pose un certain nombre de problèmes, allant de la difficulté de détection des structures en cause jusqu'à l'impossibilité de détection dans le voisinage à cause de formes trop étirées.
5. Le bruit sur les images de départ est important, ce qui ne facilite pas non plus la tâche de segmentation avant la détection.

Cette fois-ci, aucun avantage ne peut réellement contrebalancer ces défauts, à part le fait que les structures soient bien creuses.

À partir de ces caractéristiques sur les images de mousses, il est maintenant nécessaire de comparer et de choisir une méthode permettant de répondre à la problématique définie dans la partie 3.1.

3.3. Avantages et Inconvénients des méthodes

Afin de déterminer les adaptations envisageables à l'algorithme central pour traiter les différents types d'images, il est d'abord nécessaire de fixer la méthode qui sera utilisée. Différentes possibilités existent pour l'algorithme de détection de sphères, en particulier celles présentées dans les parties 1.2.3.3 et 1.2.4.2 :

1. Une méthode fondée sur la **Transformée de Hough Sphérique** est utilisable. La Transformée de Hough présente l'avantage de réussir à retrouver assez fidèlement des structures sphériques même si celles-ci sont incomplètes. Cependant, la méthode « simple » présente le désavantage de se comporter assez mal face aux déformations, qui ont tendance à provoquer des étalements de pics non-négligeables sur les accumulateurs de Hough.
2. Une méthode utilisant le **Tenseur de Structure**. La détection de sphères à partir d'informations locales est relativement aisée, le gradient d'un point d'une sphère pointant toujours vers son centre. Le tenseur de structure semble donc être une méthode toute désignée pour ce type de problème. La méthode ne nécessite pas non plus de segmentation précise, un seuillage et/ou un gradient local suffisant à déterminer les points d'intérêts. Cependant, le tenseur de structure ne se comporte pas idéalement sur des images au bruit très marqué, comme les images de mousse de silice qui sont étudiées ici.

L'algorithme fondé sur le tenseur de structure est finalement assez général, mais semble plutôt bien adapté à la détection de sphères. De plus, sa complexité algorithmique reste toujours appréciable quelle que soit son utilisation.

3. Les hypothèses que l'on a sur notre image nous permettent aussi de réfléchir de nouveau à une **méthode géométrique**. La sphère étant évidemment une structure géométrique élémentaire, les propriétés particulières de cet objet peuvent être utilisées pour déduire des méthodes directes de détection de sphères.

De la manière la plus simple possible, une sphère correspond à un ensemble de points à égale distance d'un point particulier, son centre. Ce centre est aussi un maximum local de la distance à la sphère. Par conséquent, la détermination des

maximums locaux des cartes de distance permet naturellement de retrouver les centres des sphères, et la valeur de la carte en ces maximums nous donne le rayon.

Cette méthode utilise avantageusement certaines propriétés des images étudiées, en particulier l'absence d'inclusions à l'intérieur des structures sphériques (qui modifieraient de façon démesurée la carte de distance).

Intéressons-nous maintenant aux **complexités respectives de ces différents algorithmes**. La **Transformée de Hough** reste toujours une méthode à complexité élevée, autant en temps qu'en mémoire. Dans le cas classique, avec une image de taille L^3 et des sphères dont le rayon maximal est de R , la taille de l'accumulateur à considérer sera de L^3R (voire de $(L + 2R)^3R$ si l'on recherche aussi des sphères à l'extérieur de l'image ayant une influence sur l'intérieur). L'accumulateur demandera donc R fois plus de mémoire que l'image de départ, ce qui peut se révéler très vite limitant si R et L sont assez grands.

D'un point de vue temporel, l'algorithme devra effectuer un nombre d'opérations proche de $\frac{4}{3}\pi R^3$ pour chaque point considéré. Le nombre de points à considérer sur l'image se comportera en $O(L^3)$. Ainsi, la complexité de la phase de remplissage de l'accumulateur est de $O(L^3R^3)$, ce qui est linéaire avec la taille de l'image. Les méthodes probabilistes comme la RHT (voir partie 3.4.1) permettent d'abaisser la constante de temps, mais ne permettent pas de gain sur la complexité asymptotique de l'algorithme.

La méthode fondée sur le **tenseur de structure** doit aussi considérer chaque point des structures, c'est-à-dire un $O(L^3)$ points. A partir de chacun de ces points, il est alors nécessaire de calculer le tenseur de structure et de le moyennner sur un voisinage de rayon ρ . A partir de cela, il devient alors possible en temps constant de calculer le rayon de courbure et la direction principale du gradient, qui nous permettent alors de retrouver le centre. La **complexité asymptotique de l'algorithme est donc de $O(L^3\rho^3)$** .

La dernière **méthode, géométrique**, calcule tout d'abord la carte de distance sur l'image considérée, ce qui peut se faire de manière linéaire en un nombre de passes fini, c'est-à-dire en $O(L^3)$. La recherche des maximums est aussi linéaire, ce qui nous donne finalement un algorithme en $O(L^3)$. On peut aussi, pour valider les résultats, vérifier

l'existence d'une sphère au rayon trouvé. Dans ce cas, la **complexité asymptotique de la méthode est en $O(L^3 R^2)$** .

Finalement, on peut affirmer qu'au niveau de la complexité asymptotique, la méthode géométrique simple est la meilleure, suivie de la méthode géométrique avec vérification puis de la méthode de Hough. La méthode de tenseur de structure se situe derrière la géométrie simple, mais sa position par rapport aux deux autres dépend du rapport entre ρ et R . En général, le voisinage considéré pour le tenseur de structure est de loin inférieur au rayon maximal des sphères, ce qui donne finalement un encadrement temporel :

(géométrie) < géométrie vérifiée ? (tenseur de structure) < (HT)

Sur des images très simples à traiter (comme les mousses de polyuréthane), seule la vitesse sera déterminante sur le choix de l'algorithme. On choisit donc la méthode géométrique.

Pour valider le choix de la méthode géométrique comme base du traitement, il est maintenant nécessaire de s'intéresser au comportement que l'on peut attendre des différents algorithmes sur les autres types d'images étudiées, comme les mousses de silice.

Commençons par la **Transformée de Hough**. Cette méthode présente les avantages suivants :

- Capacité à détecter des sphères incomplètes si la partie restante est significative, et cela quelle que soit la répartition des morceaux restants.
- Découlant directement de la propriété précédente, la transformée de Hough est capable de détecter des structures intersectant l'image mais dont le centre est extérieur.
- Capacité à détecter des structures sphériques même si celles-ci contiennent des inclusions. Le voisinage d'une sphère influe peu sur son traitement, c'est-à-dire que l'accumulateur ne sera que très peu modifié au voisinage du pic la représentant. Par déduction, la transformée de Hough est aussi robuste face au bruit.

- Robustesse face aux légères déformations. Celles-ci provoquent de l'étalement de pics, mais ceux-ci restent détectables. Leurs positions précises peuvent devenir incertaines, mais l'application de filtres (comme effectué avec la Transformée de Hough Floue (Han, et al., 1994)) permettent d'y remédier.

Du côté des défauts, on notera :

- Absence de différenciation entre l'intérieur et l'extérieur des sphères. Ceci peut entraîner, pour des images denses de sphères approximatives, des détections de faux positifs (voir paragraphe suivant).
- Surévaluation des grandes sphères par rapport aux petites en raison du nombre de points supérieurs sur les premières. Ceci est compensable facilement sur l'intérieur de l'image par une division simple, moins sur les bords, où le dénominateur dépend de la position du point dans l'accumulateur.

Le problème de détection des faux positifs mérite d'être vu en détail, celui-ci étant d'une importance capitale dans la qualité des résultats et donc dans le choix de la méthode utilisée.

Dans le cas d'images de mousses où les sphères sont déformées ou incomplètes, il est nécessaire, pour des méthodes procédant à des calculs sur le nombre de points appartenant à la sphère (comme la transformée de Hough), de diminuer le taux d'acceptation. Cela signifie que l'on accepte alors des sphères si moins de points vérifient les bonnes propriétés (pour la transformée de Hough par exemple, on diminuera la valeur minimale admise pour les pics de l'accumulateur). Cependant, en acceptant des résultats moins solides, on augmente aussi le risque d'erreurs. Dans le cas où la densité de sphères est faible, le problème ne se pose pas, la probabilité de détecter un faux positif est faible. Mais, dans le cas d'une forte densité, il est alors tout à fait possible de voir apparaître des faux positifs issus de structures fortuites approchant des sphères d'une façon indétectable pour l'algorithme. La Figure 64 explique la cause du problème : dans cette image, les cercles noirs sont présents sur l'image, et, avec leur disposition, si le critère d'acceptation est assez faible, alors le cercle rouge peut être accepté alors qu'il n'existe pas.

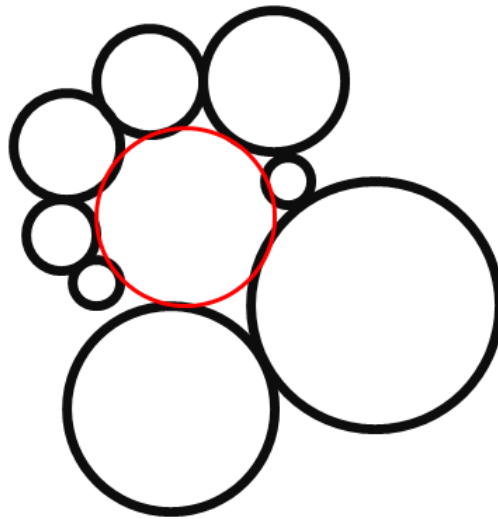


Figure 64 : Schéma explicatif de la détection de fausses sphères. Ici, les cercles noirs sont réels, et leur disposition particulière fait que le cercle gris, inexistant au départ, peut être détecté si les critères d'acceptation sont assez bas.

Un constat très important peut être fait : **Les fausses sphères sont toujours issues d'un ensemble de vraies structures qui leur sont extérieures et tangentes.** En effet, la détection d'une fausse sphère ne peut se faire qu'avec la présence d'un ensemble significatif de points considéré à un rayon donné et avec leur absence avant ce rayon.

Le problème des faux positifs doit nécessairement être résolu pour obtenir des résultats exploitables sur des images de structures sphériques déformées et denses, comme le sont les mousses de silice.

Revenons maintenant aux comparaisons des méthodes, et à celle fondée sur le **Tenseur de Structure**. Ses avantages sont :

- Capacités à détecter des sphères incomplètes tant que la partie restante est assez grande, peu importe la disposition des parties manquantes.
- Comme explicité dans les avantages de la méthode de la Transformée de Hough, la propriété précédente entraîne la capacité à retrouver des sphères dont le centre est à l'extérieur de l'image.
- Capacités à détecter les sphères même si celles-ci présentent des inclusions.

- Pas de problème de fausses sphères. En effet, le calcul du rayon de courbure permet de distinguer l'intérieur et l'extérieur des sphères. De fait, il n'existe normalement pas de faux positifs car rien n'y est détecté.

Cependant, du côté des défauts, il est à noter :

- Gros problèmes de calculs face aux déformations de sphères de type « approximation polyédrique ». Dans de tels cas, il devient impossible de déterminer le côté intérieur de l'extérieur (Figure 65). Le calcul du rayon en est aussi affecté, et les résultats peuvent devenir totalement incohérents.
- Même face à d'autres types de déformation, le tenseur de structure se comporte assez mal. Son aspect très local et calculatoire fait que les déformations ne sont pas compensées par ce qui peut se passer plus loin.
- La méthode ne nécessite pas de segmentation préalable, mais se révèle en contrepartie assez sensible au bruit. Des filtrages préventifs peuvent être effectués, mais ceci ne sont pas toujours suffisants, et augmentent le temps de calcul.
- De la même manière, les jonctions entre sphères sont sources d'erreurs et d'incohérences.

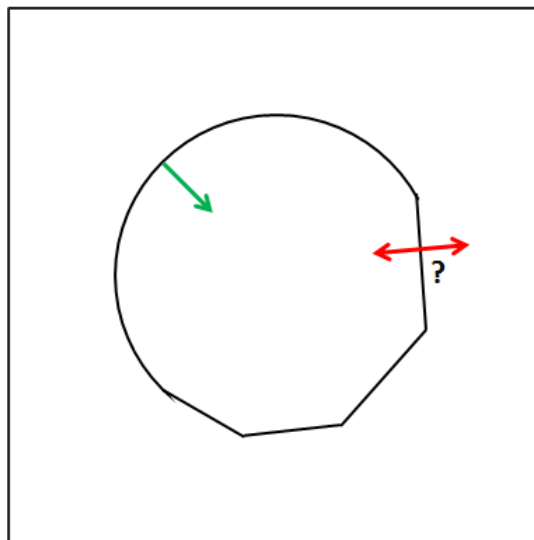


Figure 65 : Exemple d'incertitude due à la structure polyédrique de parties des structures. Ici, la flèche verte indique le sens du centre. La double flèche rouge indique pour sa part que sur les segments, il n'est pas possible de déterminer le sens du centre.

Le très gros avantage des méthodes utilisant le tenseur de structure est l'inexistence des problèmes de fausses sphères. Grâce à cela, il est possible de détecter des structures difficiles à observer via les autres méthodes sans rajouter d'erreurs dans les résultats. Cependant, il devient beaucoup plus difficile de détecter certains types de structures, en particulier si celles-ci sont localement planes. Dans de tels cas, il est possible de corriger ces problèmes en augmentant la taille du voisinage considéré. Mais d'une part, augmenter la taille du voisinage consiste à augmenter ρ , et donc fait exploser le temps de calcul, d'autre part, sur des images denses, il est fort probable que des structures proches finissent par interagir avec le point considéré et dénaturent donc le résultat. Cependant, pour les petites sphères, le ρ à considérer pour corriger les défauts est faible. De fait, le tenseur de structure est très performant sur les petites sphères.

Finalement, si l'on en vient à la **méthode géométrique**, on a pour avantages :

- Toujours la capacité à détecter des sphères incomplètes. Cependant, cette fois, il est nécessaire qu'aucun hémisphère ne soit vide, auquel cas on observe une fuite de la carte de distance vers l'extérieur, et l'absence de maximum.
- Résistance au bruit assez forte. Les variations locales de l'image n'ont que peu d'incidence sur cette méthode globale. Cependant, le bruit peut entraîner la présence d'inclusions dans les sphères.
- Résistance à la déformation. De la même manière que pour le bruit, la déformation n'implique que des modifications légères sur la carte de distance.

Du côté des inconvénients, on note :

- Problème des faux positifs : la carte de distance possède en effet des maximums entre les sphères.
- Incapacité à détecter des sphères dont le centre est exclu de l'image, la distance augmentant constamment dès que l'on s'éloigne de l'image.
- Incapacité à détecter les structures présentant des inclusions.
- Sous-évaluation des rayons, le maximum de la carte de distance ne correspondant qu'à la distance avec le point le plus proche.

On constate ici que le problème des inclusions est sans importance, vu que nos images n'en possèdent pas. La sous-évaluation des rayons peut être corrigé par la méthode géométrique avec vérification, qui recherche ensuite le rayon le plus probable pour la sphère. Finalement, le problème de détection des sphères externes n'est pas réellement un problème pour peu que l'on considère une image d'assez grande taille dont on éliminera finalement les bordures.

Si on considère simultanément les problèmes de temps et de qualité des résultats, on notera les points suivants :

1. La méthode géométrique « classique » est la plus rapide, mais aussi celle qui donne les moins bons résultats.
2. La méthode dérivée du tenseur de structure donne des résultats soit incomplets, soit en temps long.
3. La méthode géométrique avec vérification est rapide mais donne des faux positifs.
4. La méthode utilisant la Transformée de Hough est lente mais donne des résultats corrects, moyennant quelques faux positifs.

On peut constater aussi que la méthode de tenseur de structure est très performante sur les petites sphères, quand la méthode géométrique l'est sur les grandes (le temps de calcul de la carte de distance étant constant). De fait, on peut imaginer des algorithmes utilisant les deux méthodes pour des meilleurs résultats.

On utilise donc comme élément central la méthode géométrique. Sur les images de mousses de polyuréthane, elle est déjà suffisante pour offrir de très bons résultats. Pour les matériaux plus complexes, on choisit d'effectuer les trois phases décrites plus haut, c'est-à-dire un pré-traitement, un post-traitement et des variations de code.

L'enjeu du pré-traitement est donc de trouver un équilibre entre quantité d'ajout et temps de calcul. Celui-ci doit être rapide, afin de continuer à profiter de la vitesse de la méthode géométrique, et doit pour autant faire ressortir assez d'information de l'image pour faciliter le travail de la méthode centrale de détection en 3D. On en revient ici aux trois méthodes présentées en 3D, juste adaptées à la 2D :

- Transformée de Hough circulaire, de complexité en $O(L^2 R^2)$.
- Tenseur de structure, de complexité en $O(L^2 \rho^2)$.
- Méthode géométrique, de complexité en $O(L^2)$.

On choisit une méthode fondée sur la Transformée de Hough Circulaire pour enrichir notre image.

Concernant le post-traitement, celui-ci consiste surtout en une vérification des résultats. Il s'agit donc ici de réussir à distinguer les faux positifs. Ceci peut se faire de différentes façons :

- Grâce à des calculs de courbure, il est possible de savoir si le centre des sphères détectées est bien à l'intérieur ou à l'extérieur d'une structure sphérique. Cependant, le calcul de courbure et la vérification des sphères sont relativement lents.
- À partir de la carte de distance, il est possible de déterminer, selon la forme que prennent les lignes de niveaux à proximité du centre, si la structure de l'image proche correspond ou non à une sphère. Cette méthode n'est pas à proprement parler un post-traitement, mais vise le même but.
- À partir des points appartenant en même temps aux structures de l'image et aux sphères détectées, il est possible de suivre la différence entre l'image originale et le résultat afin de déterminer si celui-ci est localement cohérent.

Dans un tel cas, les trois méthodes sont cohérentes et présentent des avantages. Les implémenter toutes les trois pourrait mener à des résultats de qualité.

Au final, la méthode choisie pour la détection de sphères consiste en :

1. Segmentation de l'image de départ.
2. Suppression des éventuelles inclusions par détermination des régions connexes (voir partie 2.4.2) et suppression des régions de très petite taille.
3. Calcul de la carte de distance.
4. Pré-traitement de l'image par une détection avec une Transformée de Hough Circulaire .
5. Détermination des maximums de la carte de distance.

6. Elimination des maximums présentant des lignes de niveaux « irrégulières » à proximité.
7. Recherche de sphères à partir de la carte de distance et de l'image pré-traitée.
8. Post-traitement : vérification des sphères via les méthodes présentées ci-dessus.

3.4. Algorithme développé

Les algorithmes de segmentation et de calcul de la carte de distance ayant déjà été présentés respectivement dans les parties 2.4.1 et 2.4.2, on n'y revient pas dans cette partie. On suppose donc par la suite disposer d'une image binaire définie sur un ensemble $E = \prod_{i < 3} \llbracket 0 ; L_i \rrbracket \subset \mathbb{N}^3$, que l'on note $\mathbf{X} = \{X_e ; X_e \in \{0 ; 1\}, e \in E\}$, et d'une carte de distance Y issue de \mathbf{X} définie par :

$$\forall e \in E, Y_e = d(e, \mathbf{X}^{-1}(1))$$

avec d la fonction de distance euclidienne.

3.4.1. Pré-Traitement : Transformée de Hough Circulaire

Le pré-traitement s'effectue sur l'image \mathbf{X} , et a pour but de renvoyer une image \mathbf{X}' enrichie (c'est-à-dire telle que $\mathbf{X}^{-1}(1) \subset \mathbf{X}'^{-1}(1)$) que l'on utilisera par la suite en substitution à \mathbf{X} comme image pour vérifier la présence de sphères.

On choisit ici de pré-traiter plan par plan avec une transformée de Hough circulaire. La Transformée de Hough nous apporte plus de garantie sur les grands cercles que le tenseur de structure.

Soit une sphère discrète \mathcal{S} et un ensemble donné $D \subset E$. On notera $\tau_D(\mathcal{S}) = \frac{\text{card } \mathcal{S} \cap D}{\text{card } \mathcal{S}}$. De même, pour un cercle \mathcal{C} , on notera $\kappa_D(\mathcal{C}) = \frac{\text{card } \mathcal{C} \cap D}{\text{card } \mathcal{C}}$.

L'enrichissement par détection de cercles sur des plans présente des avantages comparés à une détection de sphères directes. En effet, si l'on suppose que l'on cherche à détecter une structure sphérique incomplète D approchable par une sphère \mathcal{S} , alors les structures circulaires résultant de l'intersection par un plan vérifieront $\overline{\kappa_D} = \tau_D$, c'est-à-dire que le ratio de points de la structure appartenant au cercle sera en moyenne égal au ratio pour la sphère.

Cela veut dire que dans le cas général, il existe des structures circulaires contenant bien plus de points, et qui sont donc bien plus simples à détecter. De plus, les structures incomplètes sur les images étudiées ont souvent une forme de calotte de grande taille. Ainsi,

la répartition des points est inhomogène, la variance de κ_D est importante. Il existe donc des cercles pour lesquels κ_D est grand devant τ_D , et qui sont donc plus facilement détectables.

Algorithmiquement, calculer des intersections avec des plans dans de très nombreuses directions possibles est fastidieux car l'image correspond à un ensemble discret : ceci implique des directions de discrétisation, correspondant aux axes d'un repère orthonormé. On se contente donc de calculer les intersections uniquement pour les plans orthogonaux à l'un des axes, pour lesquels aucun travail de projection n'est à effectuer. Pour un plan d'équation $x = k$ (respectivement y, z), on appelle plans voisins les deux plans d'équations $x = k + 1$ et $x = k - 1$ (respectivement y, z).

On constate que dans les images présentées, les sphères les plus souvent sujettes à l'incomplétude et à la déformation sont celles de grande taille. Ceci est en partie explicable par le fait qu'une grande déformation sur une très petite sphère ne sera pas toujours suffisante pour déplacer la surface de la structure ne serait-ce que d'un voxel. Les grandes sphères étant bien mieux résolues, toute légère déformation implique très rapidement des modifications importantes et mesurables sur l'image. Ainsi, la méthode de pré-traitement est majoritairement destinée aux sphères de grande taille, maillons faibles de la détection.

Cela implique que les structures visées par le pré-traitement varient localement assez peu, deux plans parallèles très proches intersectant une grande sphère formant deux cercles de rayons semblables (Figure 66). Ainsi, si on détecte un cercle sur un plan donné, on peut envisager que les deux plans directement voisins puissent eux aussi présenter des cercles de rayons équivalents à un voxel près. Ainsi, on effectue des détections uniquement tous les trois plans, et on enrichit ensuite l'image non pas d'un unique cercle, mais d'une couronne cylindrique basée sur ce cercle, de hauteur 3 voxels, et d'épaisseur 3 voxels.

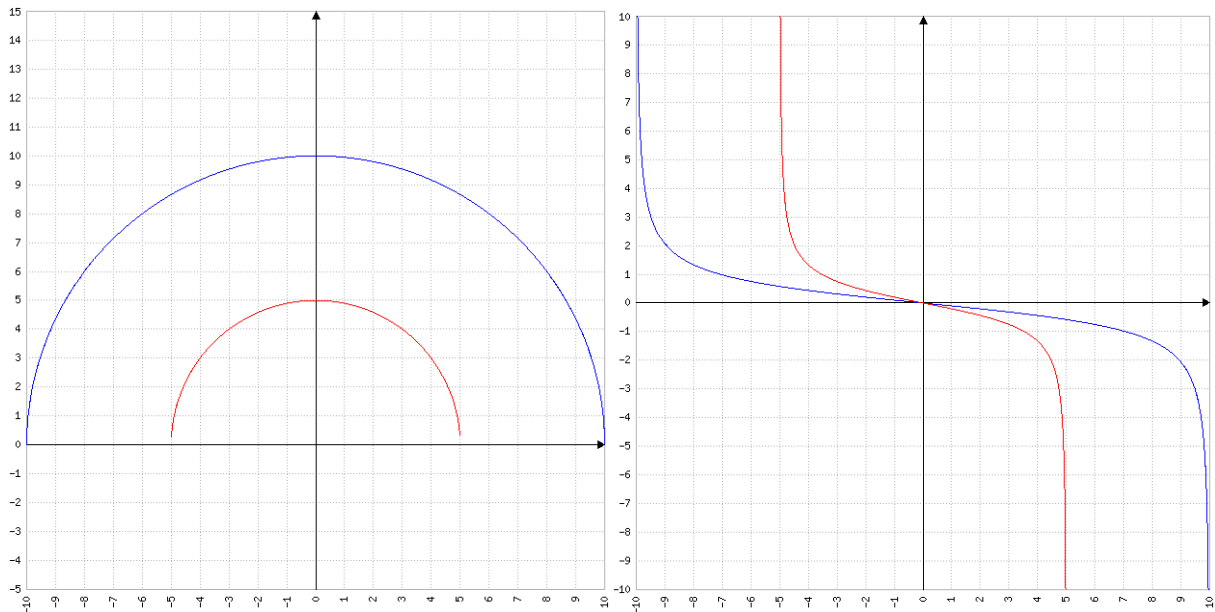


Figure 66 : A gauche, graphique de l'évolution du rayon du cercle résultant de l'intersection d'une sphère centrée en O avec un plan d'équation $z = k$ (en bleu pour une sphère de rayon 10, en rouge de rayon 5). A droite, dérivée de l'évolution. La dérivée pour le rayon 5 est, en valeur absolue, toujours plus grande que pour le rayon 10. La variation du rayon du cercle intersecté y est donc plus forte.

Maintenant que l'on a posé les modalités du pré-traitement, intéressons-nous de plus près à la transformée de Hough elle-même. Celle-ci est décrite de manière générale dans la partie 1.2.3. Il existe différentes variantes de la Transformée de Hough « classique » permettant de gagner du temps sur son exécution. La Transformée de Hough circulaire classique a une complexité algorithmique en $O(L^2R^2)$, ce qui nous donne donc un pré-traitement en $O(L^3R^2)$, ce qui est bien linéaire avec la taille de l'image, L et R étant indépendants. Le premier objectif du pré-traitement est donc atteint (voir plus haut). Toutefois, suivant la taille de l'image, la constante de temps est donc influencée quadratiquement par R .

On cherche donc à réduire cette constante de temps en considérant des méthodes plus rapides utilisant la Transformée de Hough.

On va utiliser la méthode de la Transformée de Hough Probabilisée (PHT) (Kiryati & Bruckstein, 1991). Sur le principe, la PHT est quasiment équivalente à la HT normale, excepté :

- le fait que l'on n'effectuera pas la phase de vote sur l'ensemble des points de l'image à considérer, mais sur un nombre plus petit $m < M$.

- La phase de détection ne se fait plus par la recherche d'un point maximal de l'accumulateur mais par la recherche d'une région de l'accumulateur. En effet, la discrétisation implique un étalement des pics, et, dans le cas de la PHT, afin de réduire au maximum le nombre de points à considérer, il est utile de bien récupérer l'ensemble du pic et pas uniquement son sommet pour le discriminer plus facilement du bruit (Kiryati & Bruckstein, 1991).

La phase de calcul de l'accumulateur est alors accélérée d'un facteur $\frac{M}{m}$. Cependant, il est à noter que le ratio $\frac{M}{m}$ est très fortement dépendant des images à traiter. L'utilité de la PHT est donc très variable : il est nécessaire de prévoir convenablement la valeur maximale que peuvent avoir les pics correspondant à des faux positifs afin de minorer le seuil d'admissibilité (Figure 67).

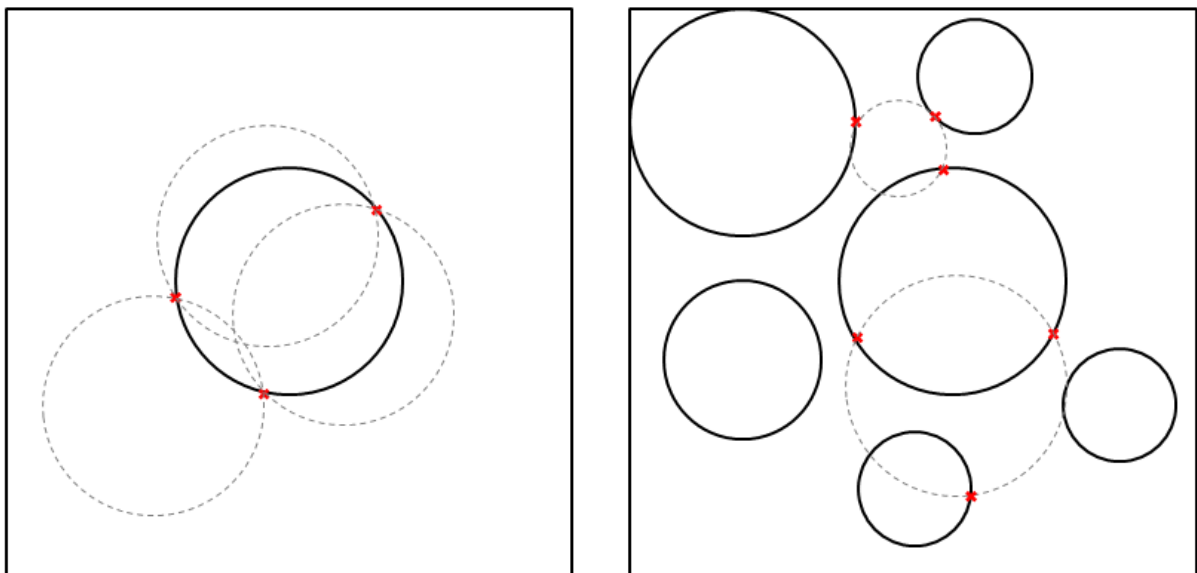


Figure 67 : Schéma explicatif de la PHT et de l'évolution du nombre de points minimal. A gauche, dans le cas d'un cercle unique, trois points suffisent pour que la PHT puisse le caractériser, tous les autres cercles sur l'accumulateur (en gris pointillé) ne passant pas par les trois points. Dans le cas de droite, représentant une image avec plusieurs cercles (traits pleins), trois points par cercle ne sont plus suffisants, les cercles en pointillés étant aussi acceptables par la PHT.

On constatera cependant (Kiryati & Bruckstein, 1991) que la PHT est une méthode très performante dans le cas de la détection d'une structure unique. Le bruit ou la présence d'autres informations de formes différentes dans l'image ne rend pas la détection difficile, même avec un échantillon de points réduits.

Cependant, cette méthode choisissant les points complètement au hasard, il est toujours probable d'obtenir une mauvaise répartition de ceux-ci, pouvant conduire à une absence de détection de certaines structures. Dans les faits, on procède donc à plusieurs tirs aléatoires avant de décider d'arrêter ou non l'algorithme, ce qui implique donc le choix d'une **condition d'arrêt** (Yla-Jaaski & Kiryati, 1994) (Shaked, et al., 1996).

Des méthodes dérivées de la PHT permettent d'améliorer les résultats tout en conservant une rapidité proche de la PHT. Un algorithme itératif développé par Galambos, Matas et Kittler (Galambos, et al., 1999), la PHT Progressive (PPHT), permet de détecter avec la même précision que la HT standard les cercles tout en réduisant le nombre de fausses détections. Cette méthode est fondée sur une actualisation de l'image après détection. Pour commencer, l'accumulateur est calculé via la PHT classique. Si l'on détecte une structure présente à un **seuil convenable** (afin de s'assurer qu'il ne s'agit pas de bruit), alors on la prend en compte dans le résultat, et on supprime tous les points qui y appartiennent. Puis on rajoute une nouvelle PHT, et ainsi de suite jusqu'à épuisement des points. On constatera donc l'existence ici d'une condition d'arrêt « naturelle », qui permet de se libérer d'un biais présent dans la PHT.

Cette méthode élimine donc de l'image les structures déjà détectées, ce qui permet :

- De réduire le nombre de fausses détections.
- D'améliorer la rapidité de l'algorithme. En effet, la HT considère tous les points, alors qu'ici, grâce à la PHT, un nombre important de points de chaque structure détectée sera éliminé sans avoir participé à la phase de vote.

La PPHT allie donc les avantages de la HT standard (détection la plus exhaustive possible) avec ceux de la PHT (détection plus rapide) tout en réduisant le nombre de faux positifs. Cependant, pour des images très denses, la rapidité de la PPHT n'est plus aussi valorisable face à la HT standard.

Dans notre cas, il est donc nécessaire, pour pouvoir utiliser la PHT, de déterminer le nombre maximal de points que peut contenir un faux positif. Il est déjà aisé de constater que ce nombre dépend du rayon du faux positif : en effet, si l'on prend un faux positif correspondant au rayon minimal des cercles présents, les points l'engendrant doivent être

issus de cercles de rayon au moins égal, ce qui leur laisse peu de place. De façon générale, le nombre de points maximal pouvant entraîner un faux positif correspond au nombre maximal de cercles de petite taille qui peuvent simultanément toucher le faux positif multiplié par le nombre de voxels en contact dans de tels cas de tangence sur un domaine discrétisé. Pour un rayon minimal de r et un faux positif de rayon R , le nombre de cercles maximal pouvant côtoyer simultanément le faux positif est de :

$$n(r) = \left\lfloor \frac{\pi}{\text{atan}\left(\frac{r}{R+r}\right)} \right\rfloor$$

Finalement, le nombre de cercles se comportant en $O(L^2)$, il n'y a pas de gain en complexité asymptotique. Cependant, la PHT permet de gagner, dans le pire des cas, un facteur non négligeable sur la détection par rapport à la HT standard.

La méthode de Hough probabilisée permet de rendre moins aigus les problèmes courant de la transformée de Hough standard, mais celle-ci repose toujours sur le même fondement de l'association d'un point d'une image vers un ensemble (généralement de « grande taille ») de points de l'espace de paramètres. De fait, la PHT ne corrige pas réellement le problème majeur de la HT standard, qui consiste en un calcul de l'accumulateur long et fastidieux.

Toutefois, cette idée de réduction du nombre d'éléments à considérer de manière parfaitement aléatoire ouvre tout un champ de possibilités sur d'autres variantes normalement encore plus gourmandes en calcul, mais pour lesquelles l'optimisation offerte par un jeu de sélection aléatoires s'avère bien plus grande que pour la HT standard.

La méthode de la Transformée de Hough aléatoire (RHT pour Randomized Hough Transform) est de celles-ci. Avant de poser le fondement de cette méthode, on commencera par noter que :

- la méthode « brute » de reconnaissance de formes, la plus lourde en calculs, fait un mapping convergent *many to one*, i.e. associe un ensemble de points de l'image à une et une seule forme.

- La méthode de Hough, en revanche, effectue un mapping divergent *one to many*, c'est-à-dire qu'elle associe à chaque point de l'image un ensemble de formes, représentées chacune par un point de l'espace de paramètres.

Dans le cas d'une forme simple qui peut totalement être caractérisée par un nombre de points fixés n petit, la connaissance d'uniquement n points permet de retourner une forme, dans un mapping qui se veut donc du n to *one*. N'importe quel n -uplet quelconque suffit à définir une forme, que celle-ci existe réellement sur l'image ou non. La détection d'une forme **présente** sur l'image ne peut donc se réduire à la détection d'un unique n -uplet la caractérisant (Figure 68).

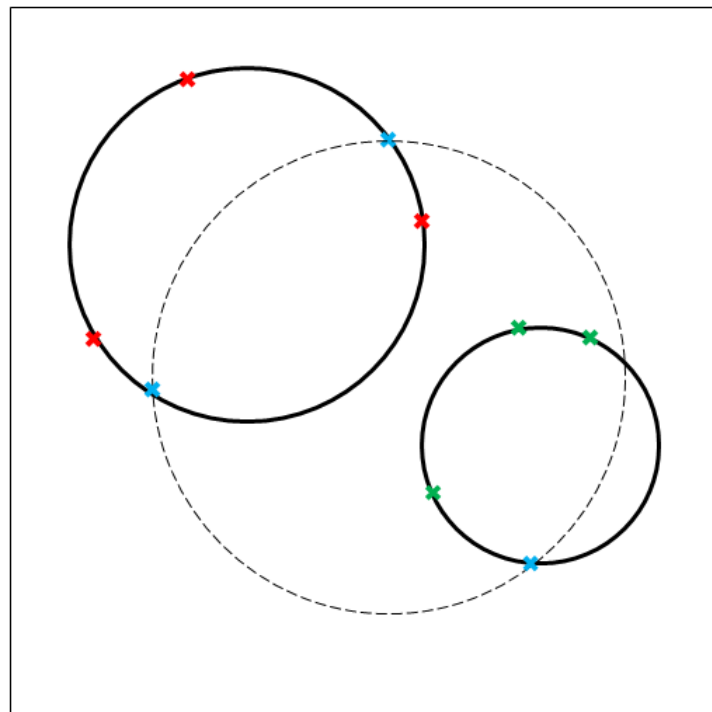


Figure 68 : Un triplet suffit à caractériser un cercle. Ici, les triplets rouge et vert caractérisent un cercle de l'image, mais le triplet bleu caractérise un cercle inexistant.

La distinction d'une forme va donc se faire sur un principe semblable à la HT standard. Un accumulateur est initialisé, dont chaque point représente donc une forme précise, et pour chaque n -uplet de l'image de base, on incrémentera donc l'accumulateur au point correspondant à la forme définie. De la même manière que pour la HT standard, les maximums supérieurs à un seuil (dépendant du bruit et de la complexité de l'image) de l'accumulateur correspondront à des formes réellement présentes.

Cette méthode présente des avantages. En effet, le mapping *n to one* renvoie un accumulateur assez net, avec un bruit de fond beaucoup plus faible que la HT standard qui a tendance à provoquer plus d'étalement de pic. De plus, pour des images avec très peu de points, cette méthode peut se révéler plus rapide que la HT standard.

Dans le cadre de la détection de cercle, la HT standard présente une complexité en $O(L^2R^2)$. Dans le cas de la méthode proposée ci-dessus, on choisit toujours un point sur l'image, mais cette fois, on en choisit deux autres dans le disque de rayon R qui s'étend autour. Ainsi, on a de nouveau une complexité en $O(L^2R^2)$. On a tout de même un temps de calcul meilleur dans tous les cas.

Cependant, les avantages de la méthode (netteté bien plus importante de l'accumulateur) permettent à celle-ci de fonctionner même sur un échantillon très limité de points. On peut alors, via des sélections aléatoires de points, rentrer dans le cadre défini ci-dessus pour lequel le mapping *n to one* est plus intéressant que la HT standard. Cette méthode, proposée par Xu, Oja et Kultanen, est donc la Transformée de Hough Aléatoire (RHT) (Xu & Oja, 1990) (Xu & Oja, 1993).

Au vu de la complexité des images à traiter, on utilise la Transformée de Hough Probabilisée. On se donne donc une image 2D binaire X définie sur $D = \llbracket 0 ; L - 1 \rrbracket^2$, et on recherche donc des structures circulaires à l'intérieur. Pour cela, le protocole est donc :

1. Création de l'accumulateur A défini comme une image nulle sur l'ensemble $E = D \times \llbracket 0 ; R \rrbracket$.
2. Calcul de la fonction de seuil $n(r)$ pour chaque $r \in \llbracket 0 ; R \rrbracket$.
3. Tant que $X^{-1}(1) \neq \emptyset$:
 - a. Tant qu'il n'existe pas de $s = (p, \rho) \in E$ tel que $A(s) > n(\rho)$:
 - i. On prend aléatoirement $x \in X^{-1}(1)$.
 - ii. Pour tout $s = (p, \rho) \in E$ tel que $\|x - p\| = \rho$, $A(s)$ est incrémenté.
 - iii. On pose $X(x) = 0$.
 - b. S'il existe un $s = (p, \rho) \in E$ tel que $A(s) > n(\rho)$:
 - i. Le cercle de centre p et de rayon ρ est détecté comme appartenant à l'image.

- ii. Pour tous les points $x \in X^{-1}(1)$ vérifiant $\|x - p\| = \rho$, on effectue $X(x) = 0$.
- iii. Pour tous les points $t = (q, \lambda) \in E$ tels que $\|q - p\| < \lambda + \rho$, on effectue $A(t) = 0$.

Finalement, on applique cet algorithme sur tous les plans qui le nécessitent, et on ajoute les structures détectées à l'image originale.

3.4.2. Détection de sphères

L'algorithme de détection de sphères à proprement parler est fondé sur le calcul des cartes de distance. Celui-ci est étudié en détail dans la partie 2.4.2. On suppose maintenant que l'on connaît la carte de distance Y (Figure 69) et l'image binaire enrichie X' —que l'on notera X par abus de langage—définies sur E . Le principe de fonctionnement de l'algorithme est alors le suivant :

1. Déterminer les maximums de la carte de distance.
2. On se donne un seuil limite τ^{min} . Pour chaque maximum, rechercher le rayon et le point proches qui correspondent aux paramètres de la sphère la plus plausible (celle dont le taux d'acceptation τ correspondant au ratio de points de la sphère appartenant à $X^{-1}(1)$ est le plus élevé). Si le taux d'acceptation de cette sphère vérifie $\tau > \tau^{min}$, la sphère est définie comme probable. Sinon, elle n'est pas considérée.
3. Tracé des cercles : On trace la sphère la plus plausible, et on supprime des possibilités toutes les sphères moins probables intersectant suffisamment la sphère tracée. On effectue ceci jusqu'à ce qu'il n'y ait plus de sphères possibles.

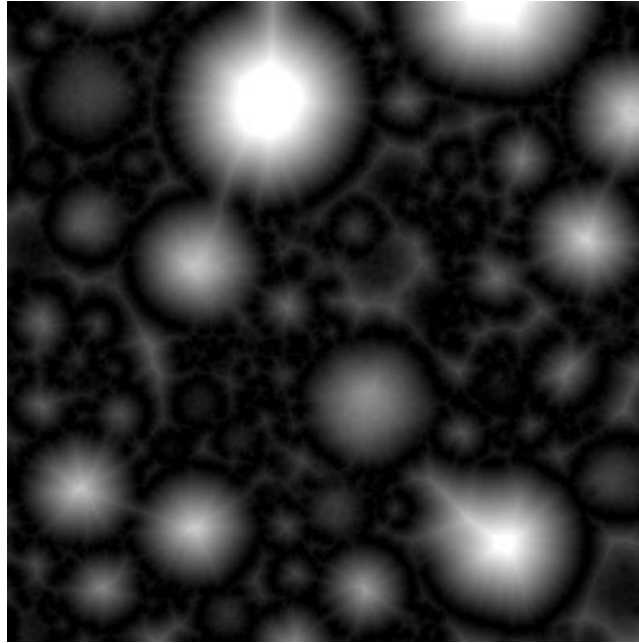


Figure 69 : Coupe de la carte de distance obtenue sur les images de mousse de silice, taille 200^2 pixels.

Deux points importants sont à noter :

- On constate que l'on ne considère pas directement le maximum de la carte de distance comme un centre incontestable de sphère. Afin de corriger les imperfections dues aux déformations des sphères, on se donne en effet des écarts possibles. On obtient alors un intervalle de recherche que l'on balaye ensuite pour retrouver la meilleure sphère. Cependant, le temps de calcul augmente très rapidement avec les écarts. On cherche donc à être le plus précis possible.
- Les sphères ne sont pas tracées directement après leur détection, mais dans une boucle à part, qui permet d'éliminer d'éventuelles sphères s'intersectant trop. L'intersection entre structures est en effet interdite sur l'image, mais les déformations peuvent entraîner sur les résultats des sphères s'intersectant très légèrement.

Étant donné les images Y et X , on effectue les étapes de l'algorithme suivantes.

On commence par déterminer l'ensemble $M \subset E$ des points réalisant les maximums locaux de Y . On se donne alors des écarts en position δ et en rayon ρ . Soit la fonction $\tau : E \times \llbracket 0 ; R \rrbracket \rightarrow [0 ; 1]$ quantifiant la présence d'une sphère dans l'image définie par :

$$\tau(p, r) = \frac{\text{card } \mathcal{S}(p, r) \cap X^{-1}(1)}{\text{card } \mathcal{S}(p, r)}$$

où $\mathcal{S}(p, r)$ représente la sphère de centre p et de rayon r .

Pour chaque point $m \in M$, on note $\Delta_m \subset E \times \llbracket 0 ; R \rrbracket$ le sous-ensemble

$$\Delta_m = \{(\mu, \nu) \in E \times \llbracket 0 ; R \rrbracket \text{ tel que } \|\mu - m\| < \delta \text{ et } Y(m) \leq \nu < Y(m) + \rho\}$$

On notera alors $(\mu_m, \nu_m) \in E \times \llbracket 0 ; R \rrbracket$ le doublet vérifiant :

$$(\mu_m, \nu_m) = \arg \max_{(\mu, \nu) \in \Delta_m} \tau(\mu, \nu)$$

Ce doublet correspond à l'ensemble centre-rayon proche de $(m, Y(m))$ représentant la sphère la plus probable. À partir de l'ensemble M , on construit un ensemble $S(M, \tau^{min}) \subset E \times \llbracket 0 ; R \rrbracket \times [\tau^{min} ; 1]$ défini par :

$$S(M) = \{(\mu_m, \nu_m, \tau(\mu_m, \nu_m)) \text{ tel que } \tau(\mu_m, \nu_m) \geq \tau^{min}, \text{ pour tout } m \in M\}$$

Ici, on a dans $S(M)$ l'ensemble des doublets centre-rayon caractérisant les sphères possibles de X , auxquels sont ajoutés les coefficients de quantification τ . On trie maintenant les éléments de $S(M)$ par coefficients τ décroissants. $S(M)$ étant un ensemble fini, on notera s_i son i -ème élément après le tri. L'ensemble $F(M)$ des sphères finalement considéré est alors construit de la manière suivante :

On commence par poser $j = 0$ et $F^0(M) = \emptyset$. Tant que $s_j = (p, r, \tau)$ existe, on effectue les étapes :

1. On détermine s'il existe $(\mu, \nu) \in F^{j-1}(M)$ tel que $\|p - \mu\| < \nu + r - \varepsilon$.
2. Si oui, alors $F^j(M) = F^{j-1}(M)$.
3. Sinon, alors $F^j(M) = F^{j-1}(M) \cup \{(p, r)\}$.
4. $j = j + 1$.

Finalement, on a déterminé un ensemble $F(M)$ de doublets caractérisant les sphères de l'image. Comme on l'a vu précédemment, les résultats ne sont pas justes sur des images denses de sphères incomplètes et déformées, des faux positifs étant ajoutés par la détection. Il est donc nécessaire de s'intéresser encore aux traitements possibles pour améliorer les résultats.

3.4.3. Traitement : Classification des maximums de la carte de distance

Parmi les traitements possibles permettant de corriger en partie le problème de détection des faux-positifs, on peut s'intéresser à la carte de distance elle-même. Celle-ci véhicule des informations non seulement par ses maximums locaux, mais aussi par la forme des surfaces de niveau autour de tels maximums.

Dans le cas d'une sphère, les surfaces de niveau de la fonction de distance dessinent de la même manière des sphères concentriques alors que dans le cas de faux-positifs, les surfaces de niveau représentent des formes présentant des points de rebroussement issus des contacts entre les vraies structures de l'image (Figure 70).

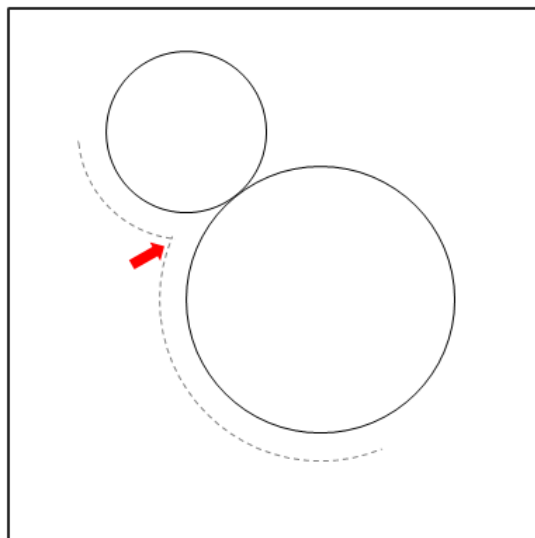


Figure 70 : Exemple d'une propagation de rebroussement. Ici, la ligne de niveau 0 (noir) présente des points de rebroussement au contact entre les cercles, qui se propagent ensuite sur les lignes de niveau proches (ligne grise et flèche rouge).

La différenciation entre les maximums associés à une sphère réelle et ceux engendrant des faux positifs peut donc se faire en observant les surfaces de niveau. On revient à un nouveau problème de détection de sphères, le principe du traitement

consistant alors à vérifier si les surfaces de niveau dessinent des sphères. On pourra tout de même, en passant à l'étude des surfaces de niveau, corriger certains problèmes rencontrés sur les images :

- Si \mathcal{S} est une structure sphérique incomplète telle qu'aucun de ses hémisphères ne soit vide, alors les surfaces de niveau proches des maximums locaux sont fermées. Ainsi, l'incomplétude problématique dans la détection directe des structures ne se pose plus ici (Figure 71), et il est possible d'être moins tolérant sur les sphères issues des surfaces de niveau que sur celles directement issues de l'image.

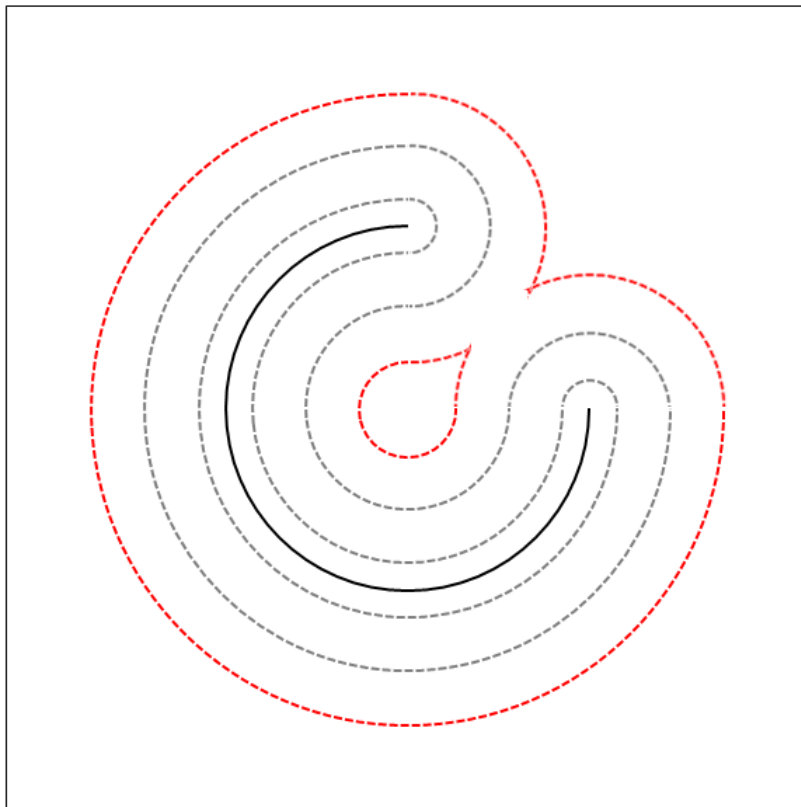


Figure 71 : Exemple de structure incomplète (en noir) et de ses lignes de niveaux (en pointillé). On constate qu'à partir d'une certaine valeur, les lignes de niveaux se ferment autour du centre de la structure (en rouge)

- Il est possible de choisir la valeur à partir de laquelle on construit les surfaces de niveau considérées. En particulier, on peut se fixer une valeur ρ donnée, et, pour chaque maximum de l'image m , considérer la surface de niveau associée à la valeur $m - \rho$. Ainsi, la valeur de la surface de niveau est choisie pour chaque maximum de manière à ce que la structure recherchée soit une sphère de rayon

ρ connue et fixé. On peut donc se restreindre à la recherche de sphères pour un unique rayon donné.

- Les surfaces de niveau proches des maximums ont tendance à régulariser les défauts dus aux incomplétudes et déformation (Figure 72). Pour une structure non sphérique, les surfaces de niveau vont tendre vers une sphère quand on s'approche d'un maximum local de la fonction de distance. Dans un ensemble discret, il est possible, si la structure n'est pas trop éloignée d'une sphère, que les surfaces de niveau les plus proches aient une discrétisation équivalente à celle d'une sphère de faible rayon, ce qui permet alors d'assimiler la structure originale à une sphère parfaite.

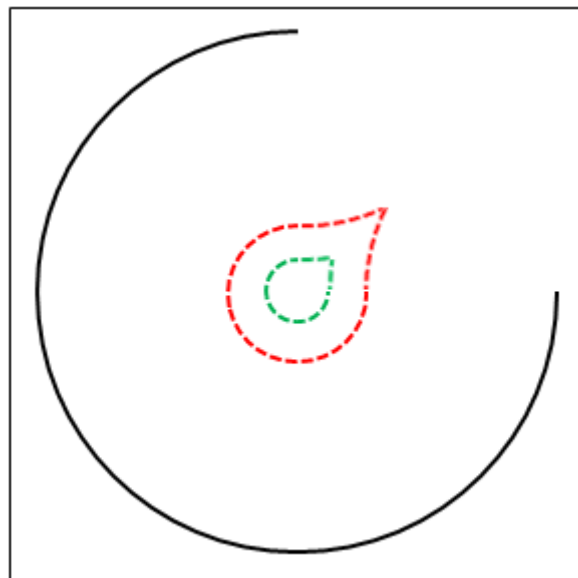


Figure 72 : Exemple de structure incomplète et de « régularisation » des lignes de niveau. Plus la ligne de niveau est proche du maximum de la fonction de distance, plus elle tend vers un cercle, comme on peut le voir avec les lignes de niveaux rouge et verte.

Cependant, il est à noter que si les surfaces de niveau « régularisent » les sphères incomplètes ou déformées, elles font de même sur les structures engendrant des faux positifs (Figure 73). L'enjeu du traitement des maximums de la carte de distance est donc de séparer les structures suivant la similarité entre les surfaces de niveau et des sphères.

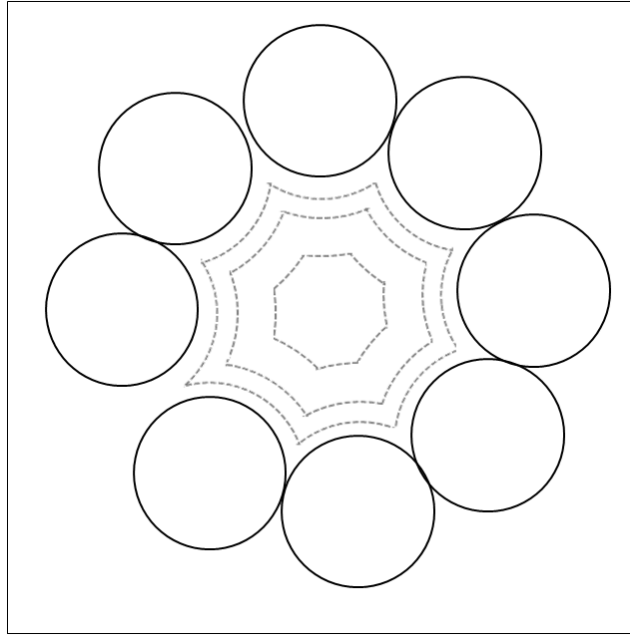


Figure 73 : Exemple de lignes de niveau dans le cas d'un ensemble de cercles pouvant entraîner la détection d'un faux positif au milieu. On constate que plus on se rapproche du centre, plus la ligne de niveau tend vers un cercle.

On est alors confronté à un problème de détection de structures sphériques plus ou moins déformées, mais complètes et de rayon constant et faible, et pour lesquelles on connaît déjà les centres possibles, qui sont les maximums locaux de la carte de distance. Pour vérifier un maximum local situé en x et de valeur m , on calculera donc, en notant \mathcal{L} l'ensemble des points de la surface de niveau associée à la valeur $m - \rho$ (avec ρ une valeur choisie à l'avance, plus petite que le rayon minimale des sphères de l'image), le paramètre :

$$\sigma(x, m) = \sum_{l \in \mathcal{L}} d(l, \mathcal{S}(x, \rho))^2$$

où $\mathcal{S}(x, \rho)$ est la sphère de centre x et de rayon ρ et d la fonction de distance euclidienne. On se fixera ensuite un seuil σ_0 , et un maximum de la carte de distance ne sera alors considéré comme un centre potentiel uniquement si

$$\sigma(x, m) < \sigma_0$$

Il est nécessaire de choisir σ_0 de manière à ne pas éliminer de centres correspondant à des vraies sphères. Le choix de ρ influe fortement celui de σ_0 . Dans les faits, les difficultés de détection ont lieu sur les grandes sphères, et les faux positifs sont plutôt des petites structures. Ainsi, le choix constant de ρ fait que dans le cas général, les irrégularités ont « eu

le temps » d'être lissées sur les structures de grande taille et moins sur celles de petite taille. La méthode est donc utilisable dans un contexte réel.

Concernant la complexité, celle-ci est négligeable, étant donné que la carte de distance est déjà connue. Il suffit juste de déduire les lignes de niveaux et calculer ensuite le paramètre σ . Ceci se fait donc en même complexité asymptotique que la détection de sphères, et s'intègre parfaitement à l'algorithme.

3.4.4. Post-Traitement : Vérification des sphères de manière locale

Rappelons rapidement la propriété principale des structures entraînant la détection de fausses sphères, énoncée dans la partie 3.3 : Les fausses sphères sont toujours issues d'un ensemble de vraies structures qui leur sont extérieures et tangentes.

On peut alors affirmer que si un point $a \in E$ appartient à une fausse sphère \mathcal{S} détectée et en même temps à l'ensemble $X^{-1}(1)$, alors ce point appartient à une structure sphérique (et on notera \mathcal{Q} la sphère la représentant) extérieure à la fausse sphère et tangente en a ou dans le voisinage de ce point, en tenant compte des erreurs de discrétisation (Figure 74).

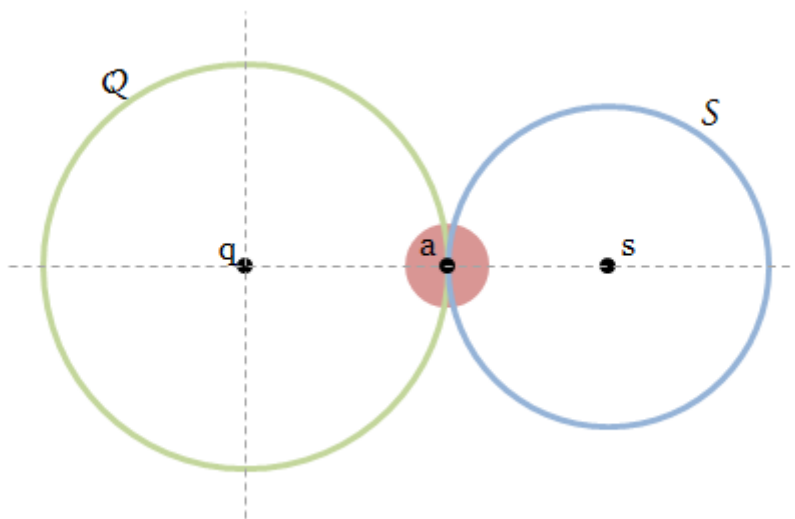


Figure 74 : Schéma explicatif de la position des structures dans le cas d'une détection de fausse sphère. a appartient à la fausse sphère \mathcal{S} et à la vraie structure \mathcal{Q} . Ces deux sphères sont tangentes en a et chacune d'un côté de leur tangente commune.

A l'inverse, si on prend un point $a \in E$ appartenant à une vraie sphère détectée et à $\mathbf{X}^{-1}(1)$, alors la structure originale et la sphère détectée sont du même côté de la tangente (et même confondues dans un cas parfait sans déformations).

Un paramètre pertinent est par conséquent le suivi de la proximité entre $\mathbf{X}^{-1}(1)$ et la sphère détectée au fur et à mesure que l'on s'éloigne de a . En effet, dans le cas d'une fausse sphère, les points de $\mathbf{X}^{-1}(1)$ forment une structure sphérique de l'autre côté de la tangente. Ainsi, si l'on s'éloigne du plan tangent, la distance entre la sphère détectée et les points de $\mathbf{X}^{-1}(1)$ va augmenter, car les deux sphères s'éloignent. Dans le cadre d'une vraie sphère, $\mathbf{X}^{-1}(1)$ est localement inclus dans le résultat, et par conséquent, la distance est constamment nulle. On va donc procéder de la manière suivante :

Définition : Soit \mathcal{E} l'ensemble des sphères retrouvées par la méthode de détection. Pour toute sphère $\mathcal{S} \in \mathcal{E}$, on note $U(\mathcal{S}, \mathbf{X}^{-1}(1)) = \mathbf{X}^{-1}(1) \cap \mathcal{S}$. Pour tout $s \in U(\mathcal{S}, \mathbf{X}^{-1}(1))$ et tout $r > 0$, on définit l'ensemble $D(s, r) = \mathbf{X}^{-1}(1) \cap \mathcal{B}(s, r)$ avec $\mathcal{B}(s, r)$ la boule de centre s et de rayon r . Finalement, on définit $D_c(s, r)$ comme la composante connexe de $D(s, r)$ contenant s .

Définition : Soit une sphère $\mathcal{S} \in \mathcal{E}$ et $s \in U(\mathcal{S}, \mathbf{X}^{-1}(1))$. On définit la fonction $f_s : \mathbb{N} \rightarrow \mathbb{R}_+$ par :

$$f_s(n) = \frac{\sum_{t \in D_c(s, n)} d(t, \mathcal{S})}{\text{card } D_c(s, n)}$$

où d représente la fonction de distance euclidienne.

L'évolution de la fonction en n nous permet ensuite d'associer à chaque point une qualification « point d'une bonne sphère/point d'une mauvaise sphère », et ensuite, par comparaison, pour chaque sphère, du ratio entre les bons et les mauvais points, on peut évaluer si la sphère est une vraie ou un faux positif.

Pour pouvoir déterminer si un point est bon ou mauvais, il est d'abord nécessaire d'étudier le comportement théorique de f_s . On commence par étudier la fonction dans un cas continu, et on extrapole ensuite les résultats au cas discret :

Soit deux sphères \mathcal{Q}, \mathcal{S} définies sur \mathbb{R}^3 , de rayon respectifs u, v . \mathcal{Q} est la sphère réelle de l'image, \mathcal{S} est une sphère détectée par la méthode grâce à \mathcal{Q} . Soit $a \in \mathcal{Q} \cap \mathcal{S}$. On définit la fonction $g_a : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ par :

$$g_a(x) = \frac{\iint_{\mathcal{Q} \cap \mathcal{B}(a,x)} d(p, \mathcal{S}) dp}{\iint_{\mathcal{Q} \cap \mathcal{B}(a,x)} dp}$$

On a alors deux cas distincts à considérer :

- \mathcal{S} correspond bien à une sphère réelle, et comme elle est détectée grâce à \mathcal{Q} , on a $\mathcal{Q} = \mathcal{S}$, d'où $\iint_{\mathcal{Q} \cap \mathcal{B}(a,x)} d(p, \mathcal{S}) dp = \iint_{\mathcal{Q} \cap \mathcal{B}(a,x)} 0 dp$ donc g_a est identiquement nulle.
- \mathcal{S} est un faux positif, auquel cas \mathcal{Q}, \mathcal{S} sont deux sphères tangentes en a . On va alors calculer $g_a(x)$.

On se place dans un repère orthonormé (O, Ox, Oy, Oz) centré sur \mathcal{Q} et tel que l'axe reliant les deux centres s, q des sphères soit l'axe (Oz) . q, s et a ont donc comme coordonnées respectives $(0,0,0), (0,0, u+v)$ et $(0,0, u)$ (voir Figure 74). On utilisera par la suite les coordonnées sphériques (R, θ, ϕ) pour repérer un point de coordonnées cartésiennes $(R \cos \phi \sin \theta, R \sin \phi \sin \theta, R \cos \theta)$.

On va d'abord déterminer, pour tout $x \geq 0$, l'ensemble des points $p(u, \theta, \phi)$ de \mathcal{Q} à distance inférieure à x de a :

$$\begin{aligned} d(p, s) &= \sqrt{(u \cos \phi \sin \theta)^2 + (u \sin \phi \sin \theta)^2 + u^2(1 - \cos \theta)^2} \\ &= u \sqrt{\sin^2 \theta + 1 - 2 \cos \theta + \cos^2 \theta} \\ &= u \sqrt{2(1 - \cos \theta)} \end{aligned}$$

et on veut que $d((u, \theta, \phi), (u, 0, 0)) \leq x$, donc

$$\begin{aligned} u \sqrt{2(1 - \cos \theta)} &\leq x \\ \Leftrightarrow 2u^2(1 - \cos \theta) &\leq x^2 \\ \Leftrightarrow \cos \theta &\geq 1 - \frac{x^2}{2u^2} \end{aligned}$$

donc

$$\begin{aligned}
\iint_{Q \cap \mathcal{B}(a,x)} dp &= \int_0^{2\pi} \int_0^{\arccos\left(1-\frac{x^2}{2u^2}\right)} u^2 \sin \theta \, d\theta d\phi \\
&= 2\pi u^2 [-\cos \theta]_0^{\arccos\left(1-\frac{x^2}{2u^2}\right)} \\
&= 2\pi u^2 \left(1 - \left(1 - \frac{x^2}{2u^2}\right)\right)
\end{aligned}$$

d'où finalement le résultat :

$$\iint_{Q \cap \mathcal{B}(a,x)} dp = \pi x^2.$$

On va maintenant calculer le deuxième terme $\iint_{Q \cap \mathcal{B}(a,x)} d(p, \mathcal{S}) \, dp$. On peut écrire que

$$d(p, \mathcal{S}) = d(p, s) - v.$$

Or, on a

$$\begin{aligned}
d(p, s) &= \sqrt{u^2 \sin^2 \theta + (u \cos \theta - u - v)^2} \\
&= \sqrt{2u^2 - 2u(u+v) \cos \theta + 2uv + v^2}
\end{aligned}$$

et par conséquent,

$$\begin{aligned}
\iint_{Q \cap \mathcal{B}(a,x)} d(p, \mathcal{S}) \, dp &= \int_0^{2\pi} \int_0^{\arccos\left(1-\frac{x^2}{2u^2}\right)} \left(\sqrt{2u^2 - 2u(u+v) \cos \theta + 2uv + v^2} - v\right) u^2 \sin \theta \, d\theta d\phi
\end{aligned}$$

ce qui donne, via un changement de variable $t = \cos \theta$,

$$\begin{aligned}
\iint_{Q \cap \mathcal{B}(a,x)} d(p, \mathcal{S}) \, dp &= 2\pi \int_{1-\frac{x^2}{2u^2}}^1 \left(\sqrt{2u^2 - 2u(u+v)t + 2uv + v^2} - v\right) u^2 dt \\
&= 2\pi u^2 \left(\int_{1-\frac{x^2}{2u^2}}^1 \sqrt{2u^2 - 2u(u+v)t + 2uv + v^2} \, dt \right) - \pi x^2 v
\end{aligned}$$

En notant $\kappa = \frac{2u(u+v)}{2u^2+2uv+v^2}$ on a :

$$\begin{aligned}
\iint_{\mathcal{Q} \cap \mathcal{B}(a,x)} d(p, \mathcal{S}) dp &= 2\pi u^2 \sqrt{2u^2 + 2uv + v^2} \int_{1-\frac{x^2}{2u^2}}^1 \sqrt{1-\kappa t} dt - \pi x^2 v \\
&= 2\pi u^2 \sqrt{2u^2 + 2uv + v^2} \left[-\frac{2}{3\kappa} (1-\kappa t)^{\frac{3}{2}} \right]_{1-\frac{x^2}{2u^2}}^1 - \pi x^2 v \\
&= \frac{4\pi u^2}{3\kappa} \sqrt{2u^2 + 2uv + v^2} \left(\left(1 - \kappa \left(1 - \frac{x^2}{2u^2} \right) \right)^{\frac{3}{2}} - (1-\kappa)^{\frac{3}{2}} \right) - \pi x^2 v
\end{aligned}$$

ce qui nous donne

$$\begin{aligned}
\iint_{\mathcal{Q} \cap \mathcal{B}(a,x)} d(p, \mathcal{S}) dp &= \frac{4\pi u^2}{6u(u+v)} (2u^2 + 2uv + v^2)^{\frac{3}{2}} \left(\left(1 - \kappa + \kappa \frac{x^2}{2u^2} \right)^{\frac{3}{2}} - (1-\kappa)^{\frac{3}{2}} \right) - \pi x^2 v \\
&= \frac{2\pi u}{3(u+v)} \left(-v^3 + \left(\frac{v^2 u + x^2(u+v)}{u} \right)^{\frac{3}{2}} \right) - \pi x^2 v \\
&= \frac{2\pi}{3\sqrt{u}(u+v)} \left(-v^3 \sqrt{u}^3 + (v^2 u + x^2(u+v))^{\frac{3}{2}} \right) - \pi x^2 v.
\end{aligned}$$

Par conséquent, on peut écrire $g_a(x)$ sous la forme

$$g_a(x) = \frac{2}{3x^2 \sqrt{u}(u+v)} \left(-v^3 \sqrt{u}^3 + (v^2 u + x^2(u+v))^{\frac{3}{2}} \right) - v.$$

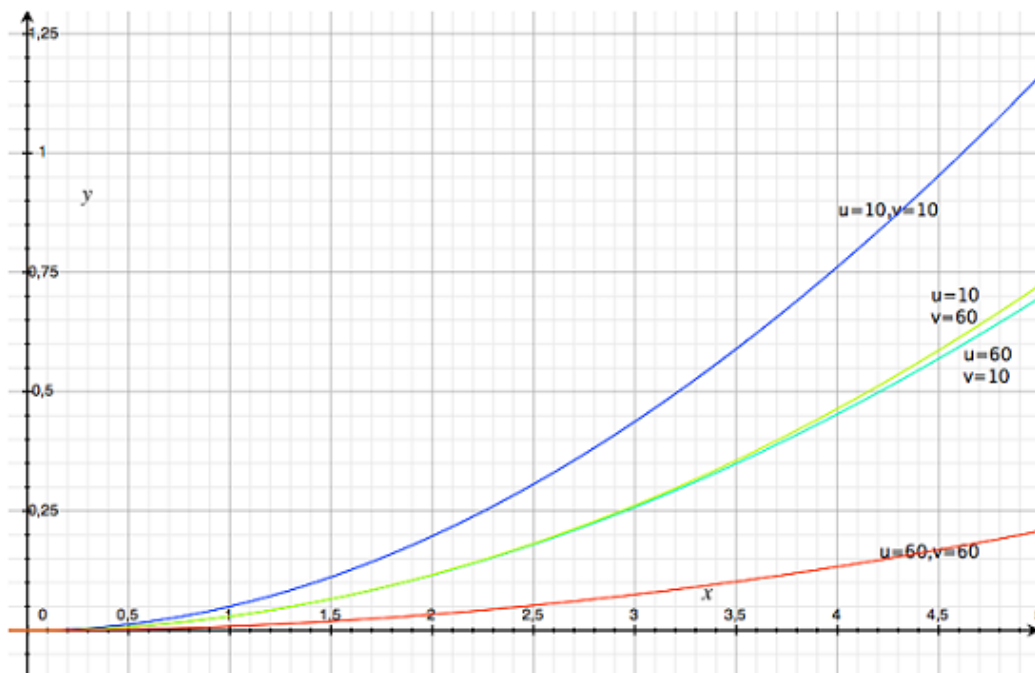


Figure 75 : Graphe de g_a selon x pour différentes valeurs de u et de v .

Maintenant que nous connaissons l'expression de la fonction et son graphe (voir Figure 75), il est possible d'étudier son comportement en 0 et en $2u$, qui correspond à la valeur maximale que peut prendre x en raison de la borne de l'intégrale en $\cos\left(1 - \frac{x^2}{2u^2}\right)$:

g_a n'est pas définie en 0. On va donc chercher à y calculer sa limite. Pour ce faire, on donne un développement limité à l'ordre 2 en 0 de la fonction $x \mapsto \left(1 + x^2 \frac{u+v}{v^2u}\right)^{\frac{3}{2}}$. En effet, on a

$$\left(1 + x^2 \frac{u+v}{v^2u}\right)^{\frac{3}{2}} = 1 + x^2 \frac{3(u+v)}{2v^2u} + o(x^2)$$

Ainsi, on peut obtenir un développement limité à l'ordre 0 de g_a en 0 :

$$g_a(x) = \frac{2v^3u}{3x^2(u+v)} \cdot x^2 \frac{3(u+v)}{2v^2u} - v + o(1) = 0 + o(1)$$

Ce qui nous donne finalement :

$$\lim_{x \rightarrow 0} g_a(x) = 0$$

Ce qui correspond à ce qui était attendu, la distance tendant vers 0 pour $x \rightarrow 0$ et l'aire considérée tendant vers 0 aussi.

La valeur de $g_a(2u)$, elle, vérifie

$$\begin{aligned} g_a(2u) &= \frac{1}{6u(u+v)} \left(-v^3 + (v^2 + 4u^2 + 4uv)^{\frac{3}{2}}\right) - v \\ &= \frac{1}{6u(u+v)} \left(-v^3 + (2u+v)^3\right) - v \\ &= \frac{1}{3(u+v)} (4u^2 + 6uv + 3v^2) - v \end{aligned}$$

ce qui donne

$$\begin{aligned} g_a(2u) &= \frac{4u^2 + 6uv + 3v^2 - 3uv - 3v^2}{3(u+v)} \\ &= \frac{u(4u + 3v)}{3(u+v)} \end{aligned}$$

On peut supposer que dans le cas discret présenté plus haut, l'évolution de la fonction la version discrète f_a sera la même que l'évolution de g_a dans le cas continu.

Ainsi, pour vérifier si une sphère est réelle, il « suffit » d'étudier le comportement de g_a en l'ensemble des points de $U(Q, X^{-1}(1))$, et de déterminer si ce comportement est plus proche de la fonction identiquement nulle ou de la fonction $x \mapsto \frac{2}{3x^2\sqrt{u}(u+v)} \left(-v^3\sqrt{u}^3 + (v^2u + x^2(u+v))^{\frac{3}{2}} \right) - v$.

Cependant, un problème apparaît. En effet, notre méthode de détection nous a rendu la sphère \mathcal{S} , ce qui nous assure de connaître v , mais la sphère Q , et donc u , sont inconnus. Il est juste possible d'encadrer u via des connaissances a priori sur les rayons des sphères de l'image, i.e. $r \leq u \leq R$. Pour être en même temps le plus précis possible pour assurer l'élimination sur toute l'échelle des rayons et pour ne pas supprimer des structures dont l'élimination serait discutable, on va donc considérer la fonction suivante :

$$h_a(x) = \min_{r \leq u \leq R} g_a(x)$$

D'un point de vue intuitif, on devine facilement que le minimum est atteint pour R le rayon maximal. En effet, plus le rayon augmente, moins les points de la sphère s'éloignent « rapidement » du plan tangent en a (voir Figure 76). Le graphe de la fonction g_a selon u semble confirmer la décroissance (Figure 77).

Ainsi, on considérera maintenant la fonction

$$h_a(x) = \frac{2}{3x^2\sqrt{R}(R+v)} \left(-v^3\sqrt{R}^3 + (v^2R + x^2(R+v))^{\frac{3}{2}} \right) - v$$

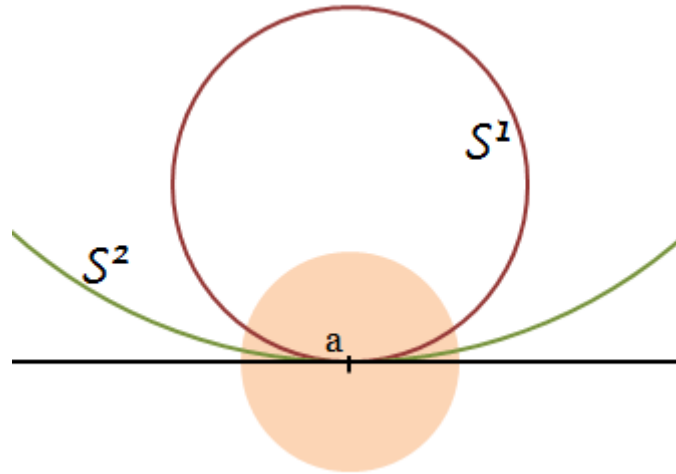


Figure 76 : Image explicative montrant la décroissance en u de g_a . En effet, plus u est grand, plus les points du cercle au voisinage de a seront proches du plan tangent au cercle en a .

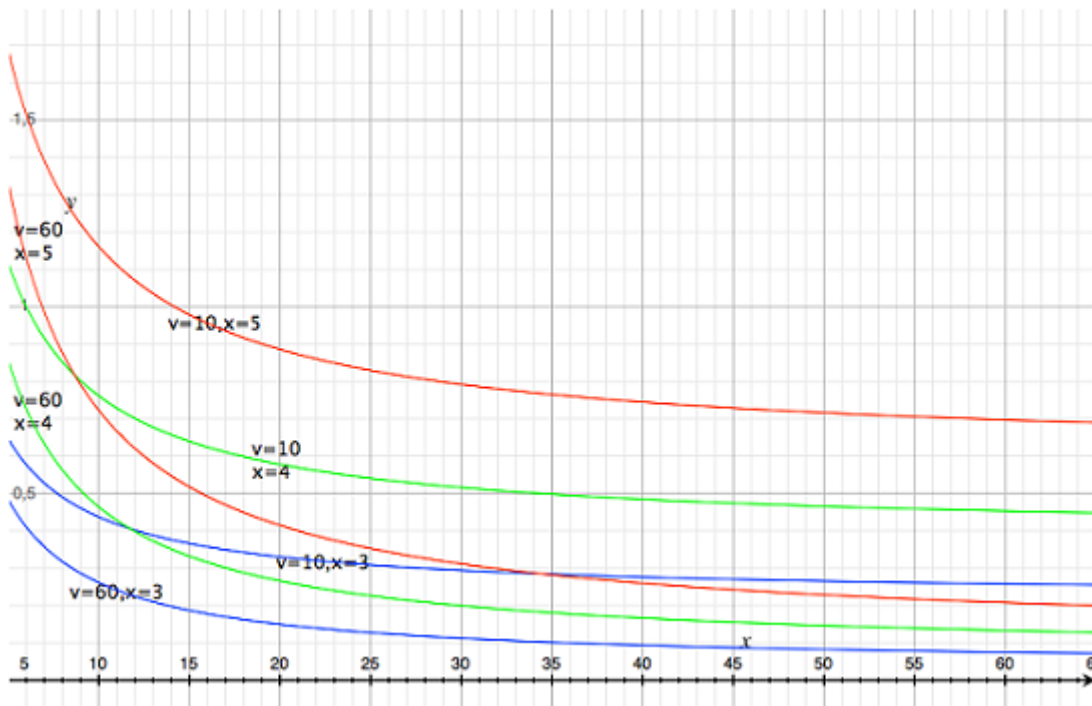


Figure 77 : Graphe de g_a selon u , pour différentes valeurs fixées de v et de x .

h_a semble être une bonne fonction pour déterminer si un cercle est réel ou faux. Cependant, il faut constater que dans le cas de nos images réelles, h_a n'est pas toujours le bon indicateur à considérer. En effet, sur ces images, nous n'avons pas de structures de sphères complètes mais uniquement de parties de sphères. Supposons avoir détecté une fausse sphère S sur la base d'un point a appartenant à une portion d'une sphère de l'image Q . Supposons que cette portion soit complète dans le voisinage de a (i.e. que tous les points de Q du voisinage appartiennent bien à $X^{-1}(1)$), mais qu'à partir d'une distance seuil d , on ait, pour tout $D > d$, que peu de points de Q appartenant encore à $X^{-1}(1)$ (c'est-à-dire que

$X^{-1}(1)$ représente \mathcal{Q} de manière incomplète par une calotte proche de a). Pour cette sphère, on constate alors que le calcul de $f_a(n)$ va être très loin de son approximation théorique g_a pour $n > d$, car la somme $\sum_{t \in D_c(a,n)} d(t, \mathcal{S})$ va se voir très peu modifiée par l'ajout de très peu de points, ce qui aura tendance à noyer l'éloignement des points à distance supérieure à d dans la masse des points plus proches, beaucoup plus nombreux. On va donc considérer une nouvelle fonction ψ_a qui sera définie comme suit :

$$\psi_a(n) = \frac{\sum_{t \in D_c(a,n)} d(t, \mathcal{S}) - \sum_{t \in D_c(a,n-1)} d(t, \mathcal{S})}{\text{card } D_c(a, n) - \text{card } D_c(a, n-1)}$$

et qui approchera une fonction définie sur $[1 ; +\infty[$ notée γ_a et définie par

$$\gamma_a(x) = \frac{\iint_{\mathcal{Q} \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S}) dp}{\iint_{\mathcal{Q} \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} dp}$$

Or, on sait que

$$\iint_{\mathcal{Q} \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} dp = \pi x^2 - \pi(x-1)^2 = \pi(2x-1)$$

et

$$\iint_{\mathcal{Q} \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S}) dp = \frac{2\pi}{3\sqrt{u}(u+v)} \left(\Gamma_{u,v}^3(x) - \Gamma_{u,v}^3(x-1) \right) - \pi v(2x-1)$$

ainsi on a :

$$\gamma_a(x) = \frac{2}{3\sqrt{u}(u+v)(2x-1)} \left(\Gamma_{u,v}^3(x) - \Gamma_{u,v}^3(x-1) \right) - v$$

avec $\Gamma_{u,v}(x) = \sqrt{v^2 u + x^2(u+v)}$.

Cependant, on peut se demander ce que l'on fait dans le cas où $\psi_a(n)$ se comporte de telle manière qu'il est difficile de décider, comme par exemple si ψ_a adopte des valeurs environ égales à la moitié des valeurs théoriques maximales prévues avec γ_a ? Dans ce cas, on ne sait pas très bien quoi faire du point en question, celui-ci pouvant très bien appartenir à une structure non prévue dans la construction de la méthode (comme si par exemple un défaut plan existait sur l'image, ce qui donnerait alors un éloignement trop faible pour être

une sphère opposée mais trop élevée pour être une partie de sphère considérée) ou à une structure de jonctions entre sphères (Figure 78). Comment déterminer dans ce cas à quel type de structure le point appartient ? On va utiliser pour cela la variance de la distance des points à la sphère, que l'on peut définir de manière idéale comme :

$$\sigma_a^2(x) = \frac{\iint_{Q \cap B(a,x) \setminus B(a,x-1)} d(p, \mathcal{S})^2 dp}{\iint_{Q \cap B(a,x) \setminus B(a,x-1)} dp} - \gamma_a^2(x)$$

et de manière discrète comme :

$$\zeta_a^2(n) = \frac{\sum_{t \in D_c(a,n) \setminus D_c(a,n-1)} d(t, \mathcal{S})^2}{\text{card}(D_c(a,n) \setminus D_c(a,n-1))} - \psi_a^2(n)$$

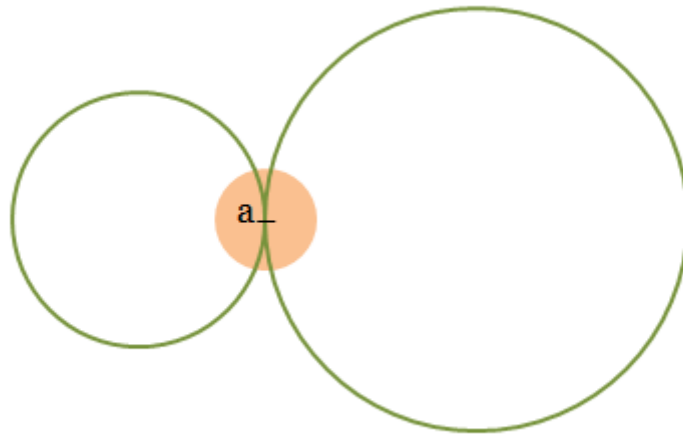


Figure 78 : Schéma d'une jonction de sphères. Si l'une des sphères est détectée, la fonction ψ_a ne suivra pas pour autant des valeurs « normales » en a pour une sphère réelle.

Calculons donc $\iint_{Q \cap B(a,x) \setminus B(a,x-1)} d(p, \mathcal{S})^2 dp$ dans le cas simple d'une « fausse » sphère tangente à une vraie sphère en a . Notons $\alpha(x) = \arccos\left(1 - \frac{x^2}{2u^2}\right)$ et $\Delta_{u,v}(x) = \Gamma_{u,v}^3(x) - \Gamma_{u,v}^3(x-1)$. On a alors :

$$\begin{aligned}
\iint_{Q \cap B(a,x) \setminus B(a,x-1)} d(p,s)^2 dp &= \int_0^{2\pi} \int_{\alpha(x-1)}^{\alpha(x)} d(p,s)^2 u^2 \sin \theta d\theta d\phi \\
&= 2\pi u^2 \int_{\alpha(x-1)}^{\alpha(x)} (d(p,s) - v)^2 \sin \theta d\theta \\
&= 2\pi u^2 \left(\begin{aligned} &\int_{\alpha(x-1)}^{\alpha(x)} d(p,s)^2 \sin \theta d\theta \\ &- 2v \int_{\alpha(x-1)}^{\alpha(x)} d(p,s) \sin \theta d\theta \\ &+ v^2 \int_{\alpha(x-1)}^{\alpha(x)} \sin \theta d\theta \end{aligned} \right)
\end{aligned}$$

On notera $I_1 = \int_{\alpha(x-1)}^{\alpha(x)} d(p,s)^2 \sin \theta d\theta$, $I_2 = 2v \int_{\alpha(x-1)}^{\alpha(x)} d(p,s) \sin \theta d\theta$ et $I_3 = v^2 \int_{\alpha(x-1)}^{\alpha(x)} \sin \theta d\theta$. On a de manière simple, via les calculs effectués précédemment que :

$$\begin{aligned}
I_2 &= \frac{v}{\pi u^2} \cdot (\gamma_a(x) + v) \cdot \iint_{Q \cap B(a,x) \setminus B(a,x-1)} dp \\
&= \frac{2v \Delta_{u,v}(x)}{3u^2(u+v)}
\end{aligned}$$

et via un changement de variable simple, on a

$$I_3 = \frac{v^2}{2u^2} (2x - 1)$$

Concernant I_1 , on a

$$\begin{aligned}
I_1 &= \int_{\alpha(x-1)}^{\alpha(x)} d(p,s)^2 \sin \theta d\theta \\
&= \int_{\alpha(x-1)}^{\alpha(x)} (2u^2 - 2u(u+v) \cos \theta + 2uv + v^2) \sin \theta d\theta \\
&= \int_{1-\frac{x^2}{2u^2}}^{1-\frac{(x-1)^2}{2u^2}} (2u^2 - 2u(u+v)t + 2uv + v^2) dt
\end{aligned}$$

ce qui nous donne :

$$\begin{aligned}
I_1 &= (2u^2 + 2uv + v^2) \left(1 - \frac{(x-1)^2}{2u^2} - 1 + \frac{x^2}{2u^2} \right) - 2u(u+v) \int_{1-\frac{x^2}{2u^2}}^{1-\frac{(x-1)^2}{2u^2}} t dt \\
&= (2u^2 + 2uv + v^2) \frac{2x-1}{2u^2} - u(u+v) [t^2]_{1-\frac{x^2}{2u^2}}^{1-\frac{(x-1)^2}{2u^2}} \\
&= (2u^2 + 2uv + v^2) \frac{2x-1}{2u^2} - u(u+v) \left[\left(1 - \frac{(x-1)^2}{2u^2} \right)^2 - \left(1 - \frac{x^2}{2u^2} \right)^2 \right]
\end{aligned}$$

ce qui donne en développant

$$\begin{aligned}
I_1 &= (2u^2 + 2uv + v^2) \frac{2x-1}{2u^2} - u(u+v) \left[1 - \frac{(x-1)^2}{u^2} + \frac{(x-1)^4}{4u^4} - 1 + \frac{x^2}{u^2} - \frac{x^4}{4u^4} \right] \\
&= (2u^2 + 2uv + v^2) \frac{2x-1}{2u^2} - u(u+v) \left(\frac{2x-1}{u^2} + \frac{-4x^3 + 6x^2 - 4x + 1}{4u^4} \right) \\
&= \frac{(2u^2 + 2uv + v^2)(2x-1) - (2u^2 + 2uv)(2x-1)}{2u^2} - (u+v) \frac{-4x^3 + 6x^2 - 4x + 1}{4u^4}
\end{aligned}$$

ce qui nous donne finalement :

$$I_1 = \frac{v^2(2x-1)}{2u^2} - (u+v) \frac{-4x^3 + 6x^2 - 4x + 1}{4u^4}$$

Donc, finalement, on a :

$$\iint_{Q \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S})^2 dp = 2\pi v^2(2x-1) - \pi \frac{u+v}{2u} (-4x^3 + 6x^2 - 4x + 1) - 2\pi u^2 I_2$$

et ainsi

$$\frac{\iint_{Q \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S})^2 dp}{\iint_{Q \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} dp} = 2v^2 + \frac{(u+v)(4x^3 - 6x^2 + 4x - 1)}{2u(2x-1)} - \frac{2u^2 I_2}{2x-1}$$

Or $4x^3 - 6x^2 + 4x - 1 = (2x-1)(2x^2 - 2x + 1)$, donc

$$\frac{\iint_{Q \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S})^2 dp}{\iint_{Q \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} dp} = 2v^2 + \frac{(u+v)}{2u} (2x^2 - 2x + 1) - \frac{2u^2 I_2}{2x-1}$$

De l'autre côté, on a :

$$\begin{aligned}
\gamma_a^2(x) &= \left(\frac{2\Delta_{u,v}(x)}{3\sqrt{u}(u+v)(2x-1)} - v \right)^2 \\
&= \frac{4\Delta_{u,v}(x)^2}{9u(u+v)^2(2x-1)^2} - \frac{4v\Delta_{u,v}(x)}{3\sqrt{u}(u+v)(2x-1)} + v^2 \\
&= \frac{4\Delta_{u,v}(x)^2}{9u(u+v)^2(2x-1)^2} - \frac{2u^2 I_2}{2x-1} + v^2
\end{aligned}$$

D'où finalement le résultat :

$$\sigma_a^2(x) = v^2 + \frac{(u+v)}{2u}(2x^2 - 2x + 1) - \frac{4\Delta_{u,v}(x)^2}{9u(u+v)^2(2x-1)^2}$$

Ainsi, on a trouvé la forme de $\sigma_a^2(x)$ dans le cas d'une fausse sphère tangente à une sphère de l'image en a .

Calculons maintenant $\sigma_a^2(x)$ dans le cas montré sur la Figure 78, c'est-à-dire le cas de deux sphères \mathcal{Q}, \mathcal{S} de l'image tangentes en a dont l'une est détectée par la méthode. On a alors

$$\iint_{(\mathcal{Q} \cup \mathcal{S}) \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} dp = 2\pi(2x-1)$$

et

$$\begin{aligned}
\iint_{(\mathcal{Q} \cup \mathcal{S}) \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S}) dp &= \iint_{\mathcal{Q} \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S}) dp + \iint_{\mathcal{S} \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S}) dp \\
&= \iint_{\mathcal{Q} \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S}) dp + \iint_{\mathcal{S} \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} 0 dp \\
&= \iint_{\mathcal{Q} \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S}) dp
\end{aligned}$$

Ainsi, on a :

$$\frac{\iint_{(\mathcal{Q} \cup \mathcal{S}) \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} d(p, \mathcal{S})^2 dp}{\iint_{(\mathcal{Q} \cup \mathcal{S}) \cap \mathcal{B}(a,x) \setminus \mathcal{B}(a,x-1)} dp} = v^2 + \frac{(u+v)}{4u}(2x^2 - 2x + 1) - \frac{u^2 I_2}{2x-1}$$

et

$$\gamma_a^2(x) = \frac{\Delta_{u,v}(x)^2}{9u(u+v)^2(2x-1)^2} - \frac{u^2 I_2}{2(2x-1)} + \frac{v^2}{4}$$

D'où finalement

$$\sigma_a^2(x) = \frac{3v^2}{4} + \frac{(u+v)}{4u}(2x^2 - 2x + 1) - \frac{\Delta_{u,v}(x)^2}{9u(u+v)^2(2x-1)^2} - \frac{v\Delta_{u,v}(x)}{3\sqrt{u}(u+v)(2x-1)}$$

La Figure 79 montre le graphe de σ_a (i.e. l'écart-type de la distance entre les points de l'image et ceux de la sphère détectée) pour différentes valeurs de u et v dans les deux cas explicités ci-dessus : le cas n°1 correspond à une fausse sphère détectée, tangente en a à une sphère de l'image, et le cas n°2 correspond pour sa part à une sphère réelle détectée tangente à une autre sphère réelle de l'image en a . On constatera l'évolution bien plus rapide de l'écart-type dans le second cas que dans le premier. On considère alors, en cas de doute sur le point en raison des valeurs de γ_a , que le point est conservé si σ_a est élevé (ce qui correspond alors soit à une situation de sphères tangentes, soit à une absence de structure claire) et qu'il est éliminé si σ_a est faible (auquel cas on a affaire à une structure où les points s'éloignent réellement de la structure détectée).

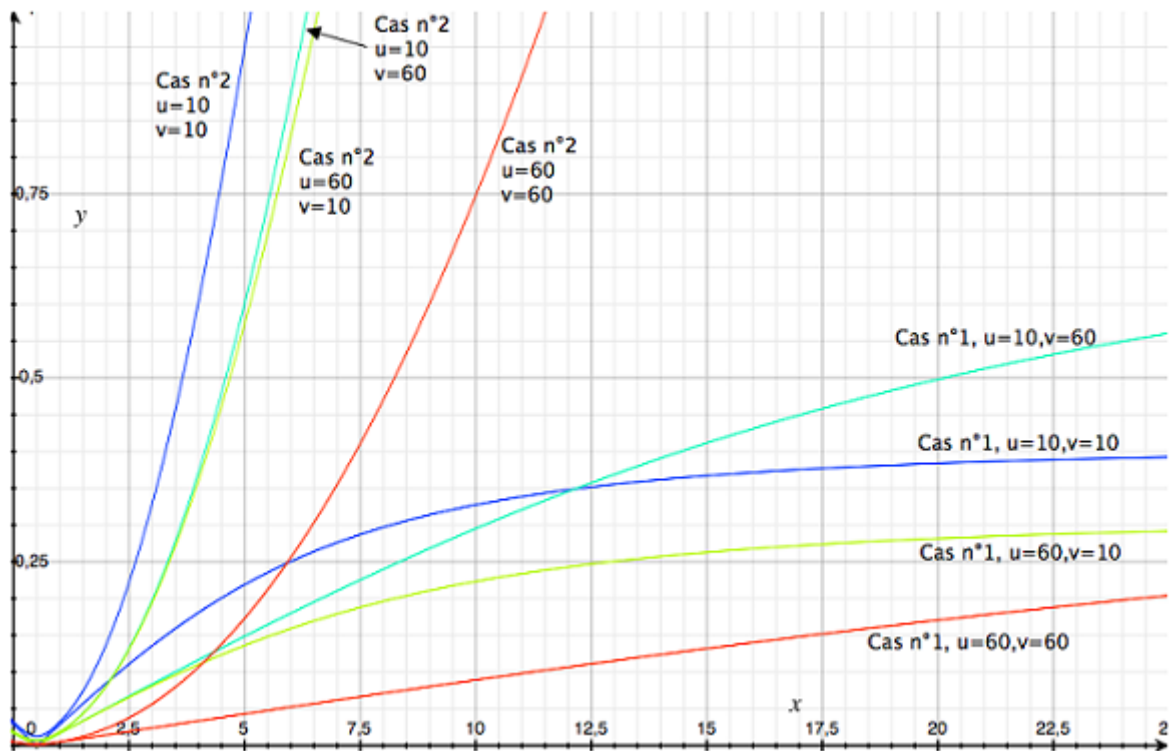


Figure 79 : Graphe de $\sigma_a(x)$ pour différentes valeurs de u, v et dans deux cas distincts : une fausse sphère détectée tangente à une sphère de l'image (cas 1) et deux sphères réelles tangentes (cas 2)

3.5. Tests et Résultats

Comme pour la méthode de détection de droites, il est nécessaire, pour valider le fonctionnement de l'algorithme de détection de sphères, de le vérifier sur une batterie d'images virtuelles préalablement créées.

Ici, on utilise quatre types d'images virtuelles. Dans les quatre cas, le rayon maximal des sphères est de 35 voxels. Les rayons minimaux et maximaux n'influent qu'assez peu sur la rapidité de la méthode géométrique de détection de sphères. Ces valeurs de rayons correspondent à peu près à ce que l'on trouve dans les images de mousses de polyuréthane. On est donc relativement proche d'une exploitation du code en contexte réel.

3.5.1. Images de sphères sans contacts

Pour commencer, on génère des images virtuelles composées de sphères parfaites sans aucun contact entre elles (Figure 80). De plus, on fera en sorte qu'aucune des sphères n'intersecte les bords de l'image, ce qui facilitera le travail de comptage et de vérification en nous libérant d'une phase de découpage des bords afin d'observer les résultats au cœur. Dans un pareil cadre, la méthode doit tout retrouver sans aucune exception. En effet, les centres des sphères doivent correspondre à des maximums de la carte de distance. Le fait que les sphères soient parfaites nous autorise à travailler avec une tolérance de 0, et comme les sphères ne se touchent pas, il est totalement exclu qu'un faux positif soit détecté.

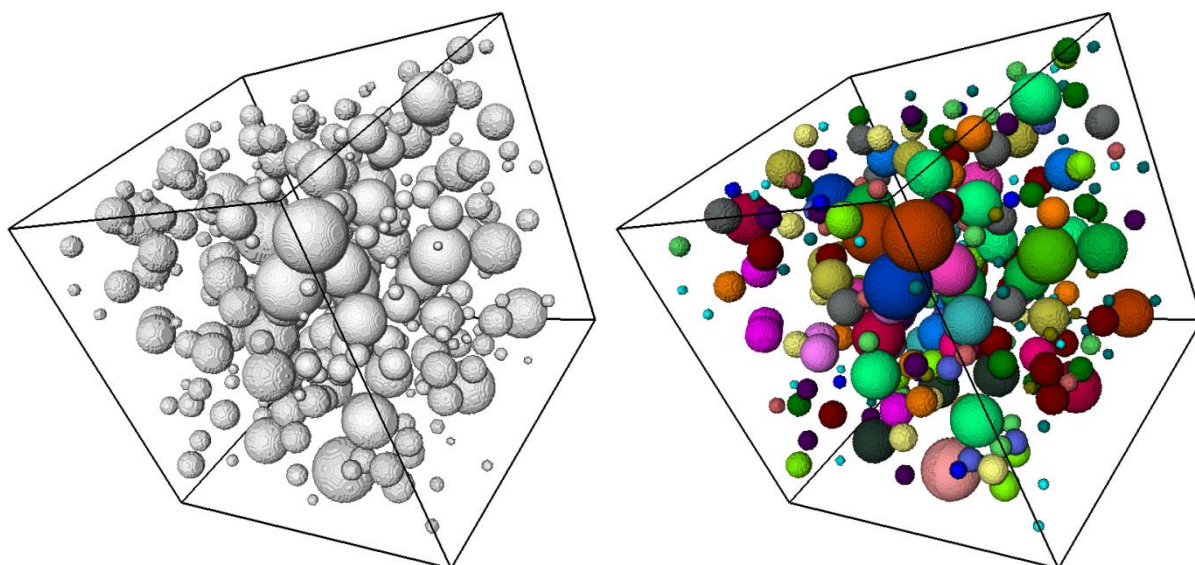


Figure 80 : Image virtuelle créée de taille 400^3 avec 300 sphères sans contact. A droite, résultat de l'algorithme de détection de sphères appliqué à l'image.

On teste ici l'algorithme sur une image notée 1.1 de taille 400^3 avec 300 sphères aléatoirement choisie (Figure 80), puis sur des images présentées dans les tableaux ci-dessous :

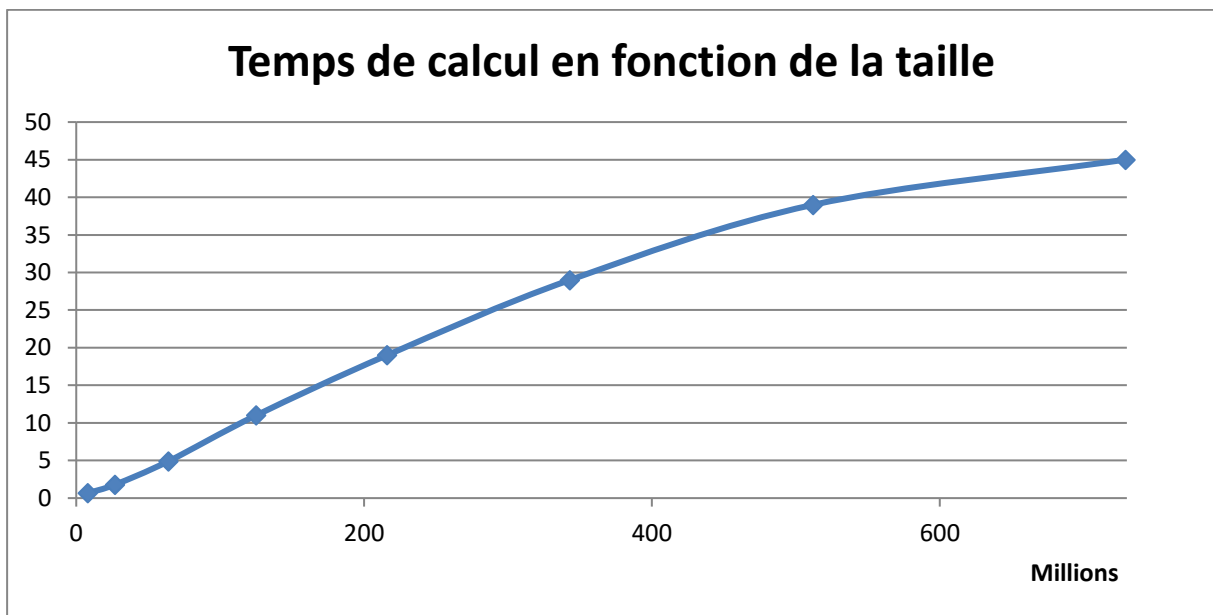
Numéro de l'image	1.2	1.3	1.4	1.5
Taille	200^3	200^3	400^3	400^3
Nombre de sphères	75	150	600	1200

Numéro de l'image	1.6	1.7	1.8	1.9
Taille	600^3	600^3	800^3	800^3
Nombre de sphères	2025	4050	4800	9600

Dans tous les cas, on obtient sur ces images un taux de détection de 100 % sur les sphères de rayon supérieur ou égal à 4 voxels (et aucune n'est détectée en dessous), ce qui montre que l'algorithme détecte correctement les sphères dans un cas simple (Figure 80). Au niveau du temps de calcul, on obtient les valeurs suivantes sur les images avec peu de sphères :

Numéro de l'image	1.2	1.4	1.6	1.8
Taille	200^3	400^3	600^3	800^3
Temps (s)	0,66	4,9	16	39

En multipliant par deux le nombre de sphères, les temps de calcul sont sensiblement équivalents pour les images de petite taille, et 50 % plus importants pour les images de grande taille.



Le comportement temporel de l'algorithme en fonction de la taille suit une courbe dont la variation se fait de moins en moins forte au fur et à mesure que la taille croît. La complexité ne se comportant pas mieux qu'en L^3 , les seules fonction polynômiales permettant d'obtenir une accélération nulle sont ceux du troisième degré. On a donc un comportement asymptotique en $O(L^3)$.

On a donc un algorithme de détection de sphères rapide et fonctionnel sur des images de sphères simples.

3.5.2. Images de sphères avec contacts et sans interpénétrations

Dans un deuxième temps, on génère des images virtuelles composées de sphères parfaites avec contacts autorisés (Figure 81). On fera toujours en sorte qu'aucune des

sphères n'intersecte les bords de l'image. La méthode doit tout retrouver à nouveau. En effet, les centres des sphères doivent correspondre à des maximums de la carte de distance. Le fait que les sphères soient parfaites nous autorise à travailler avec une tolérance de 0. Le fait que les sphères se touchent ne modifie pas le comportement de la carte de distance. Des sphères se touchant ne peuvent pas créer un emplacement où une fausse sphère serait détectée avec une tolérance de 0, car chaque point de la fausse sphère serait alors un point tangent d'une sphère réelle, et la non-intersection des sphères fait que seules des sphères de rayon inférieur au voxel permettent de tel arrangement.

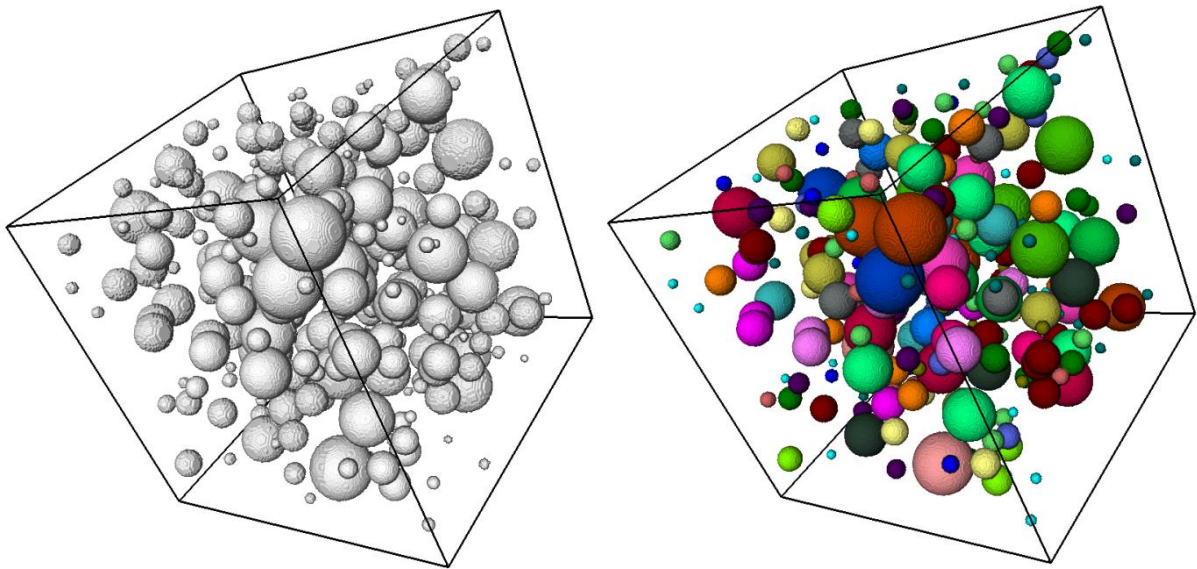


Figure 81 : Image virtuelle créée de taille 400^3 avec 300 sphères avec contacts. A droite, résultat de l'algorithme de détection de sphères appliqué à l'image.

On teste ici l'algorithme sur une image notée 2.1 de taille 400^3 avec 300 sphères aléatoirement choisie (Figure 81), ainsi que sur différentes images semblables.

On obtient à nouveau un taux de détection de 100 %, avec aucune fausse sphère détectée. Les temps de calcul sont comparables à ceux obtenus pour les images sans contacts de la partie 3.5.1.

3.5.3. Images de sphères avec interpénétrations

Ensuite, on génère des images virtuelles composées de sphères parfaites avec des interpénétrations. On fera toujours en sorte qu'aucune des sphères n'intersecte les bords de l'image. Ces interpénétrations peuvent, cette fois, modifier la structure de la carte de

distance à l'intérieur des sphères et plus difficile la détection des sphères en raison de déplacements des centres associés. De plus, les intersections facilitent la détection de fausses sphères, car les sphères créant les « faux emplacements » peuvent maintenant s'intersecter, et donc recouvrir une plus vaste partie de la fausse sphère. S'il est nécessaire d'augmenter la tolérance pour s'affranchir du déplacement des centres, il devient alors possible de voir apparaître des faux-positifs.

Ici, on choisit de limiter les interpénétrations à des rayons relativement faibles, de l'ordre de 4 voxels maximum sur les images que l'on étudie.

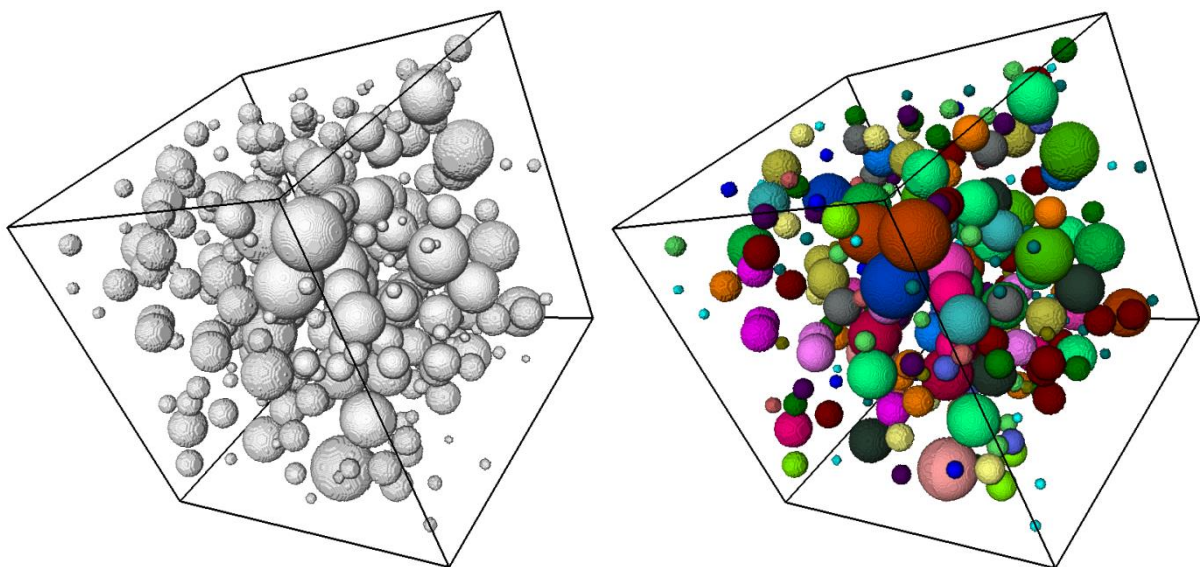


Figure 82 : Image virtuelle créée de taille 400^3 avec 300 sphères avec interpénétration maximale de 4 voxels. A droite, résultat de l'algorithme de détection de sphères appliqué à l'image.

Malgré les réserves que l'on a émises, on obtient à nouveau un taux de détection de 100 % sur les sphères de rayon supérieur à 4 voxels, avec aucune fausse sphère détectée, et aucun déplacement significatif de sphères, le tout avec une tolérance de 0. Le temps de calcul reste égal à ceux des parties précédentes.

Les interpénétrations autorisées sont ici assez faibles pour ne pas influencer sur les résultats.

3.5.4. Images de sphères incomplètes avec interpénétrations

Dans un dernier temps, on génère des images virtuelles composées de sphères incomplètes et qui peuvent s'interpénétrer, afin de s'approcher le plus possibles des

conditions réelles de travail de l'algorithme (Figure 83). Les structures présentées ici sont sous forme d'intersection de sphères par des plans, telles que la surface restante soit au moins égale à 50 % de la sphère originale, ceci afin de ne pas créer de « fuite des maximums » de la carte de distance (c'est-à-dire des cas où il n'existe pas de maximum local en raison de la présence d'un hémisphère vide). Sans les interpénétrations, on peut donc s'attendre à retrouver toutes les sphères avec une tolérance supérieure à 50 %, mais il sera impossible de tout détecter avec une tolérance inférieure (Figure 83).

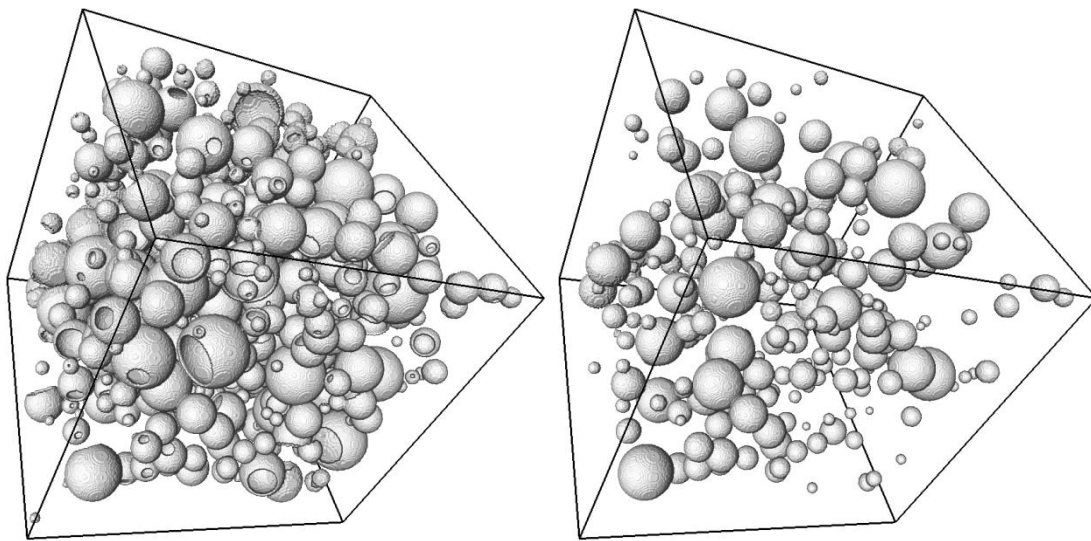


Figure 83 : À gauche, image virtuelle créée de taille 400^3 avec 600 sphères incomplètes (jusqu'à 50 % de la surface totale absente) avec interpénétration maximale de 5 voxels. À droite, résultat de la détection avec une tolérance de 10 %.

Dans les faits, les interpénétrations provoquent quelques déplacements de centres qui font que l'on ne détecte pas 100 % des sphères du premier coup avec une tolérance de 50 % (Figure 84). Si l'on augmente cette tolérance à 70 %, alors on a détecté l'ensemble des sphères de l'image de rayon supérieur à 6 (Figure 85). En dessous, les interpénétrations et les coupes, combinées à la discrétisation toujours compliquée des petites structures, rend la détection plus difficile. Une des questions importantes lorsque l'on considère une tolérance aussi élevée concerne la présence éventuelle de fausses sphères. On constate, si l'on prend l'image représentant ce qui a été ajouté par la détection, qu'uniquement des parties de calottes manquantes y sont représentées (Figure 86). Ainsi, aucun faux-positif n'a été ajouté. Pour autant, si on augmente la densité de sphères, des faux-positifs finissent nécessairement par apparaître.

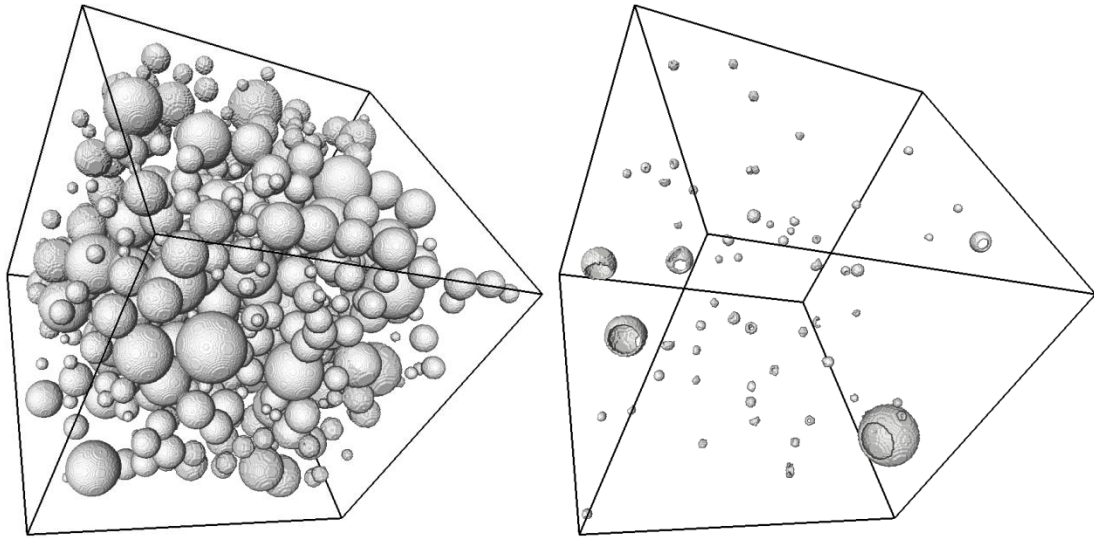


Figure 84 : À gauche, résultat de la détection de sphères sur l'image de la Figure 83 avec une tolérance de 50 %. A droite, image représentant ce qui n'a pas été détecté. Cinq structures de rayon supérieur à 6 n'ont pas été retrouvées.

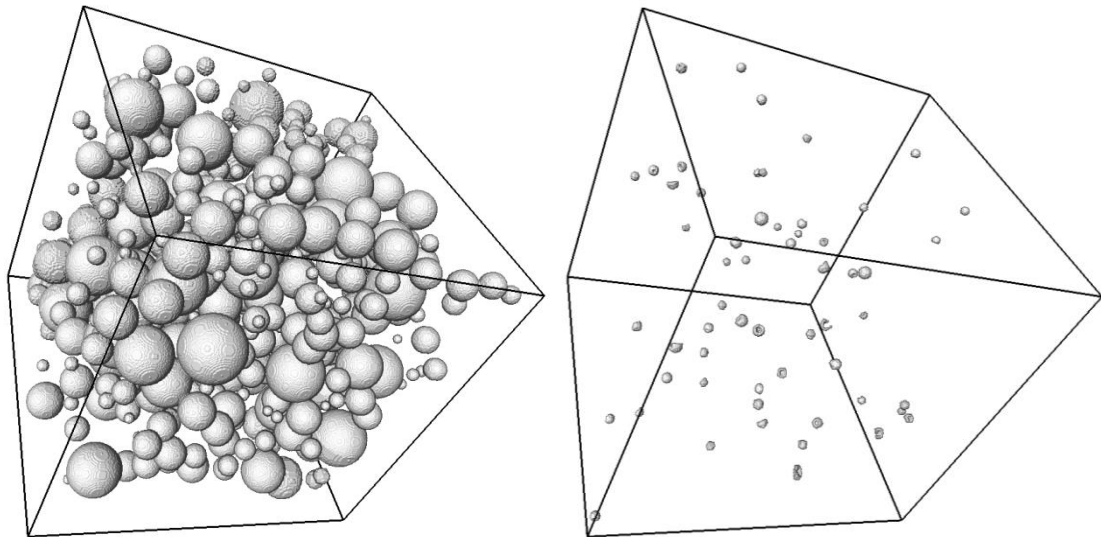


Figure 85 : À gauche, résultat de la détection de sphères sur l'image de la Figure 83 avec une tolérance de 70 %. A droite, image représentant ce qui n'a pas été détecté. Toutes les grandes sphères ont été retrouvées

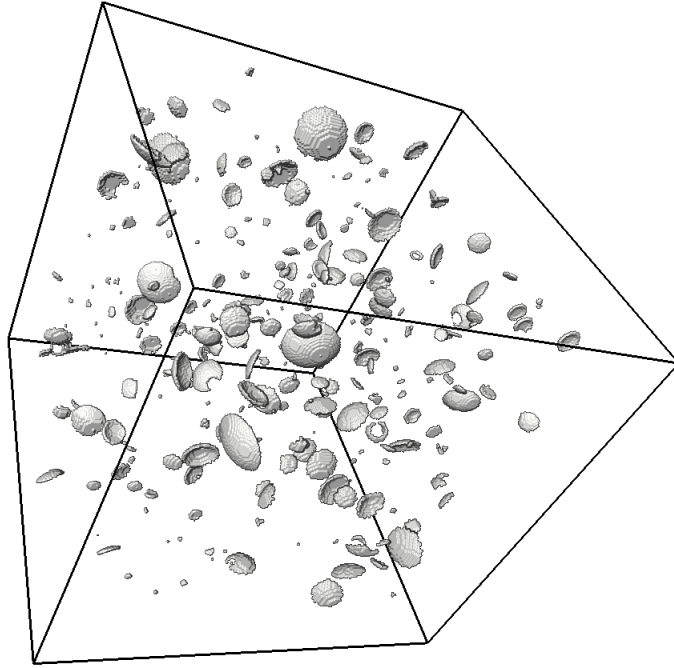


Figure 86 : Image représentant ce qui a été ajouté à l'image de la Figure 83 par la détection de la Figure 85. On remarque qu'uniquement les morceaux de calottes manquantes sont présents. Aucun faux-positif n'a donc été détecté.

4. Applications et utilisations des résultats

4.1. Matériaux composites carbone/résine

Note : Sauf mention contraire, les résultats présentés ici sont obtenus sur des images de composites avec une résolution de $0,7 \mu\text{m}/\text{vx}$.

Sur les images de matériaux carbone/résine présentées dans le chapitre 2, le but recherché consiste en la modélisation des propriétés thermiques de structures macroscopiques constituées entre autres de tels matériaux. Afin de pouvoir effectuer de telles modélisations, il est nécessaire, dans un premier temps, de retrouver sur les images la localisation et l'orientation des constituants, qui, associée à la connaissance des propriétés thermiques de chaque constituant, permet de simuler le comportement thermique du matériau. Cependant, si l'on cherche à modéliser des structures de taille importante tout en tenant compte de la structure microscopique du matériau, la quantité d'éléments de discrétisation nécessaire est énorme. La modélisation du matériau à une échelle différente permettrait d'alléger la charge de calcul en augmentant la taille de l'élément de discrétisation. Ici, les fibres étant réunies sous forme de fils, ceux-ci peuvent être considérés comme constitués d'une structure homogène aux propriétés macroscopiques équivalentes à celles des groupes de fibres détectés. La détection des fils et leur modélisation sous forme d'un matériau homogène équivalent permettent donc de réduire les besoins de discrétisation pour une modélisation thermique ultérieure.

La méthode d'homogénéisation consiste en la recherche d'un volume élémentaire représentatif (VER). Le VER est le volume minimal à partir duquel les propriétés moyennes d'un volume du matériau ne dépendent pas de la taille du volume considéré. Ainsi, le VER correspond au volume où les effets locaux dus aux constituants n'influent plus sur le comportement macroscopique. Celui-ci dépend donc fortement des propriétés et du matériau considérés. Pour les matériaux composites carbone-résine, on peut par exemple construire un VER global, à partir duquel les propriétés thermiques du matériau ne varient plus, et on peut aussi par exemple chercher un VER pour les fils, afin de représenter chaque fil de manière homogène.

Suite à la détermination du VER sur les images tomographiques de départ, il est nécessaire, pour l'homogénéisation, de déterminer les propriétés thermiques moyennes de celui-ci, ce qui est fait grâce à des procédés numériques simulant des expériences thermiques sur le matériau. Des algorithmes de résolution numérique de l'équation de la chaleur permettent d'obtenir des profils thermiques, via lesquels il est alors possible de retrouver les propriétés d'un matériau homogène équivalent.

Les algorithmes de simulation thermique ont besoin des valeurs des conductivités thermiques en chaque point de l'image, et donc au moins de la localisation de chaque constituant. Dans le cadre de constituants anisotropes, comme les fibres, présentant des résistivités axiales et radiales différentes, l'orientation des constituants—qui nous permet ensuite de connaître la position relative de chaque point—est aussi nécessaire.

C'est dans ce cadre que l'on recherche donc, en premier lieu, les localisations et orientations des constituants dans les matériaux composites carbone/résine. Suite à cela, la connaissance des fils peut nous permettre de simplifier la modélisation du matériau en cherchant un VER pour de telles structures. Il devient alors plus simple de poursuivre l'approche multi-échelle jusqu'à trouver un VER plus grand, cette fois pour le matériau entier.

On commence donc par s'intéresser aux résultats de la segmentation. Les leviers que l'on possède pour influencer sur la segmentation par champs de Markov sont le type de distribution choisie pour chaque classe, les valeurs initiales de ces distributions, et le paramètre d'importance du voisinage β . En ce qui concerne le type de distribution, celui-ci se doit d'approcher les données. Dans tous les cas présentés ici, on se limitera à des gaussiennes. Les valeurs initiales de ces distributions, qui sont ici totalement déterminées par la moyenne et la variance, influencent la segmentation initiale, avant de débiter les itérations de l'algorithme. On tente d'approcher le plus possible la forme de l'histogramme de l'image à partir des différentes distributions initiales. Le paramètre β , finalement, est le levier majeur sur lequel on agit pour obtenir une segmentation approchant le but recherché. Il est à noter qu'au départ, avec une segmentation par seuillage classique avec un seuil convenablement choisi, les centres des fibres sont souvent pleins, et celles-ci sont souvent reliées entre elles. De fait, un seuillage simple s'approche du résultat recherché pour la

méthode de détection utilisant la carte de distance. Le terme de voisinage n'a donc pas besoin d'être très fort. De fait, on prendra un β faible si à la suite, on recherche avec la carte de distance, et un β fort pour la méthode des régions connexes.

Différents résultats sur une image test sont présentées sur la Figure 87 et la Figure 88 : à partir de l'image de départ, on obtient différentes segmentations selon les valeurs de β , qui peuvent être comparées à des méthodes de segmentation plus classiques comme un seuillage associé à un filtre médian (Figure 88).

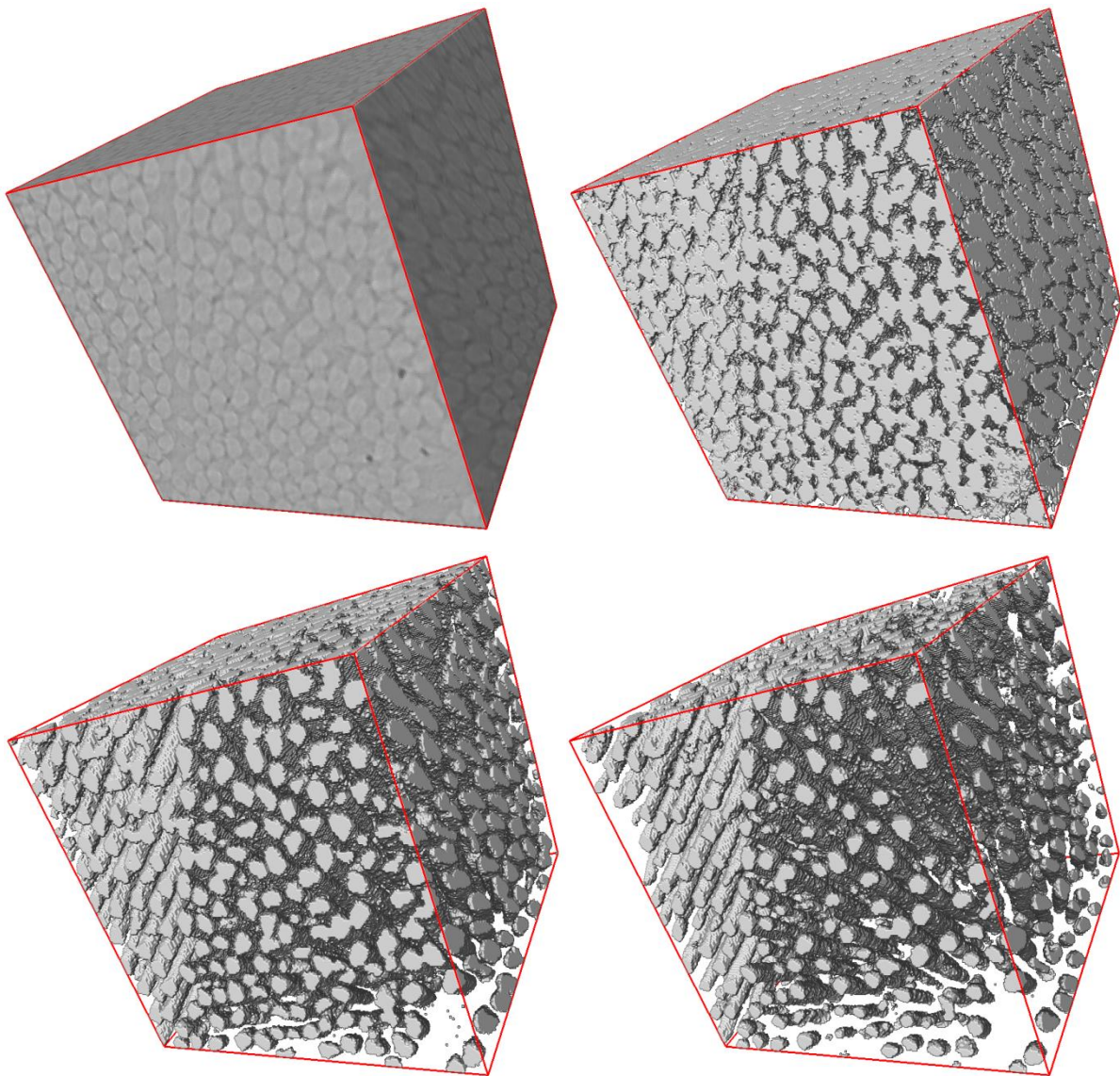


Figure 87 : En haut à gauche, vue d'une image de départ d'un matériau composite carbone-résine, de taille 200^3 voxels. A droite, segmentation par champs de Markov avec un paramètre $\beta = 0,1$. En bas, respectivement à gauche et à droite, segmentations avec des paramètres de $\beta = 0,3$ et $\beta = 0,5$.

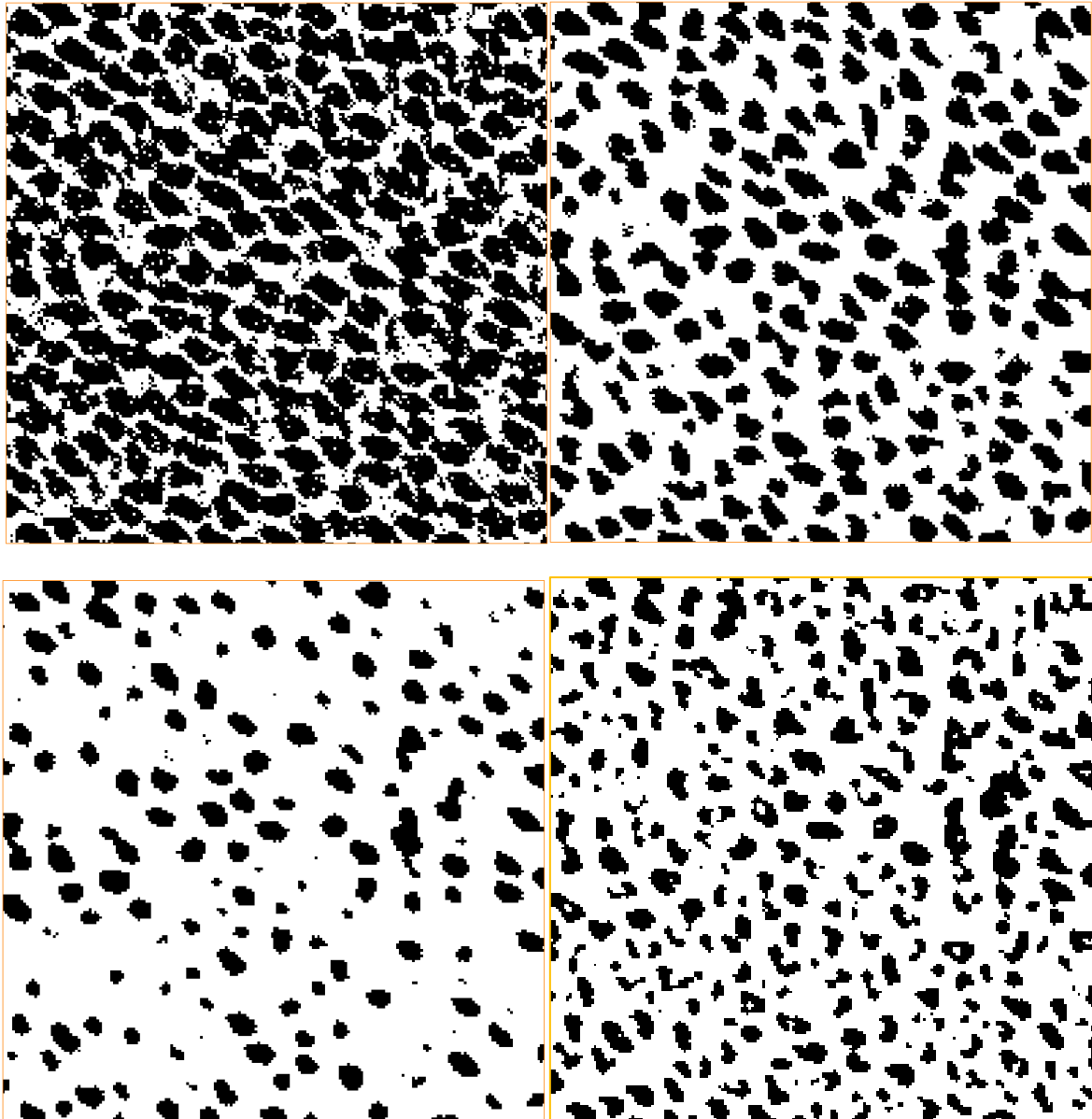


Figure 88 : Coupe des segmentations obtenues pour le matériau présenté sur la Figure 87. En haut, coupe pour les segmentations de Markov avec $\beta = 0, 1$ et $\beta = 0, 3$. En bas à gauche, coupe pour une segmentation de Markov avec $\beta = 0, 5$, et à droite, coupe d'une segmentation filtre médian+seuillage.

Suite à la segmentation par champs de Markov, on applique aux images l'algorithme de détection de fibres présenté dans la partie 2.4.2. On teste différents jeux de paramètres, en particulier en modifiant la tolérance τ et la méthode choisie (entre méthode de distance et méthode connexe). La quantification des résultats pourrait se faire simplement en suivant le nombre de fibres détectées dans l'image, ce qui peut nous amener à calculer manuellement un pourcentage de détection. Cependant, ce seul résultat occulte les éventuels autres défauts que peut occasionner la détection, comme des fausses fibres retrouvées ou plus simplement les déplacements des fibres occasionnés par les erreurs de détection des centres sur les faces (voir Figure 41 et Figure 42). Il est de toute façon

impossible, à partir d'une image réelle, de quantifier précisément le résultat, cela dépendant de l'importance que l'on accorde aux différents critères entre eux.

Pour autant, de la même manière que l'on a éliminé les droites aberrantes grâce à une triangulation de Delaunay (partie 2.4.2), on peut aussi quantifier en partie la qualité de la détection via la triangulation. On rappelle les présupposés sur les fibres qui sont utiles dans ce cas :

- Les fibres d'un même fil sont d'orientations proches. Celle-ci peut varier à l'intérieur, mais de manière « continue », c'est-à-dire que les fibres voisines d'un même fil partagent leur orientation à un ε près. Ainsi, si on note d la distance du grand cercle, et $(\theta_1, \phi_1), (\theta_2, \phi_2)$ les orientations de deux fibres voisines d'un même fil, on a $d((\theta_1, \phi_1), (\theta_2, \phi_2)) < \varepsilon$.
- Les fils sont constitués d'au moins 3 fibres de génératrices non coplanaires.
- Chaque fibre est reliée à au moins une autre fibre du même fil par une arête de Delaunay.
- Un fil correspond à l'ensemble des polyèdres définis par trois fibres dont trois des six sommets forment un triangle de Delaunay. On peut de plus supposer que comme les variations d'orientation sont négligeables à l'intérieur d'un même fil, les trois autres sommets du polyèdre forment aussi un triangle de Delaunay.

Pour chaque fibre f d'orientation (θ_f, ϕ_f) , on notera $\mathcal{D}(f)$ l'ensemble des fibres reliées à f par une arête de Delaunay. Pour toute fibre $g \in \mathcal{D}(f)$ d'orientation (θ_g, ϕ_g) , on peut calculer la distance du grand cercle entre les deux orientations. Selon les présupposés, chaque fibre à une voisine appartenant au même fil et dont les orientations sont proches à ε près. On note

$$\delta_f = \min_{g \in \mathcal{D}(f)} d((\theta_f, \phi_f), (\theta_g, \phi_g))$$

et on peut déduire des présupposés que, dans le cas où les droites sont bien détectées, on a

$$\forall f, \delta_f < \varepsilon.$$

On va donc quantifier chaque droite détectée f par δ_f .

On peut finalement construire les indicateurs, avec F le nombre total de droites,

$$\Delta = \frac{\sum_f \delta_f}{F}.$$

Δ est un indicateur de l'éloignement des orientations des droites à leurs plus proches voisines. Il est difficile de quantifier directement les résultats grâce à Δ , mais deux résultats différents sur la même image peuvent être comparés grâce à leurs valeurs de Δ respectives.

On présente ici, pour l'image de la Figure 87 (on choisit l'image segmentée avec $\beta = 0,3$ comme référence), les résultats de l'algorithme de détection de droites selon la tolérance et la méthode utilisée. Les couleurs servent à représenter l'orientation des fibres, deux fibres de couleurs proches ayant des orientations proches aussi. Deux tableaux récapitulatifs présentent aussi les nombres de droites détectées ainsi que les valeurs de Δ :

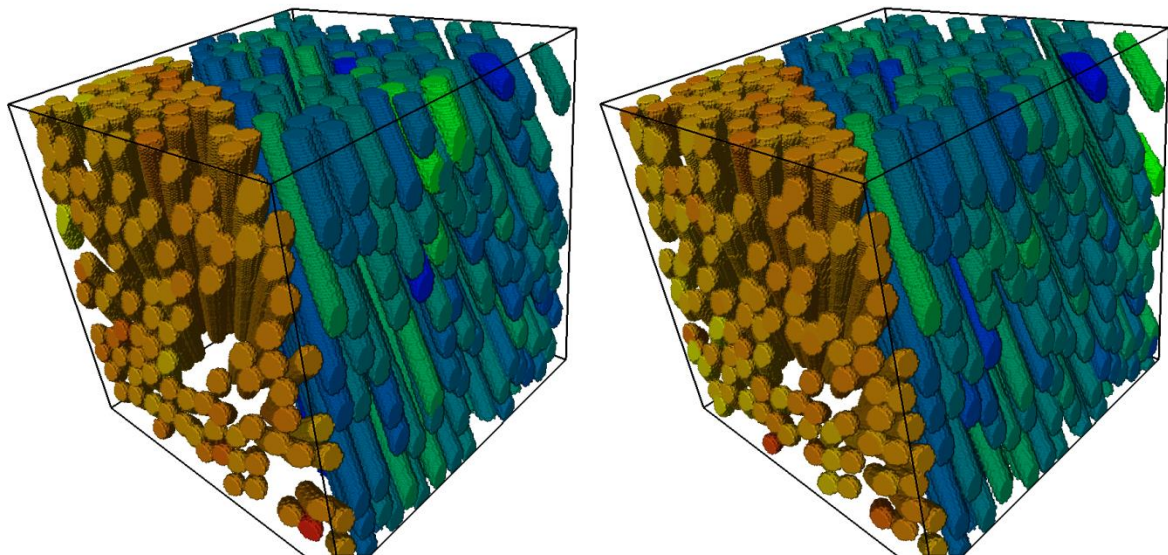


Figure 89 : Résultats de la détection de droites sur l'image présentée Figure 87 avec une tolérance $\tau = 0,1$. À gauche, la méthode utilisée est fondée sur les régions connexes, à droite sur les cartes de distance.

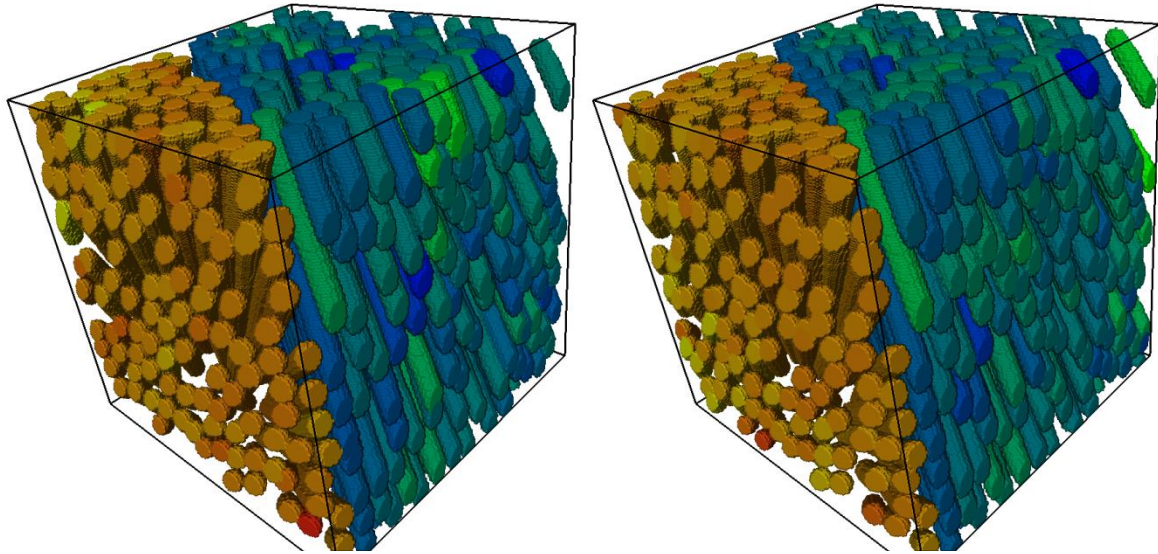


Figure 90 : Résultats de la détection de droites sur l'image présentée Figure 87 avec une tolérance $\tau = 0,3$. À gauche, la méthode utilisée est fondée sur les régions connexes, à droite sur les cartes de distance.

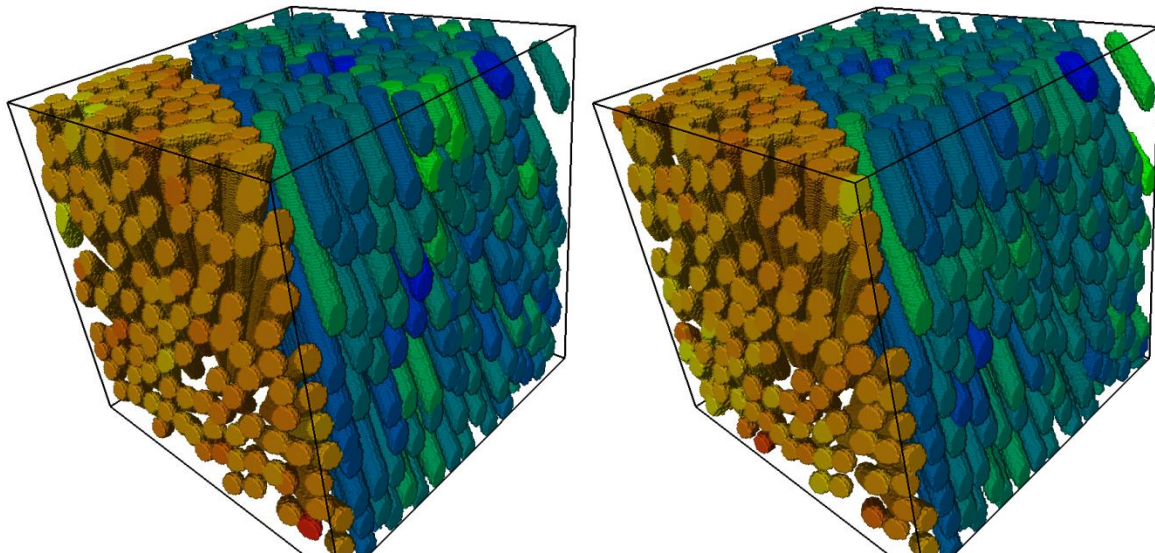


Figure 91 : Résultats de la détection de droites sur l'image présentée Figure 87 avec une tolérance $\tau = 0,5$. À gauche, la méthode utilisée est fondée sur les régions connexes, à droite sur les cartes de distance.

Méthode	$\tau = 0,1$	$\tau = 0,3$	$\tau = 0,5$
Régions Connexes	299	354	367
Carte de Distance	362	363	404

Tableau 3 : Nombre de droites détectées pour différentes tolérances.

Méthode	$\tau = 0,1$	$\tau = 0,3$	$\tau = 0,5$
Régions Connexes	$1,5 \times 10^{-3}$	$1,4 \times 10^{-3}$	$1,4 \times 10^{-3}$
Carte de Distance	$1,2 \times 10^{-3}$	$1,2 \times 10^{-3}$	$1,2 \times 10^{-3}$

Tableau 4 : Valeur de Δ pour différentes tolérances.

On peut observer, à partir de ces résultats que la méthode de carte de distance rend des résultats avec un meilleur Δ , ce que l'on peut constater sur les figures ci-dessus : en effet, on trouve une coloration moins uniforme sur les images de gauche correspondant aux régions connexes, et celles-ci sont réparties de manière assez aléatoire. Moins la répartition est uniforme, plus le δ_f sera localement élevé, ce qui augmente finalement Δ .

Concernant les temps de calcul, celui-ci est en moyenne de 6 ms par droite détectée pour la méthode des régions connexes, et de 15 ms par droite pour la méthode de cartes de distance.

Dans les images présentées plus haut, la suppression des droites aberrantes par triangulation de Delaunay n'est même pas nécessaire. Cependant, sur d'autres images, il est nécessaire de l'effectuer. La Figure 92 et la Figure 93 montrent un cas où cela est nécessaire. La première présente l'image de départ (de taille 400^3 voxels) et la recherche de droites effectuée, et la seconde figure présente les résultats de la classification sans élimination de Delaunay (à gauche) et avec (à droite). Les classes représentées par les couleurs bleu et jaune, proches en orientation, sont maintenant spatialement séparées. Il est cependant à noter que la méthode supprime aussi un peu d'informations pertinentes le long des arêtes du volume.

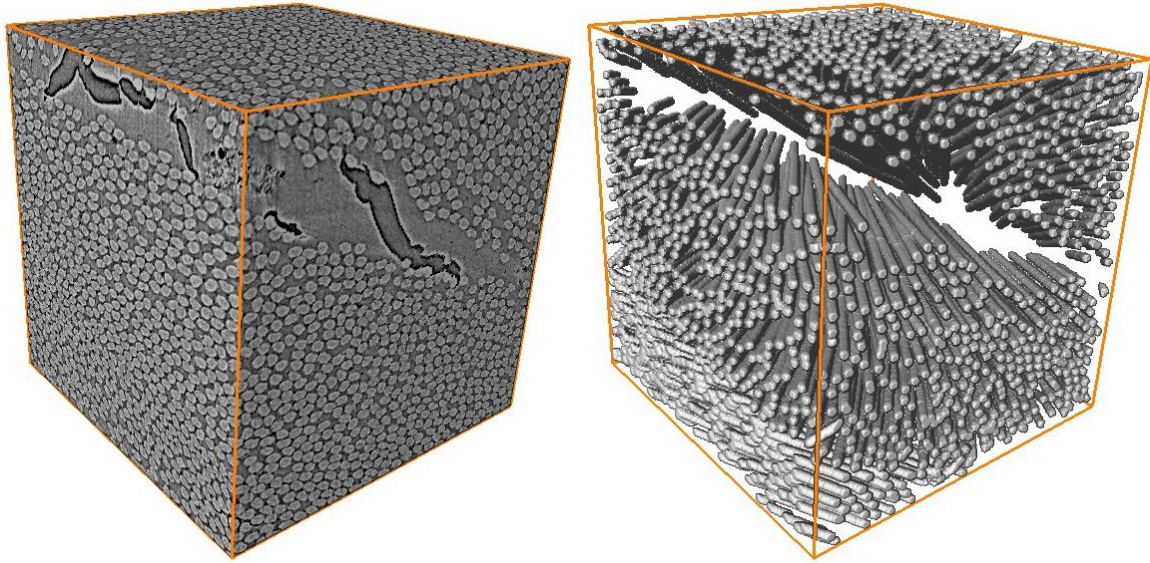


Figure 92 : À gauche, image de composite carbone-résine de taille 400^3 voxels. À droite, résultat de la détection de droites qui s'ensuit.

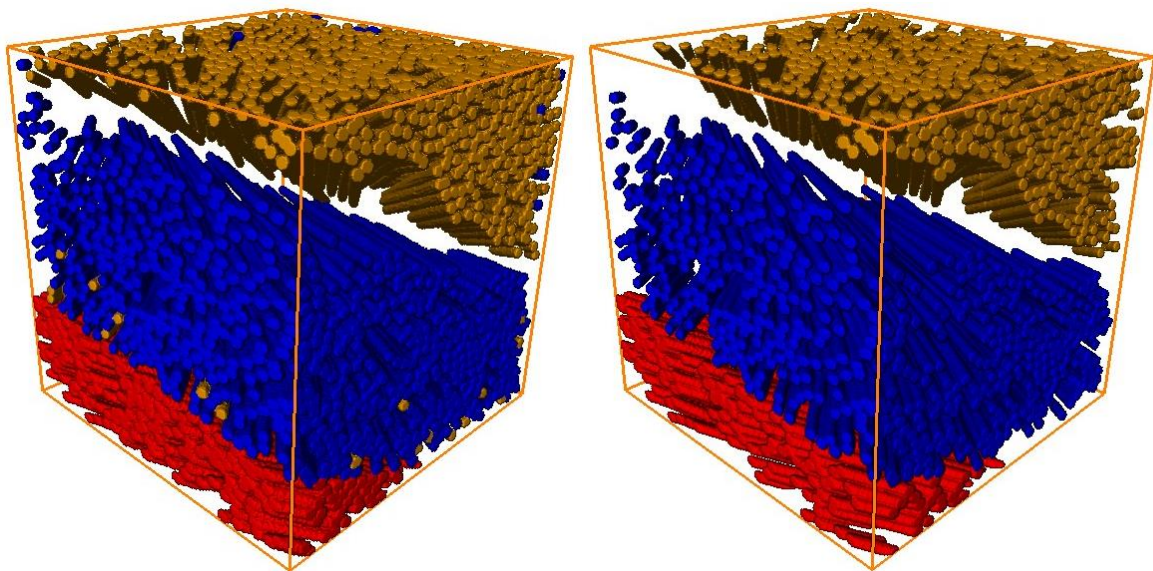


Figure 93 : Représentation de la classification obtenue par méthode des K-means sans élimination de Delaunay (à gauche) et avec (à droite).

Suite à la suppression des droites aberrantes, l'algorithme peut maintenant effectuer la classification, qui est déjà illustrée en partie sur la Figure 93. La Figure 94 présente des projections cartographiques de l'image sphérique obtenue avec les droites détectées sur la Figure 92, et l'image servant à l'initialisation via la méthode des noyaux. Finalement, la méthode des K-means permet le tri entre différentes classes des fibres (Figure 95).

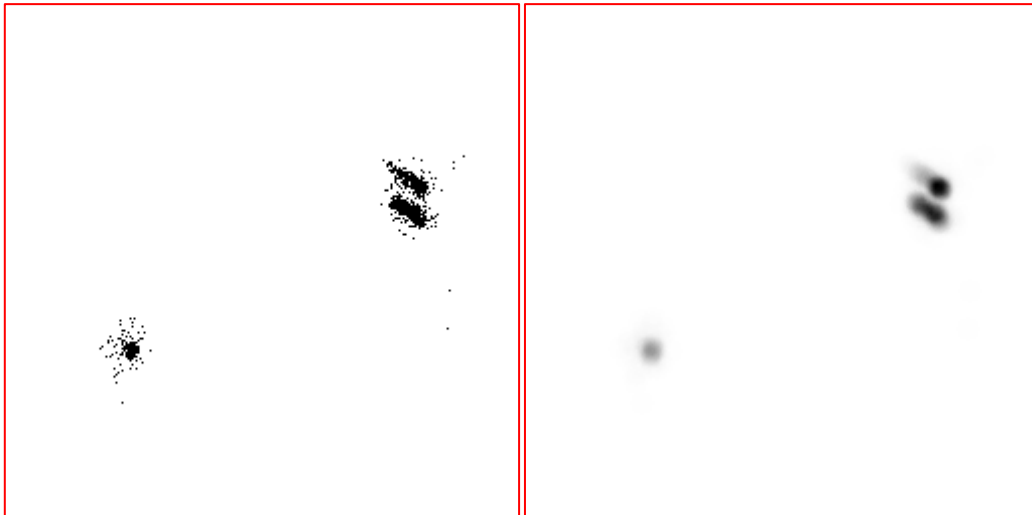


Figure 94 : Projection cartographiques des orientations des droites détectées à la Figure 92 (à gauche), et image utilisée pour l'initialisation des K-means (à droite)

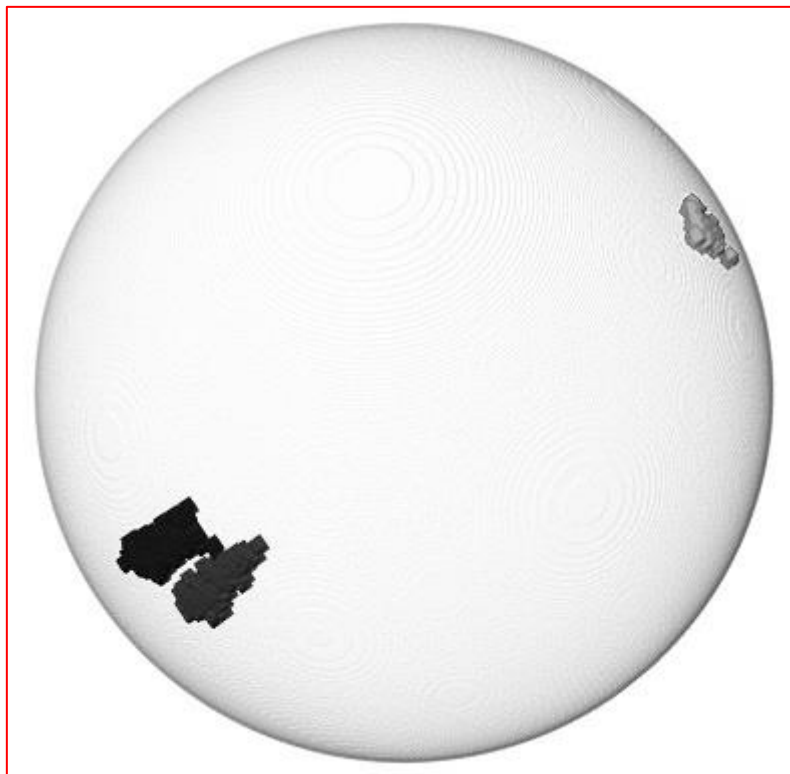


Figure 95 : Vue de la sphère de représentation des orientations. Chaque niveau de gris correspond à une classe retrouvée.

On a finalement présenté ici les résultats des méthodes développées sur des images de taille raisonnables, qui ne profitent pas complètement des gains de temps attendus par la complexité. Des images de taille bien plus conséquentes peuvent être étudiées de la même manière.

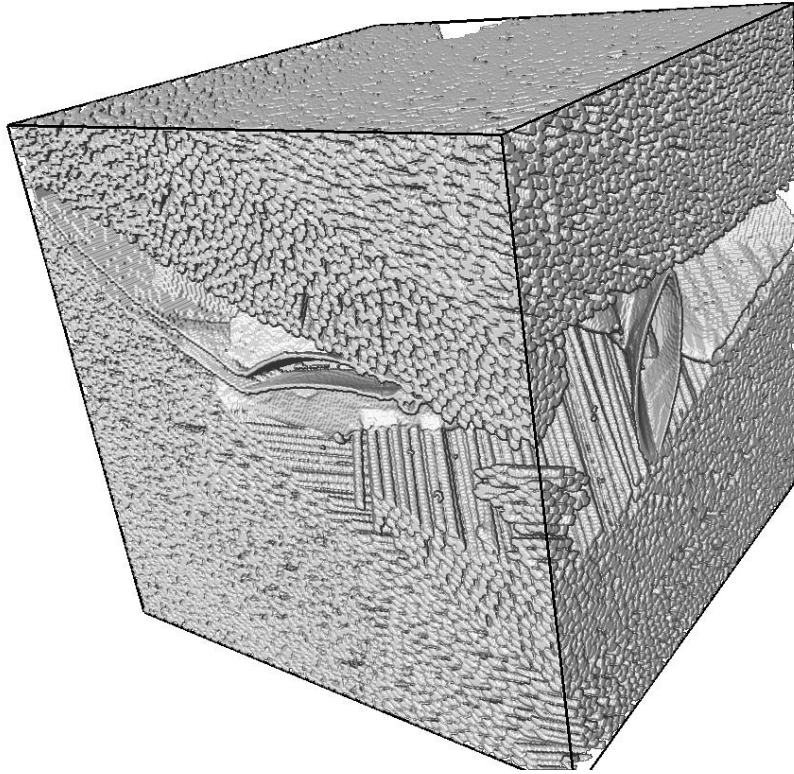


Figure 96 : Image segmentée de composite carbone-résine de taille 800^3 .

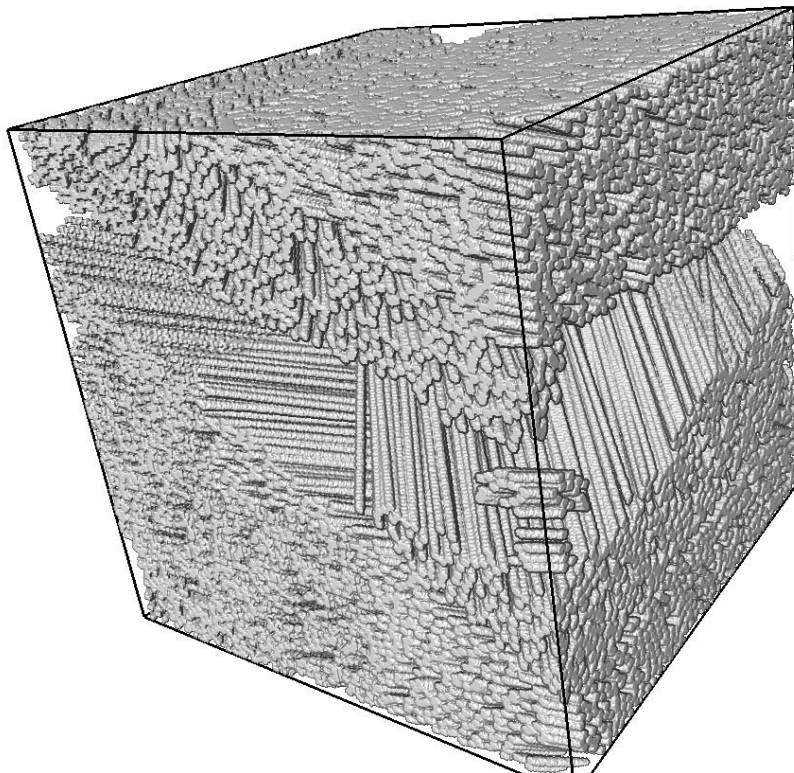


Figure 97 : Résultat de la détection de fibres sur l'image de la Figure 96, avec la méthode des régions connexes et une tolérance de 30 %.

La Figure 97 présente, par exemple, un résultat sur une image de composite carbone-résine de taille 800^3 . À ces tailles, le temps de calcul n'est plus aussi négligeable qu'auparavant. Des ordres de grandeur des temps de calcul sont présentés sur le tableau ci-dessous. Il est difficile de donner des temps précis car la méthode dépend de très nombreux paramètres extérieurs :

Taille	100^3	200^3	400^3	800^3
Régions connexes	10^{-1} s	10^0 s	$3 \cdot 10^1$ s	10^3 s
Carte de distance	$4 \cdot 10^{-1}$ s	$6 \cdot 10^0$ s	$2 \cdot 10^2$ s	$6 \cdot 10^3$ s
Segmentation	$3 \cdot 10^{-1}$ s	$2 \cdot 10^0$ s	$2 \cdot 10^1$ s	10^2 s
Détections de fils	$5 \cdot 10^{-1}$ s	$2 \cdot 10^0$ s	10^1 s	$5 \cdot 10^1$ s
Total avec régions connexes	1 s	5 s	60 s	1150 s

Tableau 5 : Ordre de grandeur des temps de calcul pour les différentes parties de la méthode.

Pour des images de taille encore supérieure, on utilise le découpage en sous-blocs pour effectuer la détection. Ici, les images les plus grandes que l'on peut obtenir à partir des tomographies disponibles sont de taille $1400 \times 1400 \times 2000$ voxels, soit près de 4 milliards d'éléments. La segmentation demandant des calculs d'actualisation globaux, celle-ci doit être faite en une seule fois, ce qui représente une difficulté supplémentaire (la taille de l'image dépassant les tailles maximales admises pour certaines bibliothèques de calcul parallèles, qui sont codées sur des entiers signés `int`, i.e. jusqu'à $2^{31} - 1$, ce qui ne dépasse pas les 3 milliards). La détection de lignes, ensuite, peut être faite par sous-blocs, que l'on réunira ensuite. Le tableau ci-dessous présente les ordres de grandeur des temps de calcul dans le cas d'un découpage en sous-blocs de 200^3 :

Taille	200^3	400^3	800^3
Temps de calcul	4 s	35 s	300 s
Accélération	1	1,1	3,3

Tableau 6 : Ordre de grandeur des temps de calcul totaux (hors segmentation) pour une image découpée en sous-blocs de 200^3 voxels.

Les Figure 98, Figure 99 et Figure 100 présentent les résultats sur une image de taille maximale. La détection se fait par sous-blocs de taille 200^3 , qui sont ensuite rassemblés pour former le résultat final. La détection de lignes seule se fait en environ 20 minutes, ce

qui représente un temps très raisonnable pour des images d'une telle taille. On observe, sur ces images, un résultat intéressant de gradient d'orientation des fibres d'un même fil, que l'on observe via des dégradés de couleurs sur la Figure 100.

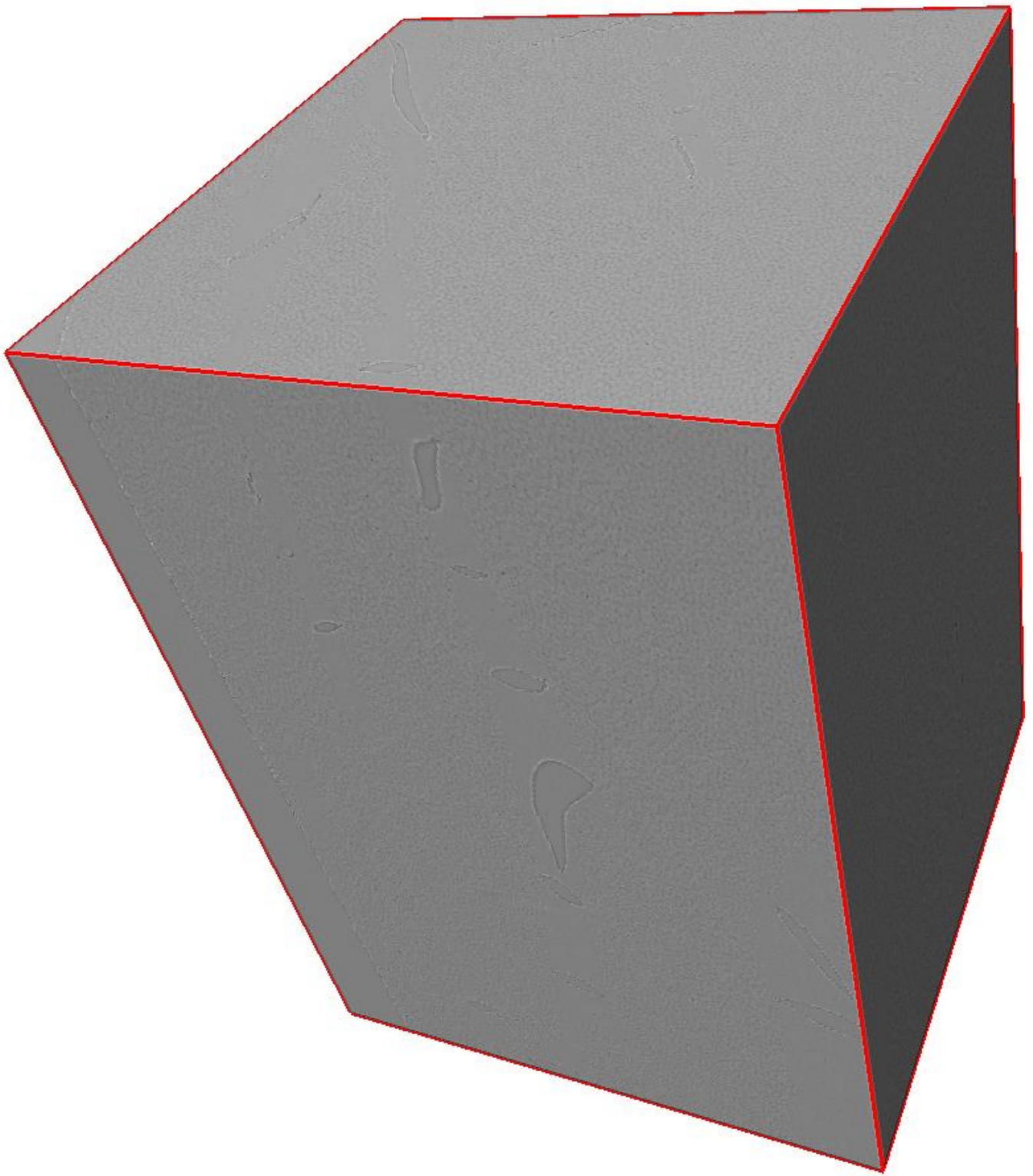


Figure 98 : Image de départ d'un composite carbone-résine de taille $1400 \times 1400 \times 2000$.

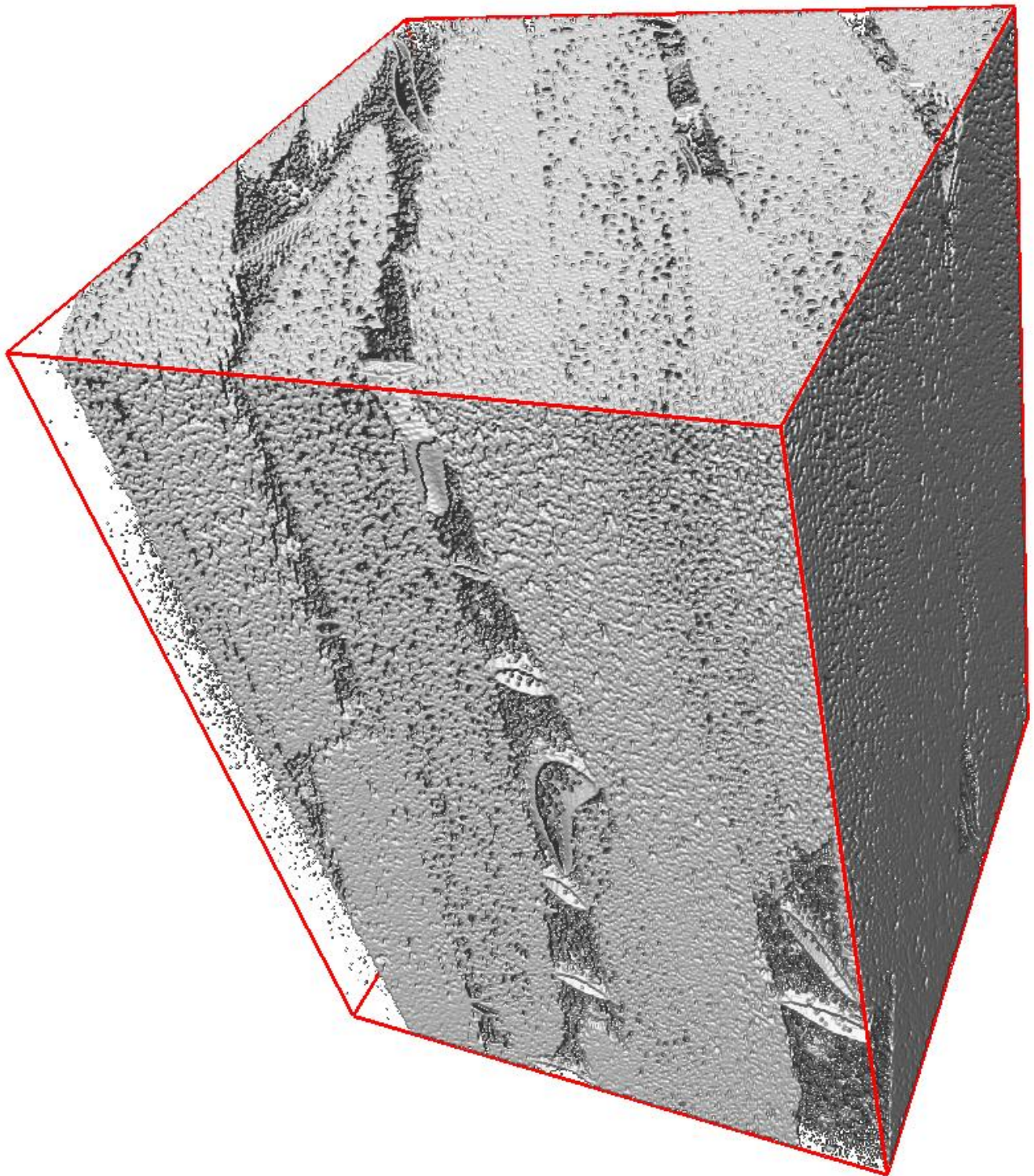


Figure 99 : Résultat de la segmentation par méthodes de champs de Markov de l'image présentée sur la Figure 98.

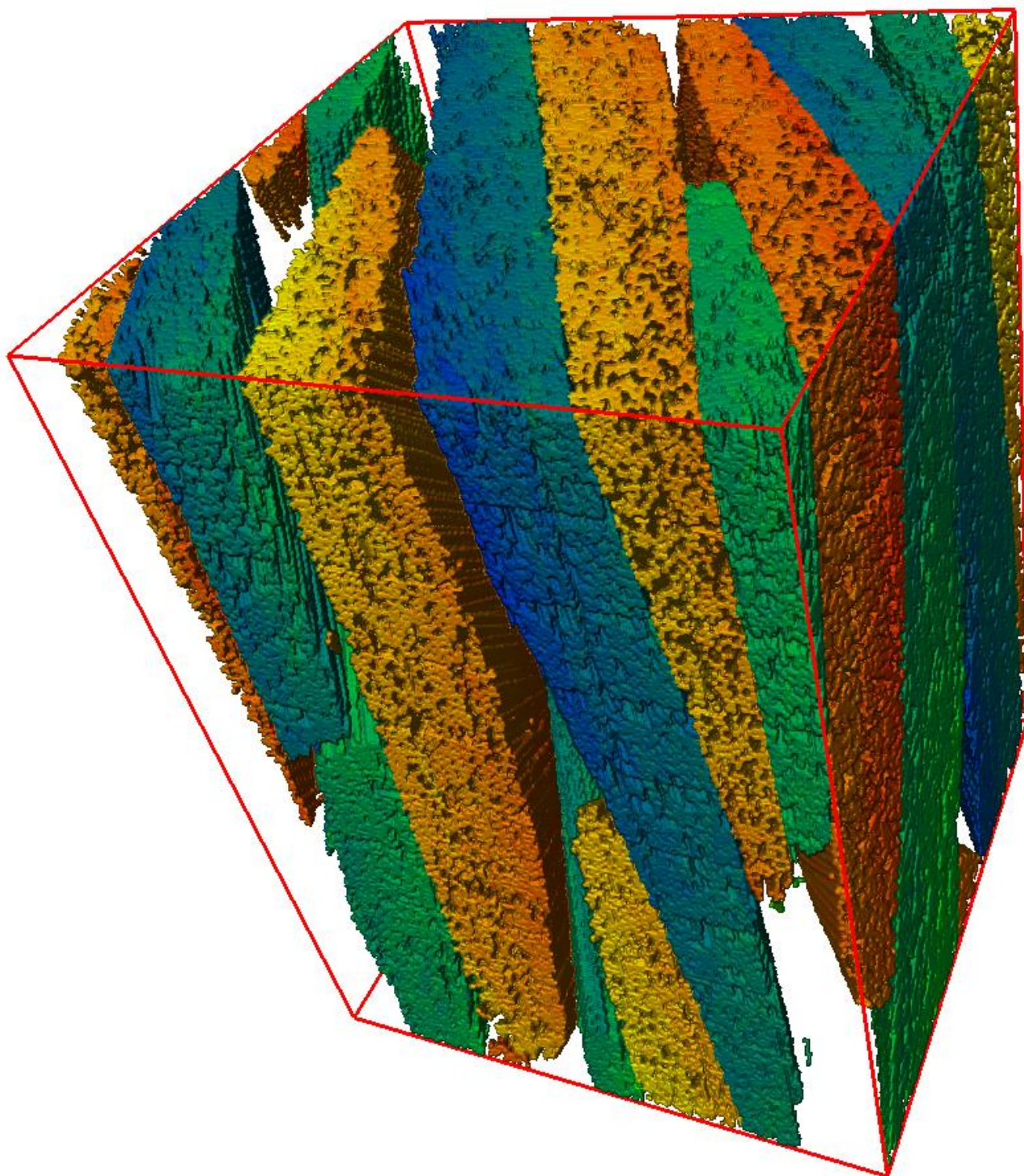


Figure 100 : Résultat de la détection de fibres sur l'image de la Figure 99 par blocs de 200^3 . Les couleurs représentent les orientations des fibres.

Concernant l'algorithme de tracé des fils, on présente ici les résultats sur une sous-image de taille $1000 \times 1000 \times 200$, présentée sur la Figure 101. La difficulté de la phase de détection de fils est l'initialisation des K-means afin de ne pas réunir de fils entre eux (comme sur la Figure 102). Il est plus appréciable, dans la majorité des cas, d'initialiser les K-means avec trop de classes de départ (Figure 103), et d'effectuer ensuite une phase de fusion pour arriver au résultat attendu sur chaque sous-bloc (Figure 104), et finalement sur l'image entière (Figure 105 et Figure 106). On constate une difficulté de fusion des fils de très petite taille en bas à droite de la Figure 105, mais la considération d'une image de taille supérieure aurait suffi à obtenir des contacts plus importants sur d'autres faces et donc à fusionner ces fils. À partir du résultat présenté Figure 106, on peut maintenant réduire le nombre d'éléments nécessaires à une simulation en fusionnant les voxels des mêmes classes entre eux à l'intérieur des fils, et en ne gardant les voxels de taille originale qu'aux interfaces. En représentant l'image par un arbre 8-aire simple, on peut déjà simplifier une sous-image choisie au hasard de taille $896 \times 896 \times 128 = 102\,760\,448$ voxels en un arbre à 14 100 520 feuilles. Sans perte d'information, on représente donc le matériau à l'échelle du fil avec 7,3 fois moins d'éléments que pour une représentation voxel par voxel.

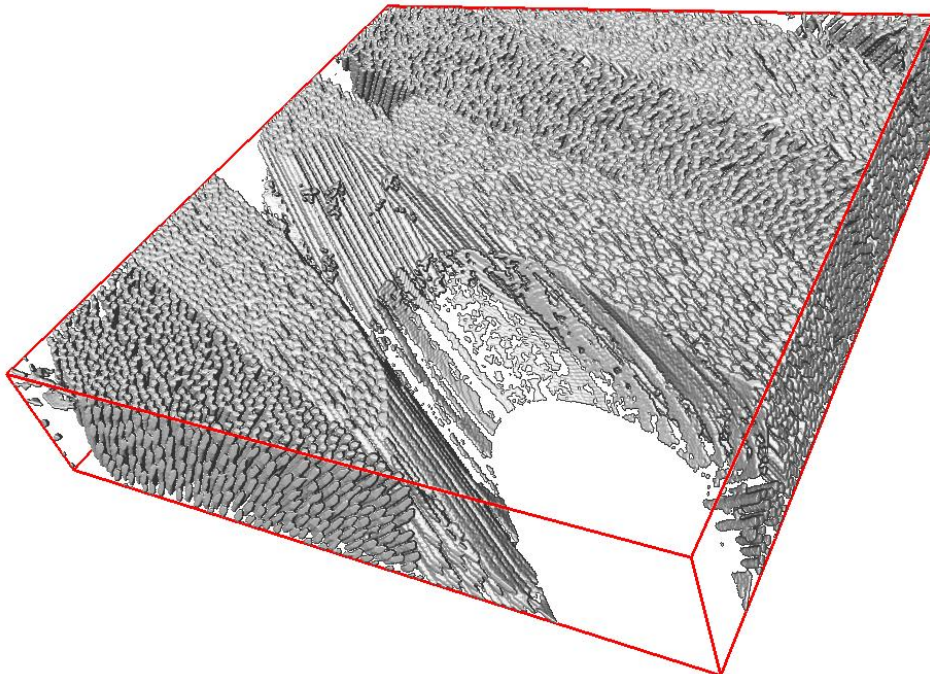


Figure 101 : Image de composite carbone-résine taille $1000 \times 1000 \times 200$ segmentée.

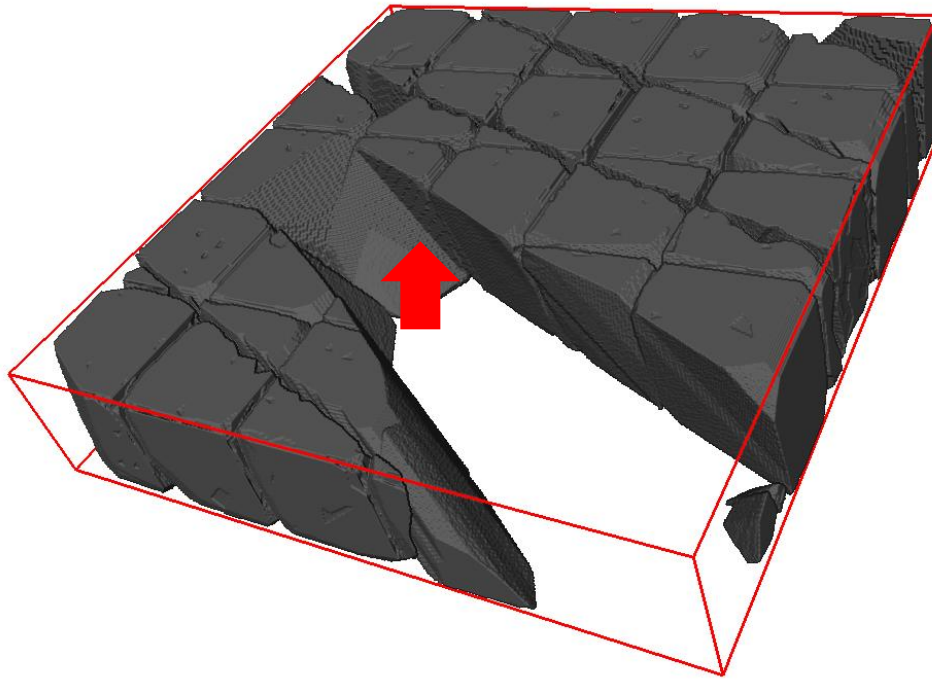


Figure 102 : Détection de fils avec une initialisation trop regroupée. Les deux fils marqués par la flèche rouge sont fusionnés et se rejoignent finalement.

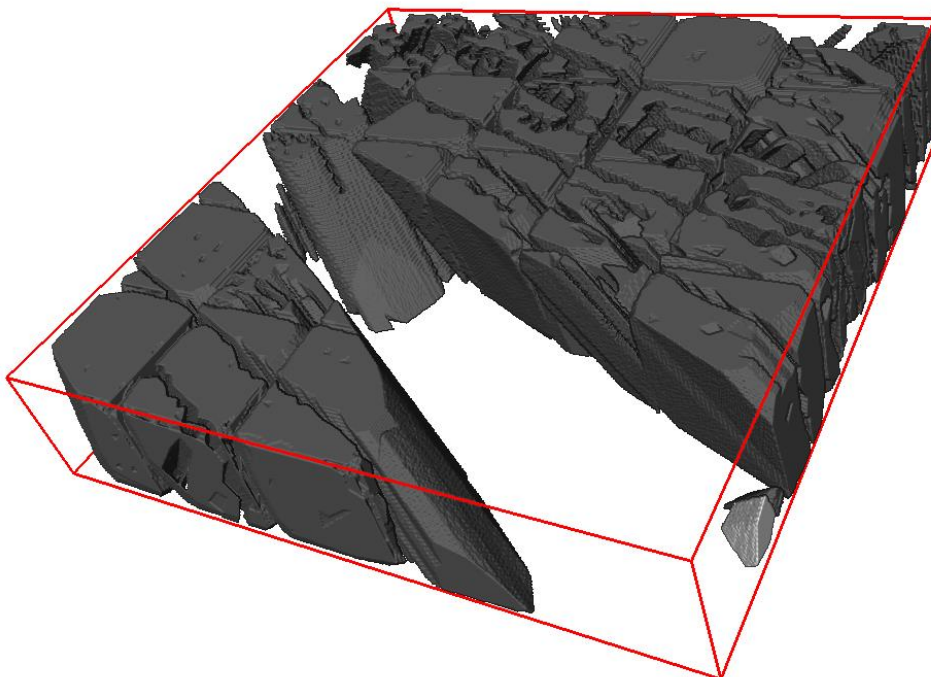


Figure 103 : Détection de fils avec une initialisation pas assez regroupée. Les structures creusées sont dues à la séparation d'un unique fil en plusieurs classes. Il est nécessaire de fusionner ces régions entre elles.

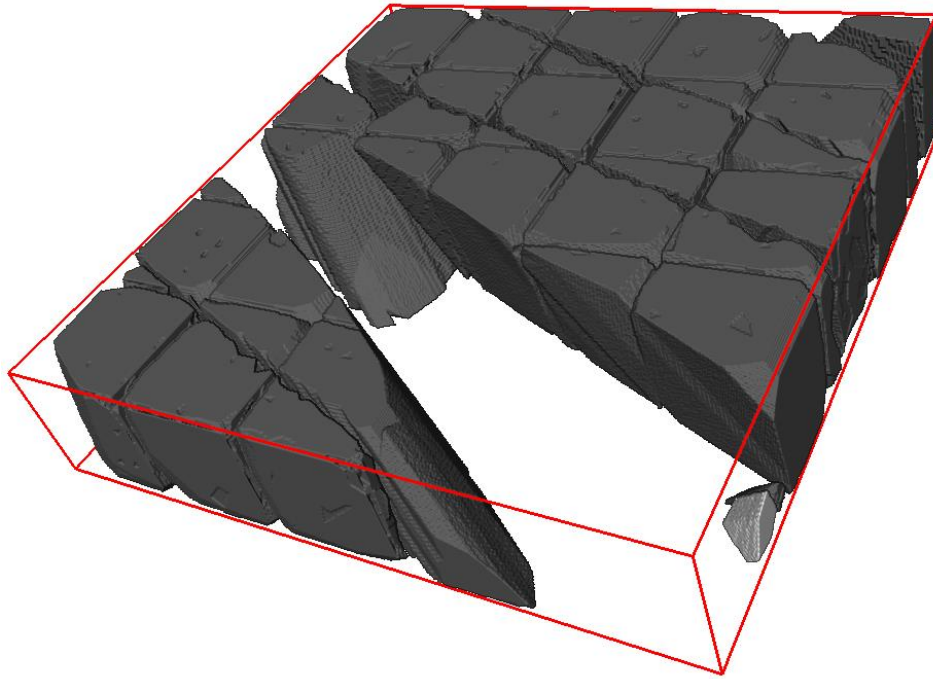


Figure 104 : Résultat de la fusion des classes de la Figure 103. On a bien la séparation entre tous les fils de l'image.

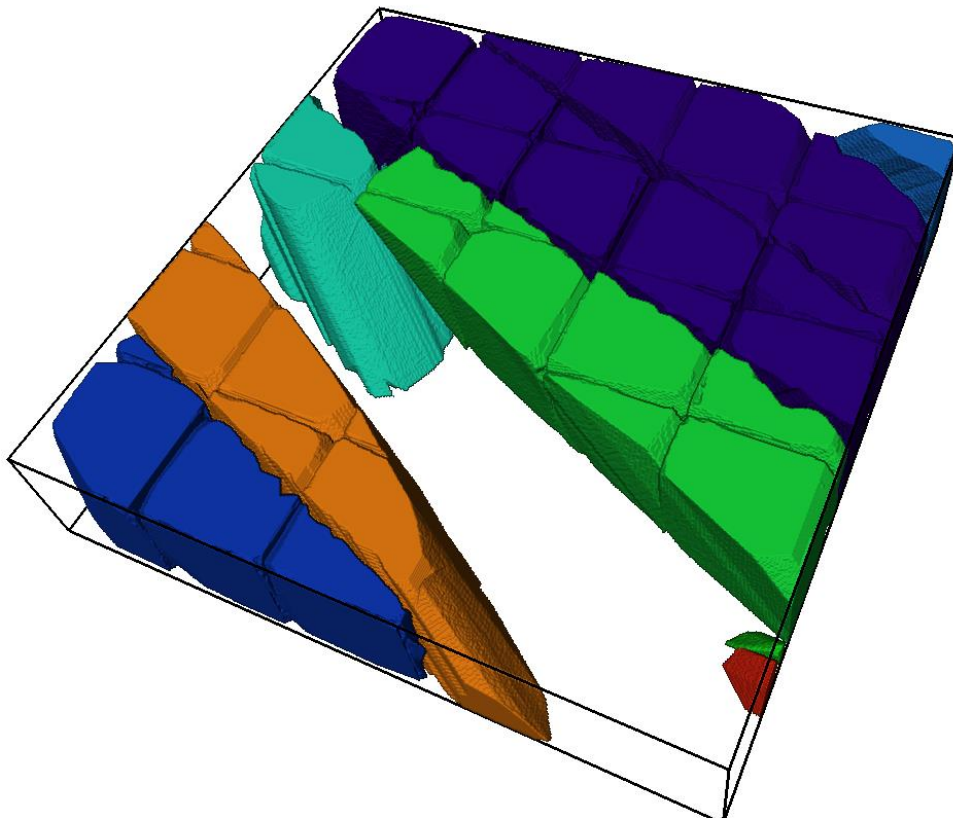


Figure 105 : Résultat final de la fusion des fils entre sous-blocs. Le défaut en bas à droite entre les deux petit fils verts et brique est dû à leur petite taille, et serait corrigé si l'on ajoutait des blocs consécutifs.

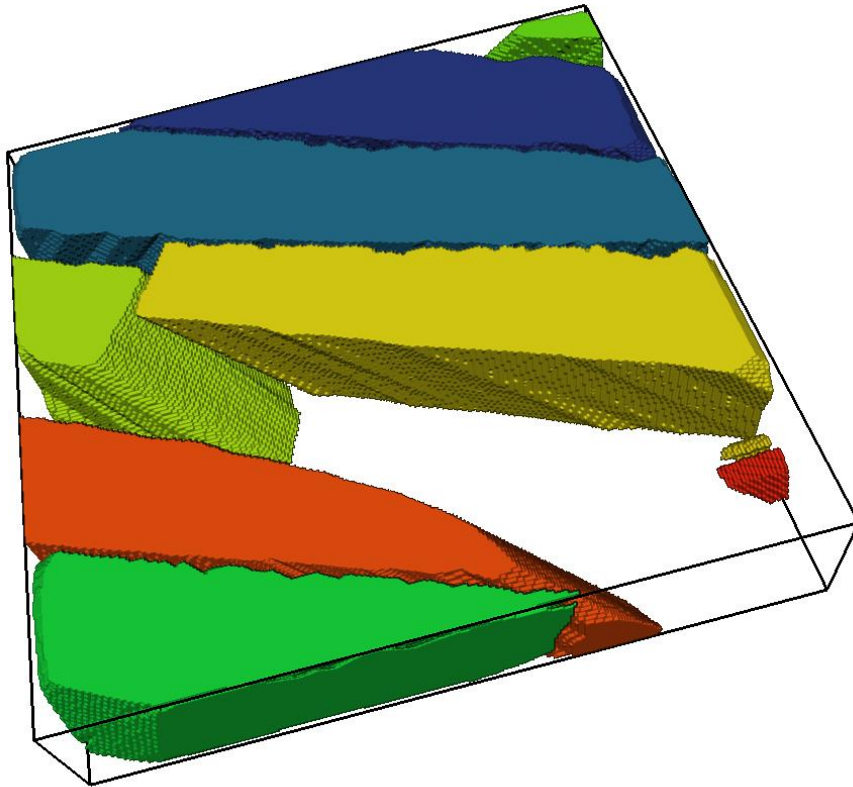


Figure 106 : Résultat final de la fusion de fils, effectué avec une mutualisation des fibres de chaque fil.

Sur des images à résolution inférieure ($1,62 \mu\text{m}/\text{vx}$, Figure 107), les algorithmes de détection de fibres présentés fonctionnent aussi de la même manière, quoiqu'offrant des détections moins performantes en raison de la plus grande proximité entre les fibres sur l'image et de leurs rayons inférieurs.

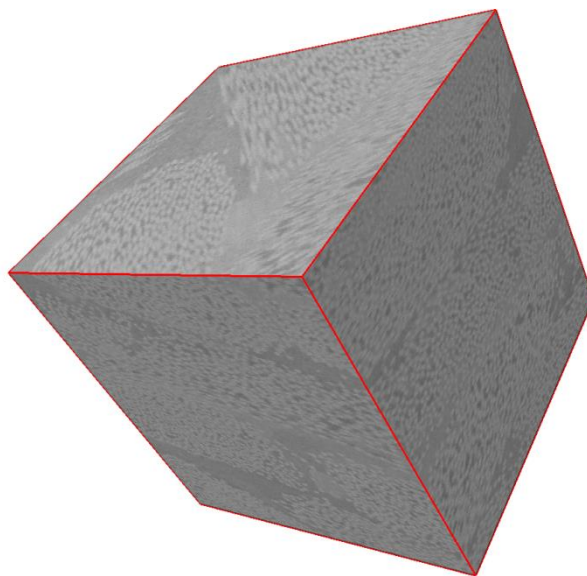


Figure 107 : Image de composite carbone-résine de taille 400^3 voxels, à une résolution de $1,62 \mu\text{m}/\text{vx}$.

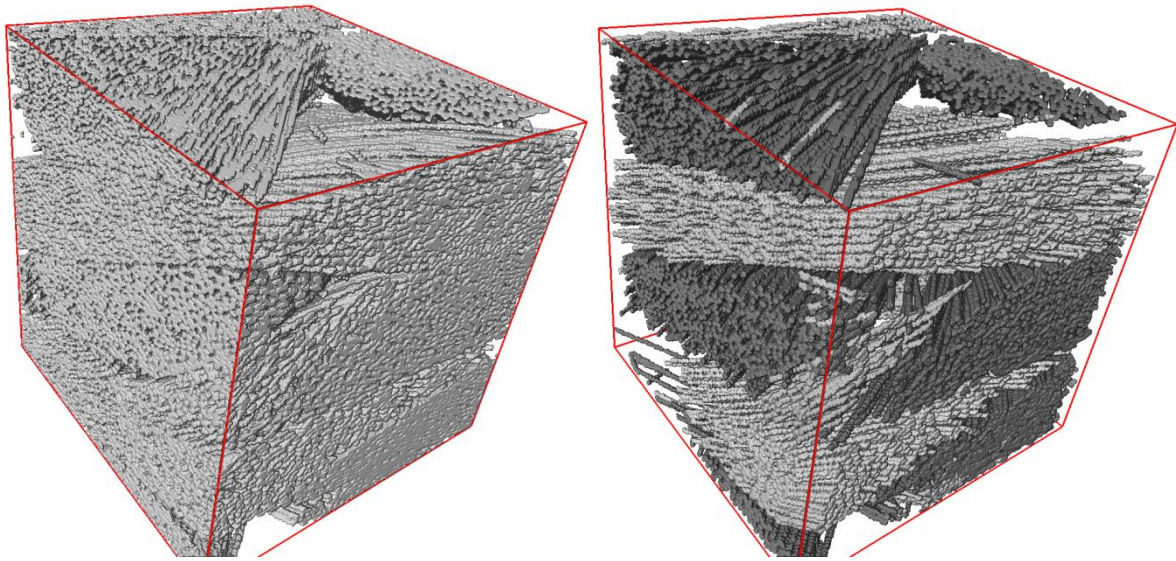


Figure 108 : À gauche, image segmentée issue de la Figure 107. À droite, résultat de la détection de droites. La détection permet toujours de repérer l'enveloppe du fil, mais certains endroits sont moins fournis.

Finalement, on peut effectuer des simulations sur les images virtuelles fournies, comme présenté sur la Figure 109, qui montre le résultat d'une simulation plaque chaude gardée, c'est-à-dire le profil thermique dans le matériau soumis à une température fixée chaude sur une face, une température froide sur la face opposée, et un comportement adiabatique sur les autres faces. Ce type de simulation est particulièrement utile pour déterminer la conductivité directionnelle du matériau testé.

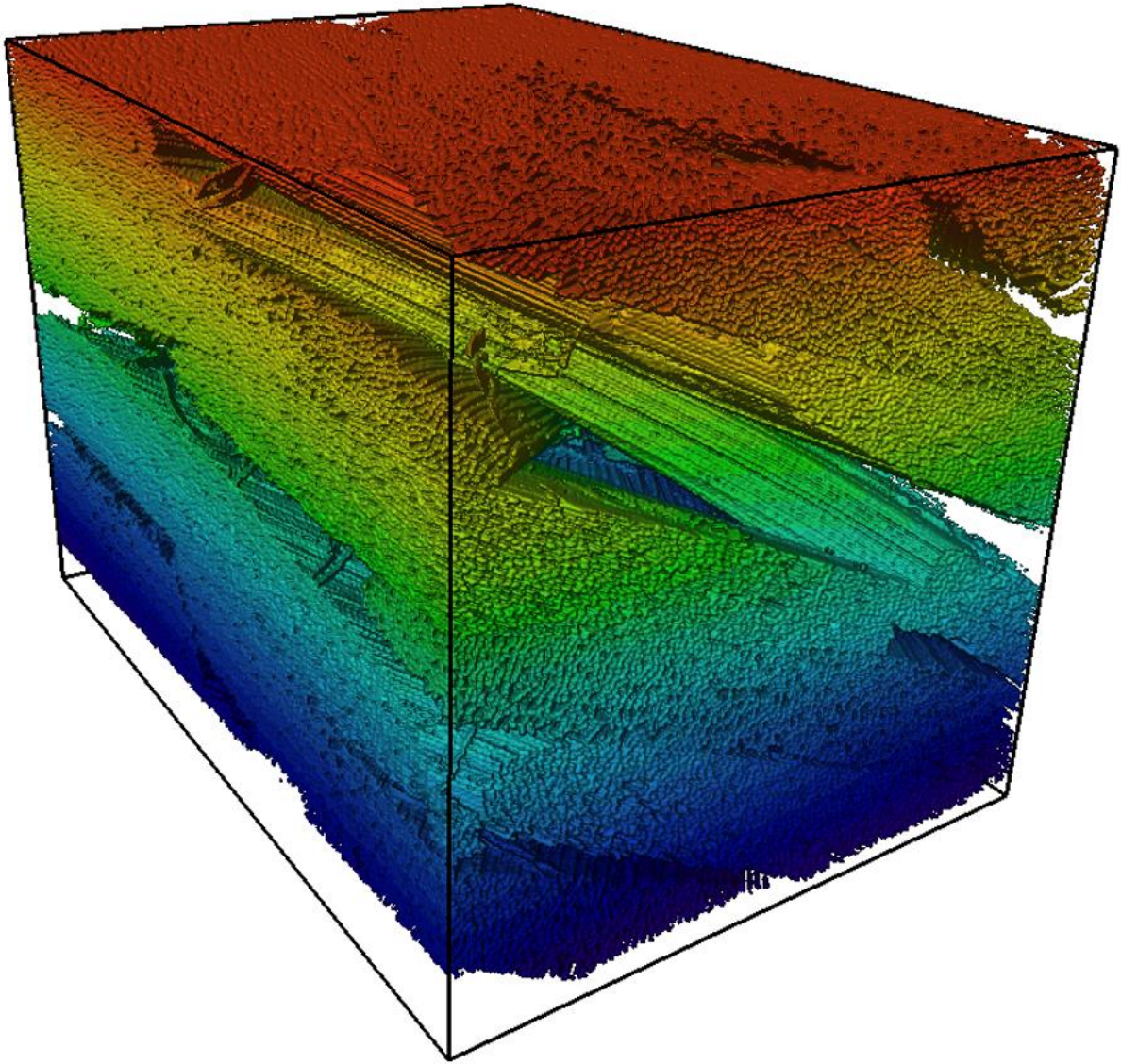


Figure 109 : Résultat sur les fibres d'une simulation plaque chaude gardée sur le matériau présenté en Figure 98.

4.2. Mousses de polyuréthane

Sur les images de mousses de polyuréthane présentées précédemment dans la partie 3, le but de l'étude est de retrouver l'ensemble des sphères afin de connaître la distribution des rayons des sphères dans le matériau, ainsi que leur répartition spatiale. Ces données permettent alors de valider ou d'invalides des procédés de fabrication de matériaux cherchant à atteindre des répartitions de taille bien particulières. Il est aussi possible de créer virtuellement d'autres mousses aux répartitions proches de celles observées pour étudier leur comportement thermique ou mécanique. Sur les mousses de polyuréthane, des tests mécaniques et des modélisations en compression sont effectués, et la connaissance précise de la structure du matériau permet alors de relier répartition de sphères et propriétés mécaniques précises. La conception virtuelle de matériaux aux répartitions proches nous permet alors, via les algorithmes de modélisation mécanique, d'observer le comportement de différentes variantes du matériau original, et de déduire ensuite si l'amélioration de certaines propriétés est suffisante pour réfléchir à des procédés de fabrication différents permettant d'obtenir les répartitions recherchées. Ici, la fabrication additive rend envisageable toute répartition de sphères, tant que celle-ci est physiquement possible.

La segmentation des images de mousse de polyuréthane est plus simple que dans le cas des images tomographiques portant sur les autres matériaux. Dans ce cadre, un calcul de gradient grâce à un filtre de Sobel suivi d'un seuillage simple est suffisant pour obtenir une segmentation de très bonne qualité (Figure 110). L'utilité du gradient porte sur le fonctionnement de l'algorithme de détection, qui détecte des sphères et non des boules. Les vides de la mousse de polyuréthane donnant des boules sur les images, le gradient permet ainsi de se ramener à l'étude de sphères. Comme on le voit sur les Figure 110, Figure 111 et Figure 112, les structures obtenues suite à la segmentation ressortent facilement et semblent plutôt simples à détecter.

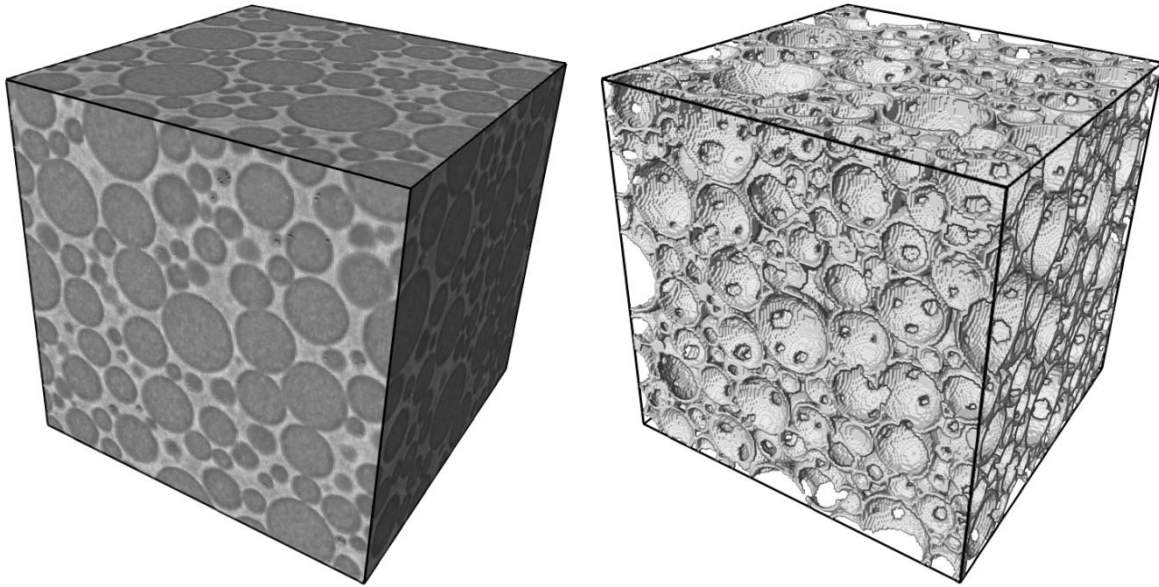


Figure 110 : Image initiale d'une mousse de polyuréthane de taille 300^3 voxels (à gauche), et résultat de l'application d'un filtre de Sobel suivie d'un seuillage (à droite).

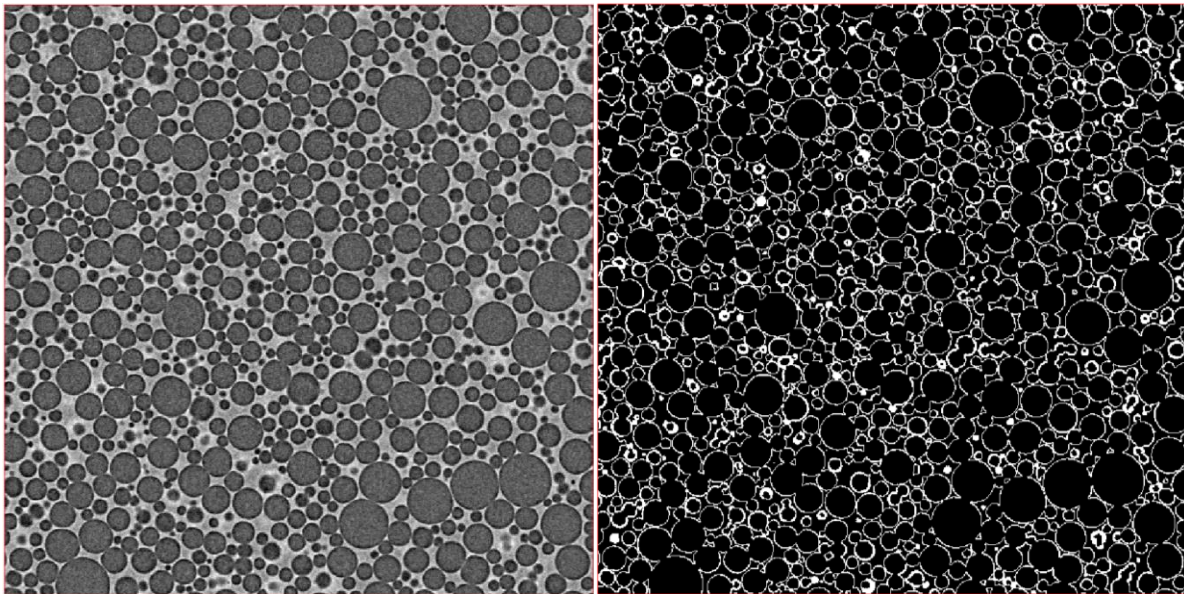


Figure 111 : Coupe de taille 500^2 d'image tomographique de mousse polyuréthane (à gauche), et résultat de la segmentation sur la même coupe (à droite).

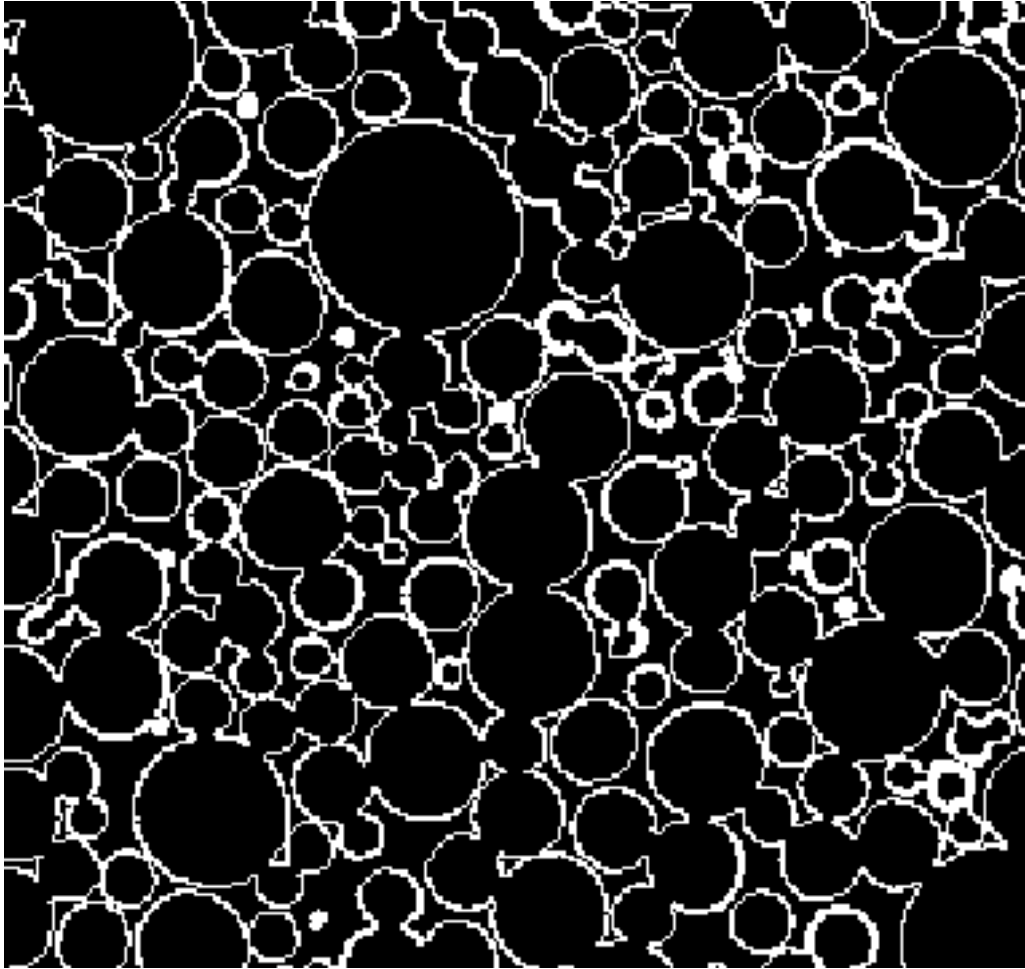


Figure 112 : Détail de la coupe de la segmentation présentée Figure 111. Malgré des légères déformations visibles à l'œil nu, les structures sont assez proches de cercles.

Comme on le voit sur la Figure 112, les structures recherchées ne sont pas exactement sphériques. Les déformations sont assez faibles pour ne pas entraîner de problèmes sur la détection des centres, mais la vérification du rayon nécessite une tolérance non-nulle pour obtenir des résultats corrects. Ainsi, sur l'ensemble de l'image présentée Figure 113, qui correspond à une mousse de polyuréthane de taille $1000 \times 1000 \times 800$ voxels et environ 10^5 sphères détectées, la première sphère acceptée ne comporte que 90 % de points à son rayon retrouvé, et la majorité d'entre elles ne sont acceptées que pour des tolérances proches de 0,4, c'est-à-dire qu'elles ne présentent que 60 % des points à leurs rayons respectifs.

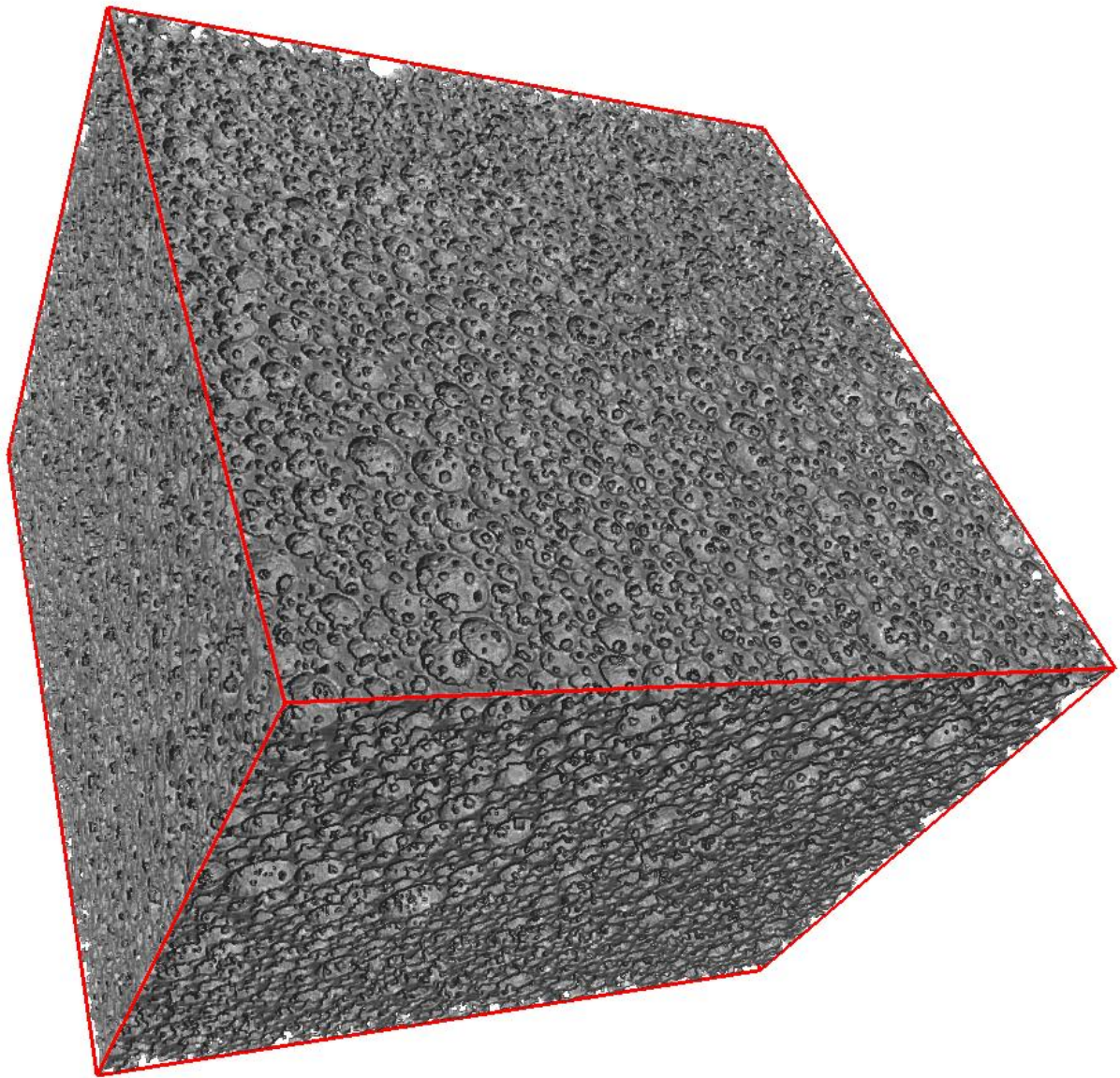


Figure 113 : Image segmentée d'une mousse de polyuréthane de taille $1000 \times 1000 \times 800$ voxels.

La détection se fait aisément avec une tolérance de 0,4, comme le montre la Figure 114, qui superpose les résultats à l'image de départ sur une coupe. La très grande majorité des sphères sont retrouvées (celles qui semblent non retrouvées et sont de petite taille sont pour la majeure partie effectivement retrouvées et leur représentation débute sur les coupes voisines à cause d'arrondis dus aux déformations), et aucune fausse sphère ne semble apparaître. La densité de ces images rend tout de même l'étude assez proche de la limite, et pour une tolérance de 0,45 (Figure 115), on voit apparaître les premières fausses sphères sur l'image.

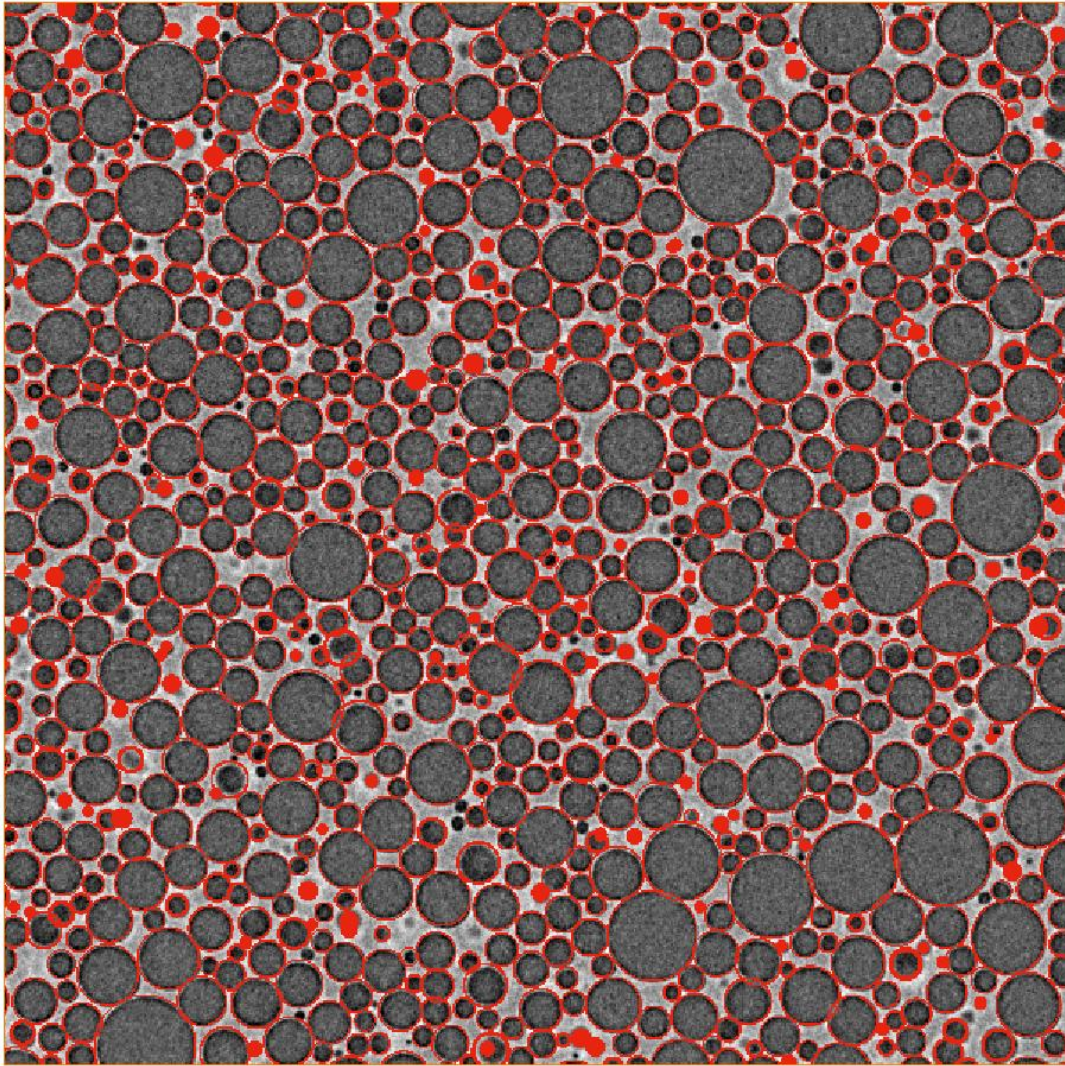


Figure 114 : Coupes superposées de l'image de départ et des sphères retrouvées (cercles rouges) avec une tolérance de 0,4 (taille 500^2 pixels).

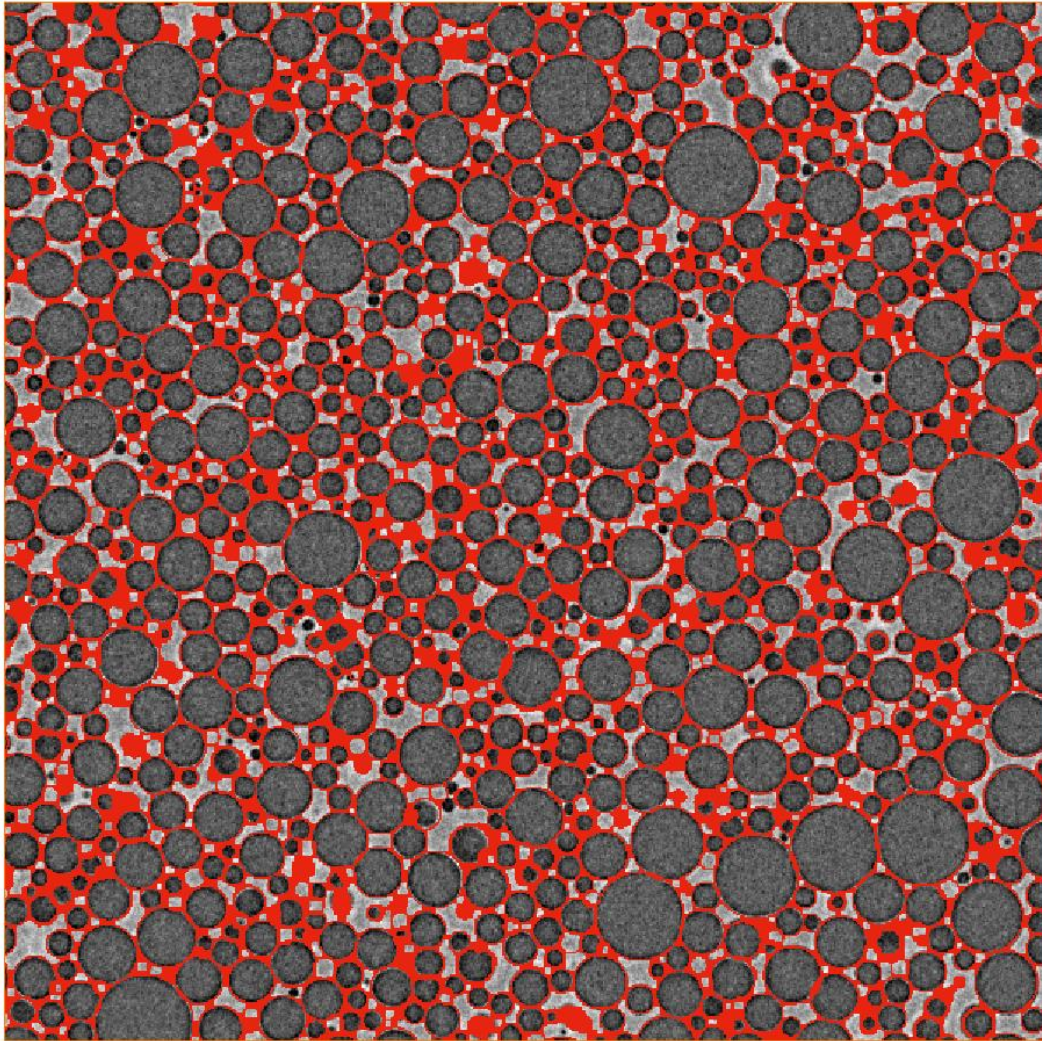


Figure 115 : Coupes superposées de l'image de départ et des sphères retrouvées (cercles rouges) avec une tolérance de 0,45 (taille 500^2 pixels).

Les résultats de la détection de sphères avec une tolérance de 0,4 sont présentés sur la Figure 116. L'échelle de couleur représente le rayon de la sphère. Comme on l'aperçoit, la très grande majorité des sphères est de rayon faible, et la densité est relativement forte. Le rayon maximal des sphères sur cet échantillon est de 42, et la répartition des tailles de sphères est représentée sur la Figure 117. Cette répartition approche la densité de probabilité d'une loi bêta sur $[r_{min}, r_{max}]$.

Taille de l'image	100^3	200^3	400^3	800^3
Temps de calcul (s)	0,1	0,5	5	40

Tableau 7 : Temps de calcul en fonction de la taille.

Le Tableau 7 donne des ordres de grandeur des temps de calcul sur différentes tailles d'images.

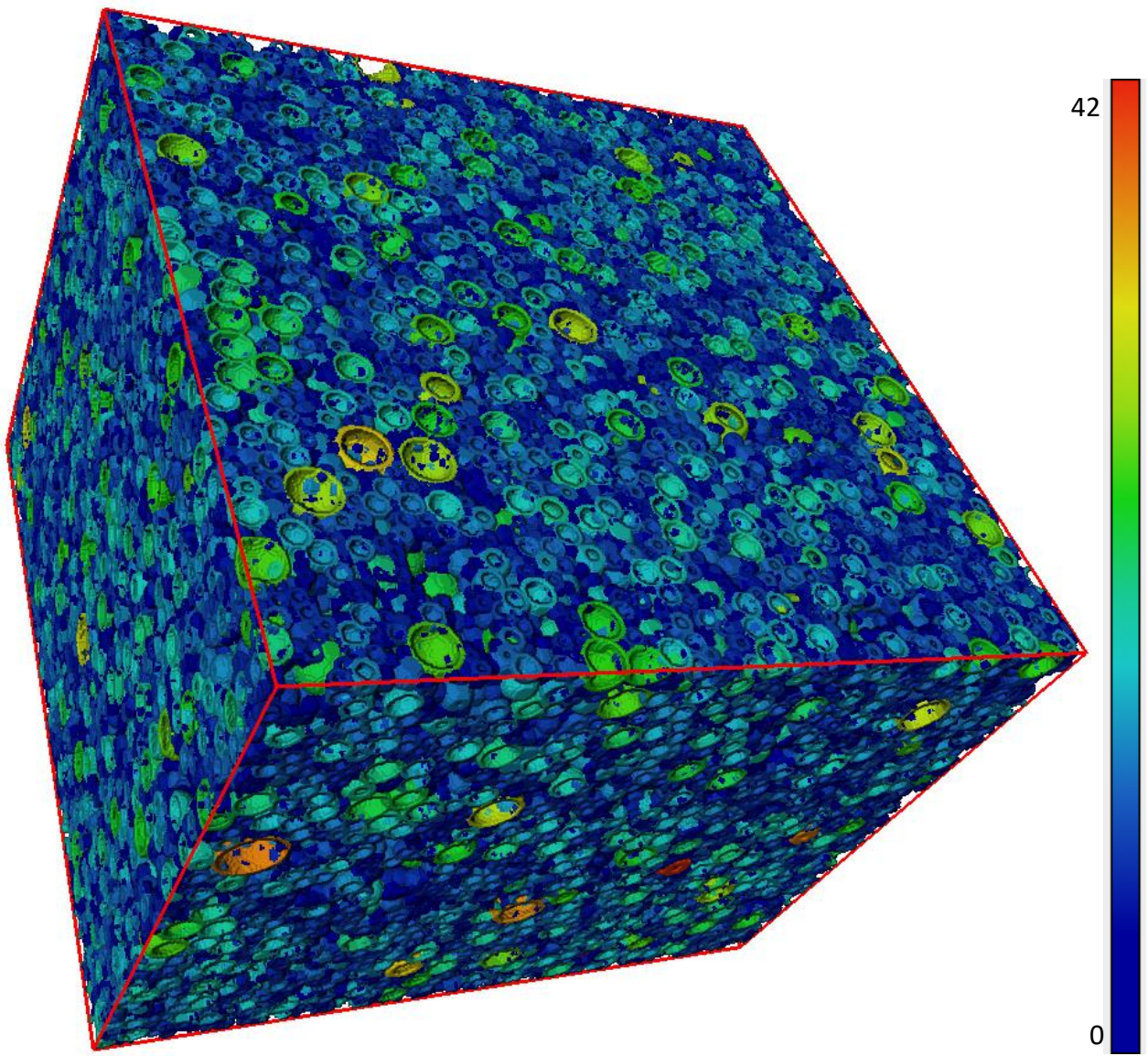


Figure 116 : Résultat de la détection de sphères sur l'image de la Figure 113. Les sphères bleues sont de rayon faible, les rouges de rayon élevé.

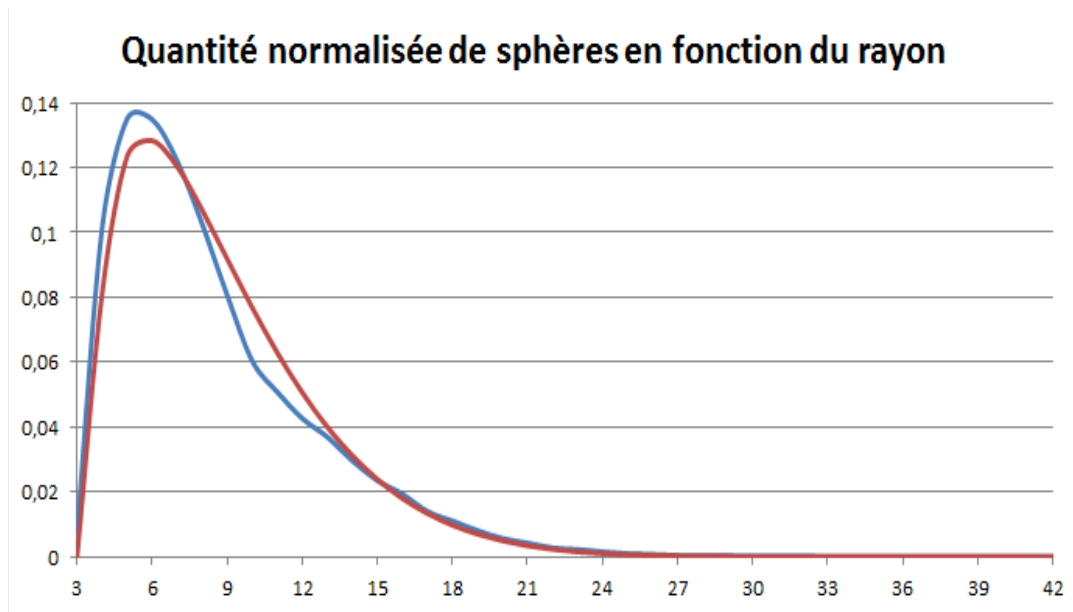


Figure 117 : Graphique représentant le nombre de sphères détectées en fonction du rayon (courbe bleue) et la densité d'une loi bêta de paramètres $\alpha = 1,5$, $\beta = 10$ (calculée par minimisation quadratique).

Concernant la différence entre les images de départ et les résultats de détection, la Figure 118 présente une partie d'image de mousse de polyuréthane à gauche et le résultat de la détection à droite. En dessous, l'image représente ce qui, sur l'image de départ, appartient à une structure détectée par la méthode (en jaune), et ce qui reste de l'image après détection (en bleu). Comme on le voit, sur les mousses de polyuréthane, il est possible de détecter de manière raisonnable l'ensemble des sphères présentes.

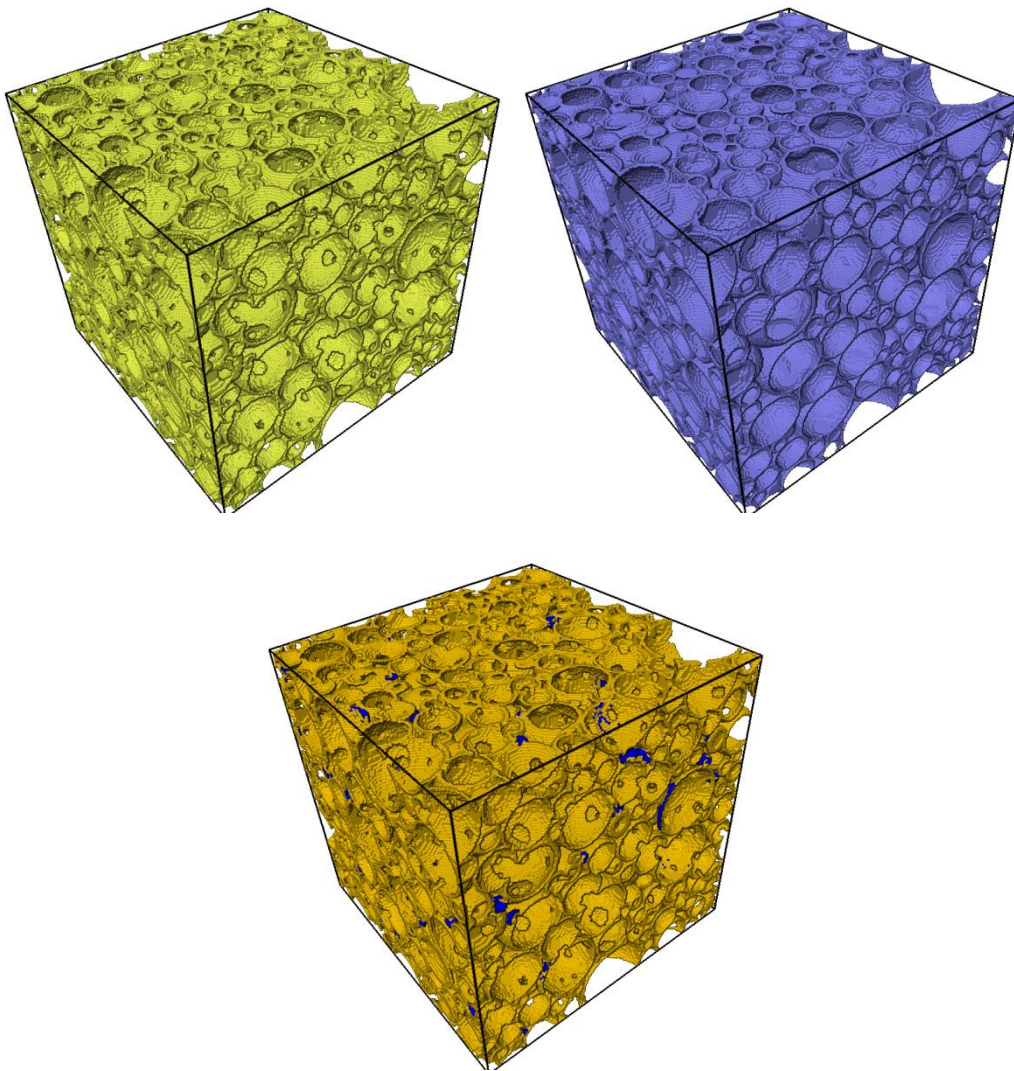


Figure 118 : En haut à gauche, image de taille 300^3 d'une mousse de polyuréthane. À droite, résultat de la détection de sphères sur cette image. En bas, « différence » entre les deux : le jaune représente les voxels de l'image de départ qui appartiennent à une structure détectée à la fin de l'algorithme, le bleu ceux qui n'y appartiennent pas. Il semble que seul du bruit ne soit pas détecté

4.3. Mousses de silice

Sur les images de mousses de silice présentées dans la partie 3, le but de l'étude est aussi de retrouver l'ensemble des sphères afin de connaître leur répartition dans le matériau, mais cette fois-ci pour effectuer des simulations thermiques. De plus, on cherche à vérifier si le procédé de fabrication de ces mousses engendre bien des structures sphériques. Les résultats désirés sont donc semblables à ceux obtenus sur les mousses de polyuréthane, mais la réalisation est cette fois plus complexe. En effet, la qualité des images est inférieure, et l'un des problèmes du procédé de fabrication étant justement la non-sphéricité des structures générées, il faut s'attendre à des difficultés plus fortes pour la détection.

Les images de départ étant fortement bruitées, on ne se contentera pas cette fois d'un seuillage simple, mais on réutilisera l'algorithme des champs de Markov présenté dans la partie 2.4.1. La Figure 119 présente une image tomographique du matériau de départ, de taille 700^3 voxels. La taille de celle-ci ne permet pas de voir apparaître le bruit, mais sur la Figure 120, correspondant à une coupe de l'image, le bruit est nettement plus visible.

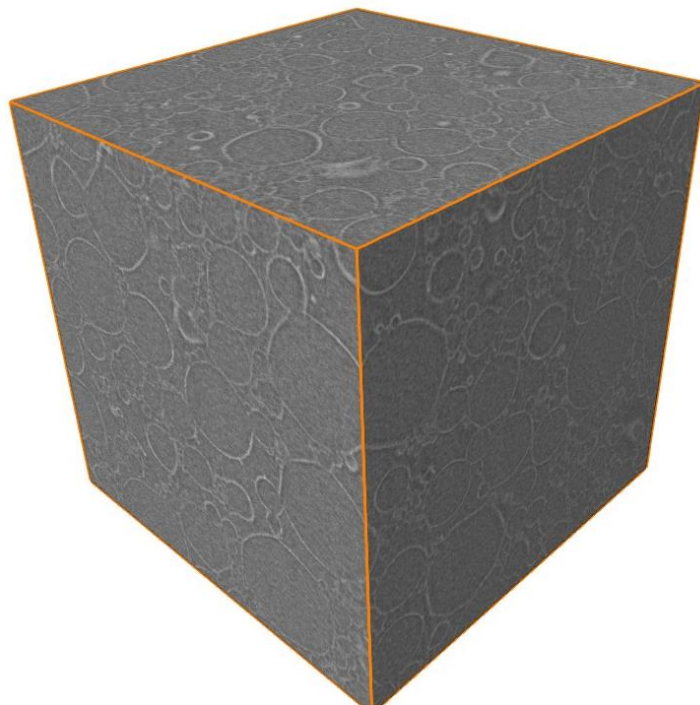


Figure 119 : Image de la mousse de silice de taille 700^3 voxels.

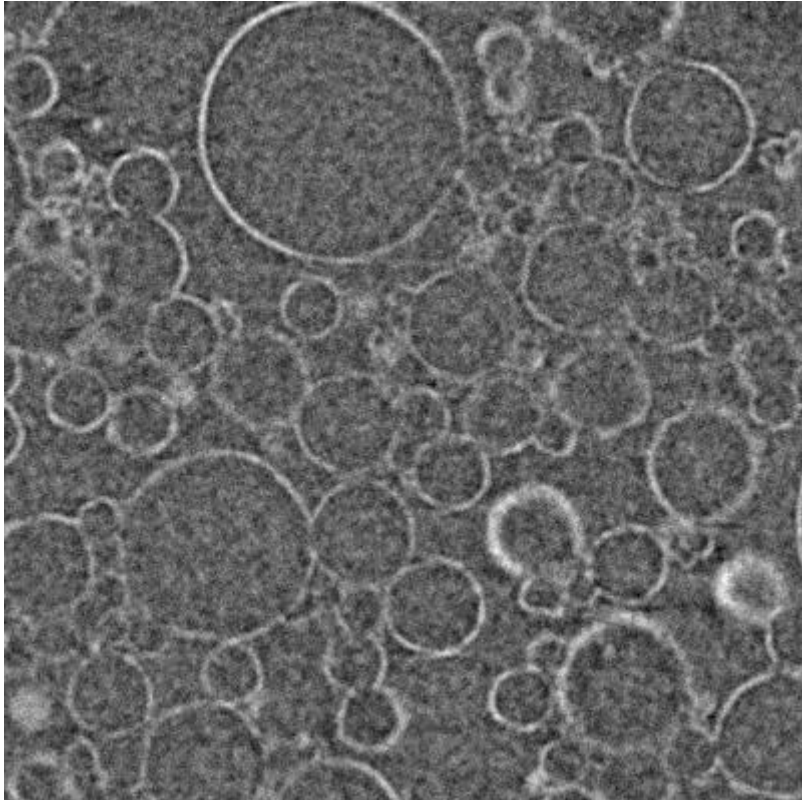


Figure 120 : Coupe de taille 700^2 pixels de l'image de la Figure 119. On observe facilement le bruit présent sur l'image.

Des résultats de la segmentation sont présentés sur les Figure 121 et Figure 122. On voit sur la première que malgré la segmentation avec un terme local, l'image utilisée reste constellée de points de bruit. Dans ce cas, ce bruit résiduel est dû au choix du paramètre β , que l'on a pris de manière à obtenir le plus possible d'informations sur les structures sphériques, quitte à conserver du bruit. L'application d'un algorithme de calcul des régions connexes et de suppression des micro-régions permet de toute manière de supprimer le bruit résiduel dû au choix de segmentation (Figure 122).

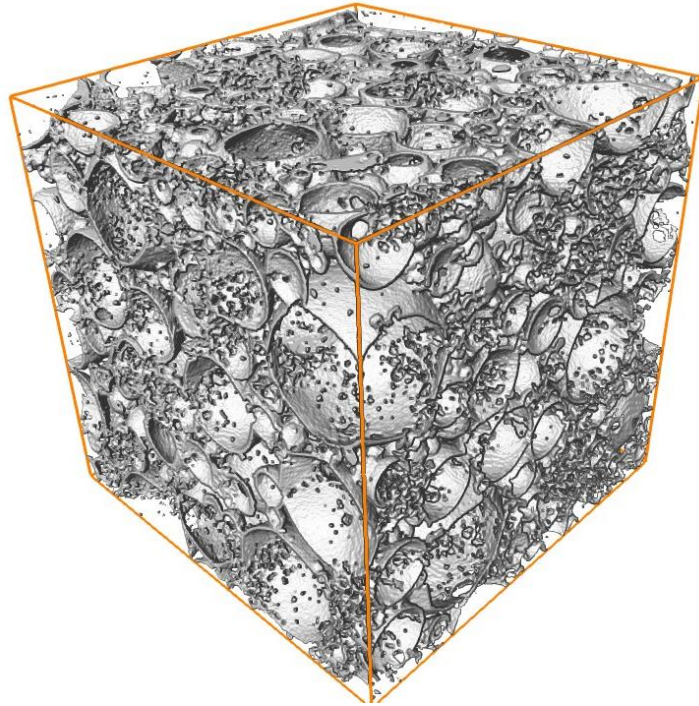


Figure 121 : Résultat de la segmentation par champs de Markov de l'image de la Figure 119. On constate la présence d'un bruit toujours important.

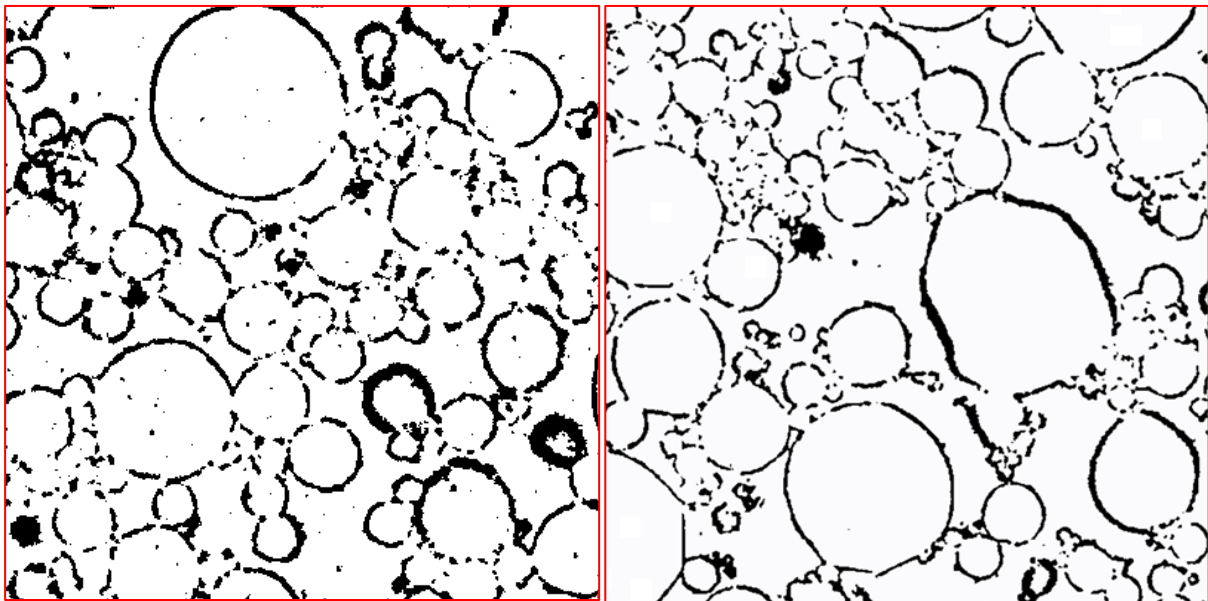


Figure 122 : À gauche, résultat de la segmentation sur la coupe de la Figure 120. À droite, résultat sur une autre coupe après l'algorithme d'élimination des micro-régions. L'intérieur des structures est maintenant net.

À partir de ces images segmentées, il faut détecter les structures sphériques qui y sont présentes. Comme on peut le voir sur l'image de droite de la Figure 122, la sphéricité de certaines structures est toute relative. Comme proposé à la partie 3, on applique d'abord aux images une détection de cercles par transformée de Hough circulaire pour les enrichir (Figure 123).

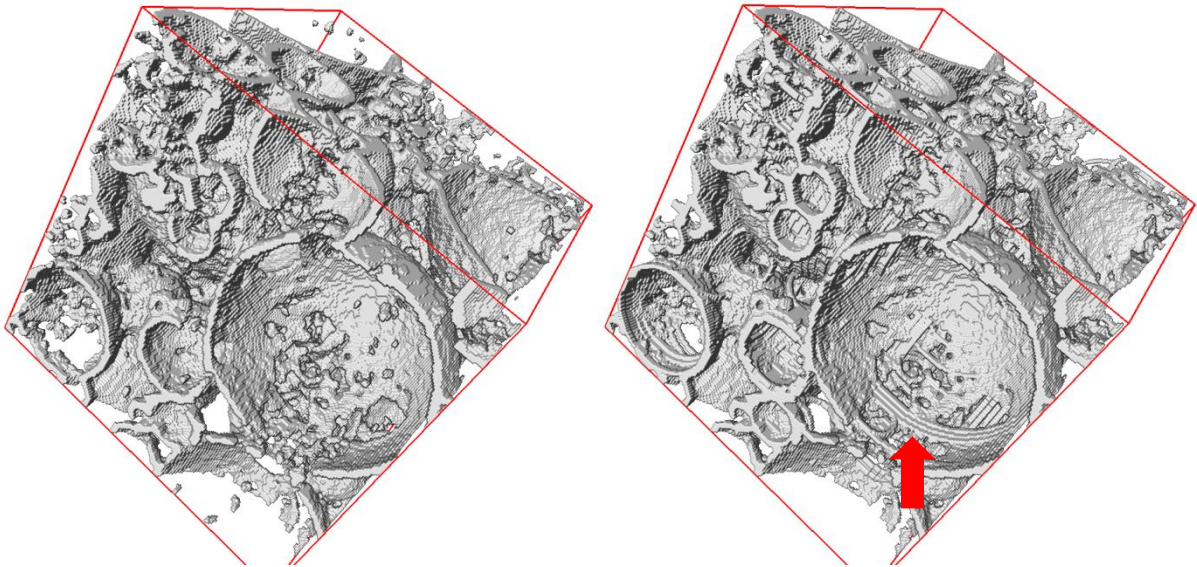


Figure 123 : À gauche, image segmentée de taille 200^3 . À droite, résultat de l'enrichissement par transformée de Hough 2D. Un des endroits où l'image est enrichie est pointé par la flèche rouge.

Suite à cela, on cherche à détecter les sphères ou au moins leurs centres. Les trois méthodes possibles, carte de distance, transformée de Hough et tenseur de structure sont illustrées sur les Figure 125, Figure 126 et Figure 127, issues du traitement de l'image présentée Figure 124 après élimination des micro-régions.

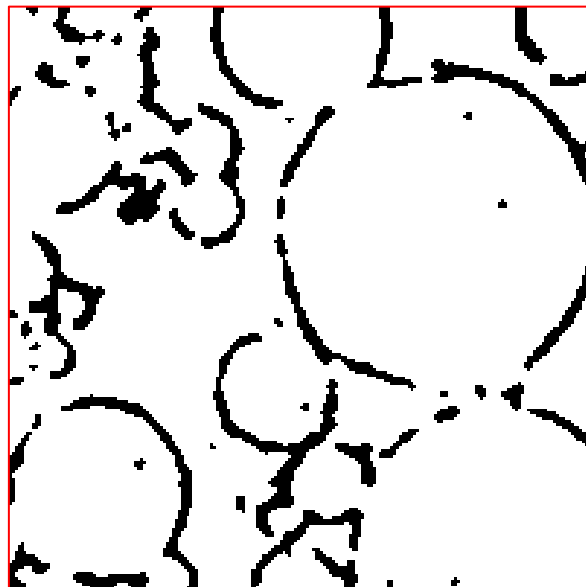


Figure 124 : Coupe de taille 200^2 pixels de la segmentation de l'image originale de la mousse de silice.

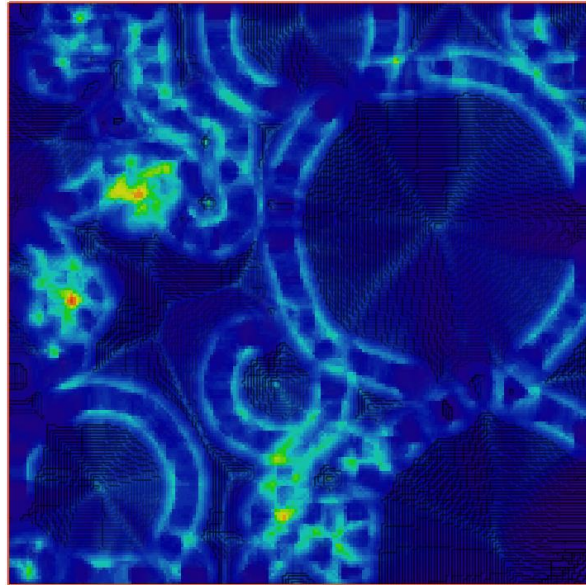


Figure 125 : Vue de face de l'accumulateur de Hough sur la coupe de la Figure 124. Les points localement plus chauds, qui sont aussi sommets de cônes, correspondent à des centres de sphères.

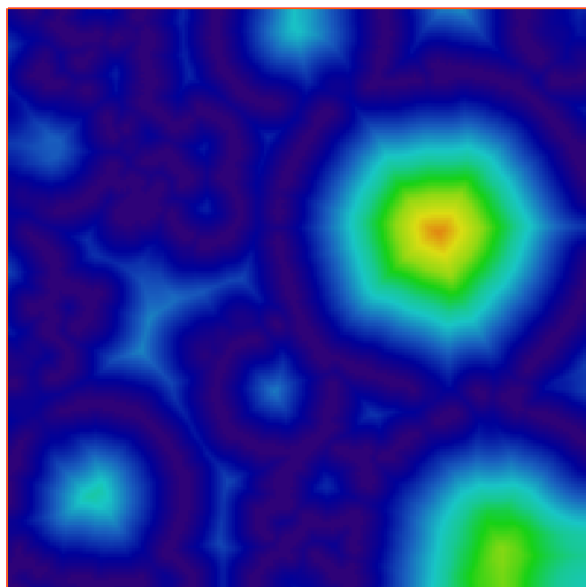


Figure 126 : Vue de la coupe de la carte de distance associée à l'image de la Figure 124, après suppression des micro-régions. Les points chauds correspondent à des centres.

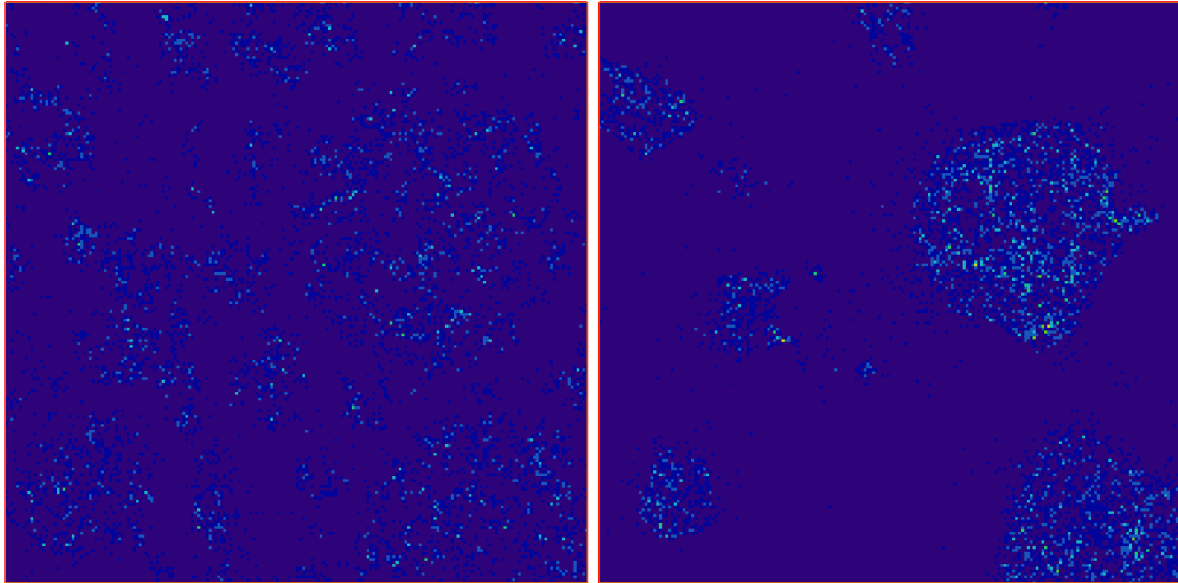


Figure 127 : À gauche, coupe de l'accumulateur du tenseur de structure avec un voisinage de taille 1. À droite, même coupe, mais pour un voisinage de taille 8.

Suite à l'algorithme de détection des centres, on peut procéder à la recherche et au tracé. Finalement, avec la détection par carte de distance, on obtient des résultats présentés sur la Figure 128. On observe bien que certaines structures sont encore absentes.

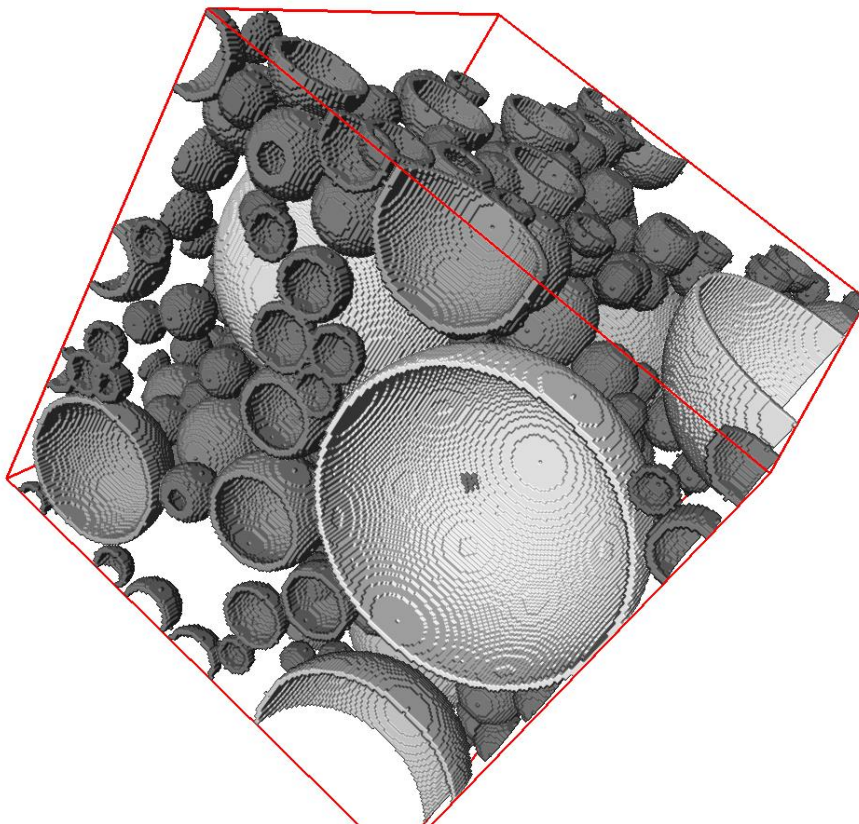


Figure 128 : Résultat de la détection de sphères pour l'image présentée sur la Figure 123, avec une tolérance de 0,8.

On applique une méthode de tolérance adaptative en fonction du rayon (il est à noter que celle-ci était déjà fixée à 0,8 pour le résultat de la Figure 128, c'est-à-dire que l'on accepte une sphère si plus de 20 % des points de la sphère appartiennent à la structure considérée), « faible » pour les petites sphères et importante pour les grandes. On détecte cette fois-ci bien plus de structures (Figure 129), mais force est de constater que des faux-positifs restent présents.

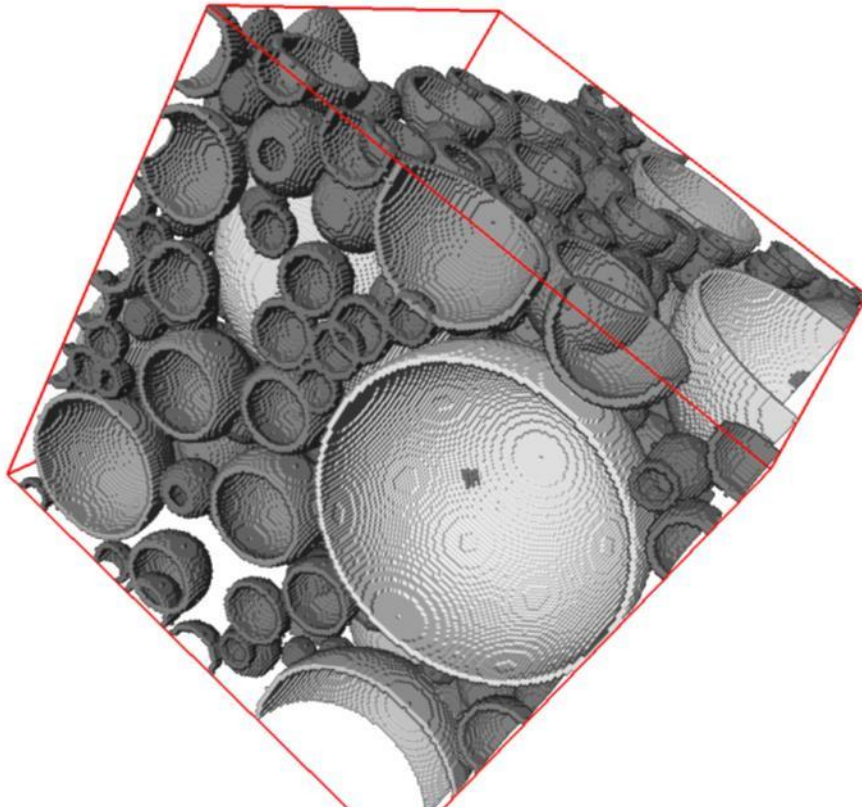


Figure 129 : Résultat de la détection de sphères pour l'image présentée sur la Figure 123, avec une tolérance adaptative.

La Figure 130 propose des superpositions des images initiales avec les résultats de détection dans le cas d'une tolérance à 0,8 et dans le cas d'une tolérance adaptative en fonction du rayon. On voit déjà que des sphères supplémentaires sont ajoutées, et celles-ci sont rarement des faux-positifs.

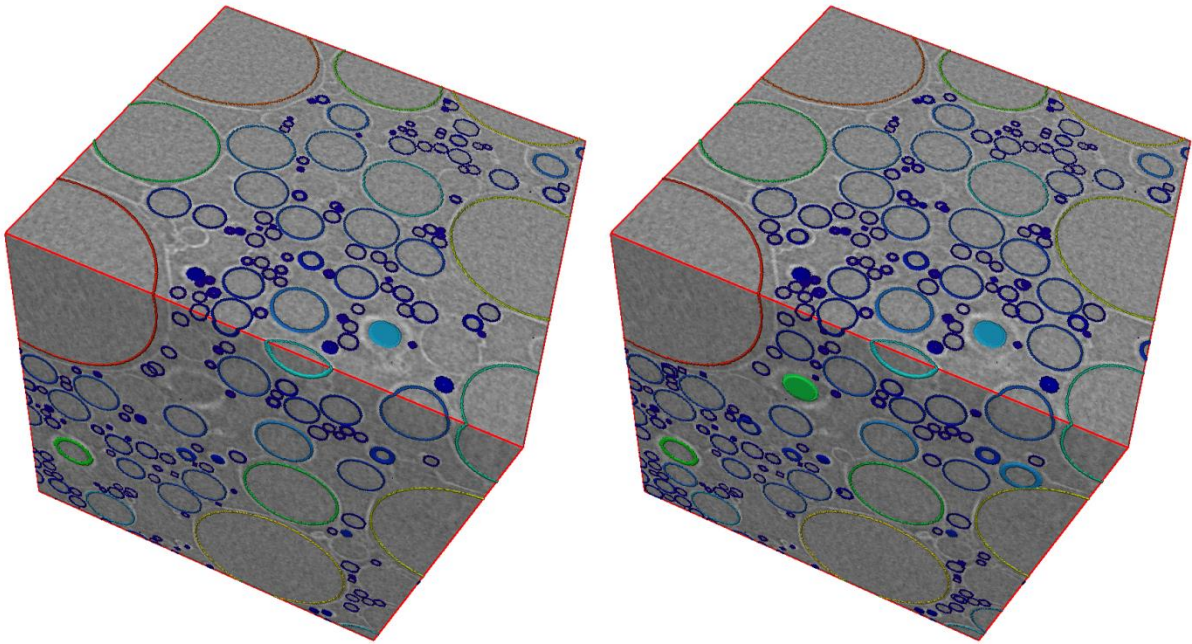


Figure 130 : À gauche, superposition de l'image de départ et de la détection sur une image de 500^3 voxels, et une tolérance de 0,8. À droite, même image, mais avec une tolérance adaptative, de 0,8 au minimum et augmentant ensuite.

Concernant l'algorithme de vérification des points des sphères, celui-ci donne des résultats corrects, mais malheureusement peu exploitables. En effet, on remarque une trop forte corrélation entre la tolérance à laquelle on accepte une sphère et les valeurs de γ_a , c'est-à-dire que dans le cas général, les sphères les plus simples à accepter (que l'on peut trouver avec une tolérance faible) sont celles pour lesquelles la vérification est la plus simple aussi (Figure 131).

Gamma moyen en fonction de la tolérance

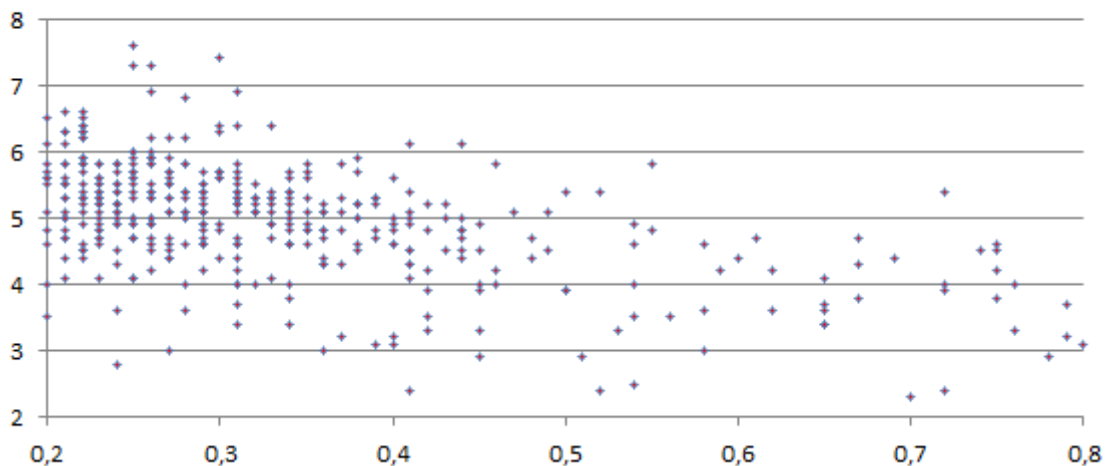


Figure 131 : Image montrant les valeurs de γ_a moyen pour chaque sphère en fonction du nombre de points de la sphère appartenant à la structure. On observe bien une décroissance moyenne de γ_a .

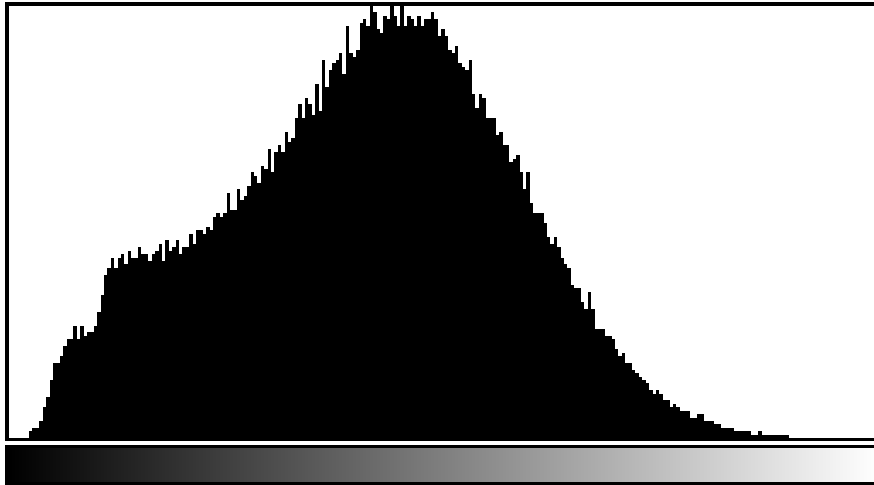


Figure 132 : Histogramme des valeurs de γ_α pour tous les points à vérifier. Il n'y apparaît pas de pics résolus correspondant aux « mauvais » points.

Même si un tel résultat était attendu, cette corrélation est trop forte par rapport aux variations entre « vraies » et « fausses » sphères à tolérance égales pour que la méthode de vérification puisse réellement classer les résultats. On peut observer ceci sur l'histogramme présenté sur la (Figure 132), où l'on n'observe pas de pics résolus entre faux positifs et vrais résultats. La Figure 133 montre le résultat de l'algorithme de vérification sur les points superposé à l'image de départ. Les points rouges correspondent à ceux acceptés (γ_α faible) et les jaunes à ceux refusés. La majorité des points refusés se trouvent sur les faux-positifs, mais certaines fausses sphères ne présentent aussi quasiment que des points acceptés. Finalement, la vérification de sphères, en l'état, ne permet pas d'augmenter de plus de quelques pourcents le nombre de sphères détectées.

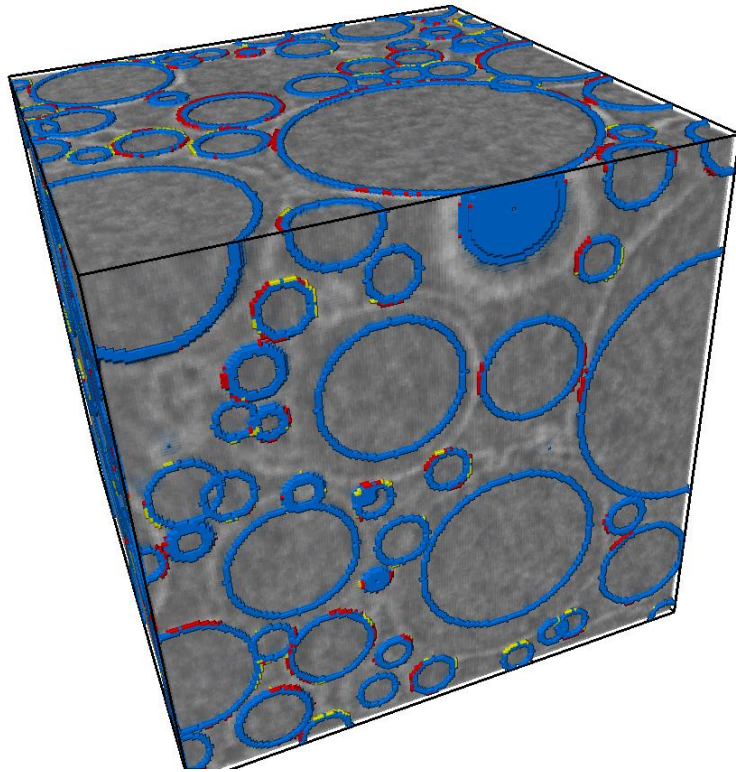


Figure 133 : Superposition d'une image initiale de mousse de silice de taille 200^3 et des sphères retrouvées et vérifiées. Les points rouges correspondent aux points acceptés, les jaunes aux refusés et les bleus au reste des sphères.

5. Conclusions et perspectives

5.1. Perspectives et améliorations

Comme on l'a observé dans la partie 4, les algorithmes développés ici offrent de bons résultats sur les matériaux pour lesquels ils ont été conçus. Il est de plus totalement envisageable, par des modifications mineures, en particulier des changements de paramètres, de penser à les utiliser sur d'autres types d'images correspondant à des matériaux différents.

L'algorithme de détection de sphères, déjà, ne demande que peu de prérequis importants, si ce n'est l'absence de grandes inclusions à l'intérieur des structures, et peut par conséquent fonctionner moyennant quelques pré- et post-traitements sur tout type d'images présentant des structures sphériques. Les perspectives concernant cette méthode concernent donc surtout le développement de multiples phases de traitements s'adaptant aux images. Concernant la détection en elle-même, on peut développer des algorithmes hybrides utilisant les tenseurs de structures, qui permettraient d'obtenir des meilleurs résultats sur les structures de petite taille en validant les centres détectés. En effet, comme on l'a vu sur la Figure 127, le tenseur de structure a des difficultés à détecter précisément et rapidement le centre des sphères (car la fenêtre de moyennage doit être grande), mais peut très rapidement déterminer l'intérieur et l'extérieur des structures (cela se fait avec une fenêtre de taille minimale). Un algorithme hybride permettrait donc possiblement d'éliminer une grande partie des faux-positifs.

D'un autre côté, la recherche d'une sphère dans les méthodes présentées se fait en vérifiant tous les points à distance comprise entre r et $r + 1$. Cette considération pénalise beaucoup plus les grandes sphères, pour lesquelles les petites déformations suffisent à s'éloigner très rapidement de la sphère de référence. Une recherche adaptée en vérifiant les points à distance comprise entre r et $r + \delta(r)$ pourrait permettre d'améliorer la détection, cependant, cette méthode possède aussi des défauts si $\delta(r)$ est trop grand, car les structures voisines peuvent alors artificiellement augmenter le nombre de points valides. Une idée serait alors de ne considérer qu'un seul point par orientation, c'est-à-dire de regarder, en discrétisant les coordonnées sphériques autour du centre, s'il existe au moins

un point dans chaque direction à distance comprise entre r et $r + \delta(r)$ qui appartienne à la structure. La réalisation technique de cette méthode reste quand même sujette à caution, la discrétisation des coordonnées sphériques ne s'accordant pas à la discrétisation cartésienne naturelle des images.

Pour diminuer la tolérance, on a utilisé ici un algorithme de détection de cercles servant à enrichir les images, mais on pourrait aussi imaginer compenser les déformations des sphères dans certains cas. Par exemple, dans le cadre de polyèdres approximant des sphères, un lissage pourrait permettre de retrouver des structures plus simples à détecter par la suite. Différentes méthodes de lissage de surface sont proposées dans la littérature, le plus souvent fonctionnant sur des modèles de triangulation (Shen & Barner, 2004), ce qui ajouterait cependant une difficulté dans la recherche d'une triangulation convenable.

Dans un tout autre axe, on peut envisager un algorithme itératif de détection de sphères, modifiant l'image en fonction des sphères ou des centres détectés, afin de faire converger les structures présentes vers des sphères parfaites. Une méthode de diagramme de Voronoï avec poids pourrait par exemple être étudiée. Chaque centre de sphère serait alors considéré comme un centre de cellule de Voronoï, avec pour poids associé le rayon de la sphère. Suite au pavage, chaque point d'une structure pourrait alors se voir appliquer un déplacement le long de la droite le reliant au centre de sa cellule, tendant à ramener les points vers la sphère associée au centre. Le comportement des fausses sphères dans ce cadre est alors à étudier, mais on peut supposer que la majorité des points les engendrant seront à la frontière de leurs cellules.

La méthode de détection de fibres, pour sa part, repose sur des présupposés bien plus nombreux sur les images. Par conséquent, il est plus complexe de l'utiliser sur des matériaux éloignés des composites carbone-résine présentés en partie 2. La contrainte la plus forte repose dans la rectitude nécessaire des fibres. En effet, celles-ci doivent pouvoir être approchées par des droites sur des distances suffisantes pour que l'algorithme puisse les détecter. Pour autant, d'autres types de matériaux présentent des fibres suffisamment droites pour être étudiées.

En particulier, des premiers tests ont été effectués sur des images tomographiques de feutres présentant des fibres approximativement rectilignes (Figure 134). Cependant, d'autres prérequis ne sont pas vérifiés, ce qui rend la détection assez peu performante en l'état. En effet, des groupes de fibres sans interstices (i.e. totalement fusionnées) peuvent apparaître, ce qui rend la détection des centres compliquée autant pour la méthode utilisant les régions connexes que pour celle fondées sur les cartes de distance. De plus, les fibres étant situées dans des plans orthogonaux à un des axes du repère de l'image, leurs intersections avec les faces qui leurs sont parallèles forment des rectangles infinis et non pas des ellipses, ce qui rend la reconnaissance difficile. Une simple rotation peut faciliter la détection, mais celle-ci n'en reste pas moins incomplète (Figure 135). Des modifications sont à effectuer pour pouvoir traiter ce type de matériau aux fibres bien moins ordonnées et qui s'intersectent beaucoup plus. Le chapitre 2.6.3 explicite les problèmes de la méthode dans le cadre de matériaux proches des feutres testés.

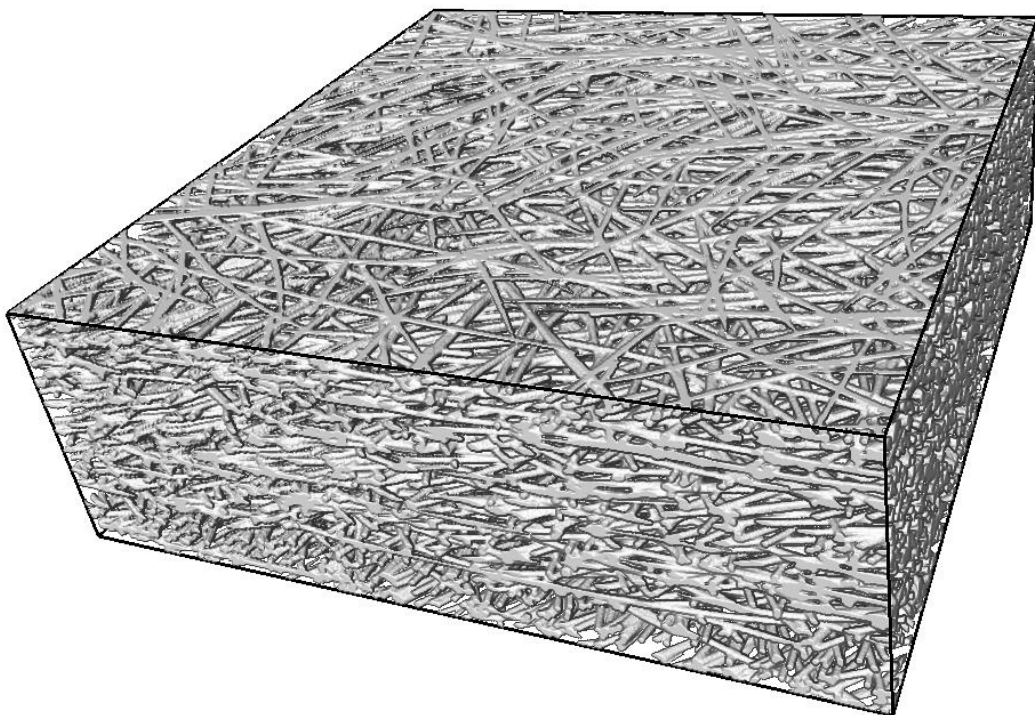


Figure 134 : Image segmentée de feutre de taille $1200 \times 1200 \times 400$.

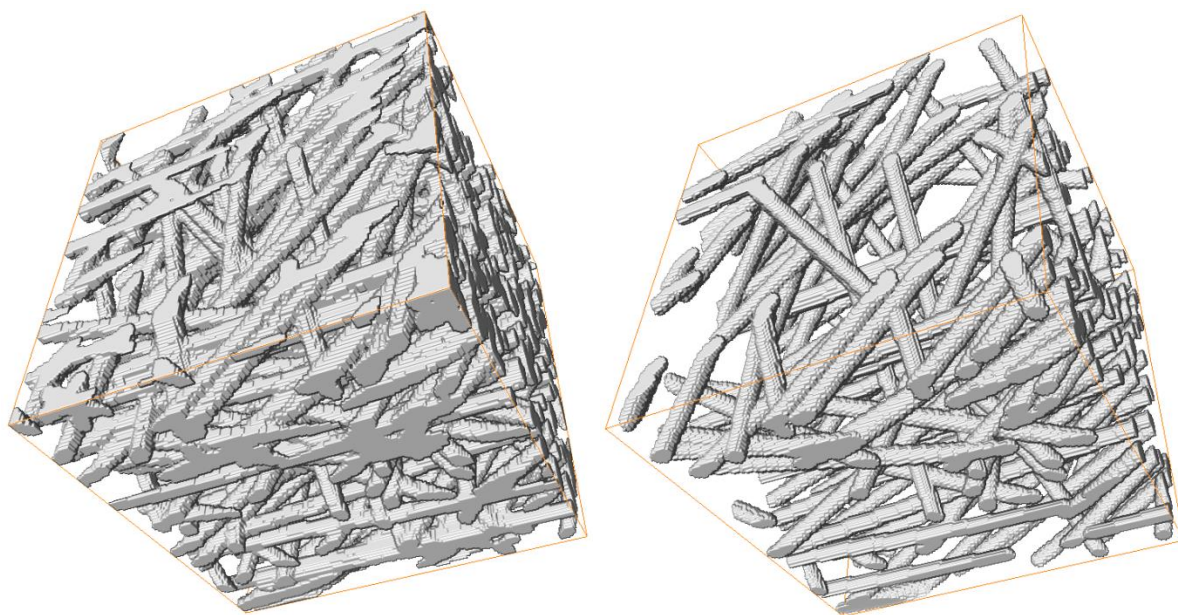


Figure 135 : À gauche, image de feutre de taille 200^3 segmentée et suivie d'une rotation. À droite, résultat de la détection de droite utilisée sur les composites carbone-résine sur une telle image.

Concernant les améliorations que l'on peut apporter à la méthode de détection de fibres, il serait appréciable de réduire sa dépendance à un jeu de bordures particulier. En effet, les centres étant calculés uniquement sur les bords, la méthode possède une forte dépendance à la qualité des bordures choisies, ce qui peut induire de forts biais. Les découpages cartésiens réduisent déjà en partie ce biais en multipliant les surfaces de contact, et donc en minimisant l'importance relative de chaque bord. Cependant, le cadre reste déterministe. Un choix judicieux serait donc de passer à une méthode aléatoire, considérant au hasard un grand nombre de cubes de taille variables et effectuant la détection sur ceux-là, avant de superposer les résultats. De cette manière, on réduit les biais dus à la considération de bords particuliers, et on minimise de la même manière la probabilité de passer à côté de certaines fibres, chaque voxel pouvant être considéré dans plusieurs blocs différents. Pour aller encore plus loin et revenir sur les difficultés présentées sur les images de feutre, on peut de plus imaginer choisir les cubes munis d'une rotation dans l'espace, ce qui permettrait de traiter plus simplement les fibres parallèles orthogonales aux axes du repère de l'image. Sur des images où le but final est de retrouver et suivre les fibres et non des groupes de fibres, des algorithmes d'associations doivent être utilisés. La méthode la plus évidente correspondrait à une simple association avec l'élément le plus proche, mais d'autres méthodes utilisant aussi des propriétés supplémentaires des éléments à associer peuvent être utilisées (Rabin, et al., 2009).

Dans un autre domaine, on peut aussi penser à modifier la manière de déterminer si une fibre est bien présente entre deux centres. Dans le cas où les fibres sont continues mais creuses, une comparaison entre une distance euclidienne simple et une distance obtenue par une méthode de Fast Marching (Sethian, 1996) permettrait de déterminer si le chemin reliant les centres est plus ou moins direct. Dans le cadre de fibres non groupées par orientations proches, les comparaisons de distance peuvent très vite donner des résultats intéressants, moins dans le cadre des composites carbone-résine où les fibres mitoyennes sont aussi d'orientation proches, ce qui engendre des différences de distance non significatives. De plus, la complexité de la méthode augmenterait alors de manière non-négligeable.

Des perspectives portent aussi sur l'étude d'autres types d'images pour étendre les études mécaniques et thermiques sur différents matériaux. En particulier, des traitements permettant de détecter des structures fissurées ont été aussi testés sur certaines images en vue d'une utilisation sur des matériaux de tissage cartésiens avec fissures intersticielles. Une méthode fondée sur un modèle variationnel issu du modèle de Rudin-Osher-Fatemi (Rudin, et al., 1991), et dont la réalisation algorithmique découle d'un modèle de Nesterov (Nesterov, 2005) est utilisée sur des images fissurées dans le but d'en extraire les contours parmi les différentes composantes obtenues (Bergounioux, 2014). Le principe de la méthode étant d'extraire de l'image la composante « texture » (régulière) afin de conserver de l'autre côté le bruit et les contours (sauts de l'image, dont font partie les fissures), comme présenté sur les Figure 136, Figure 137 et Figure 138. L'élimination du bruit se fait ensuite en modélisant les structures du bruit par un arbre, ce qui permet alors de calculer si une composante connexe, en fonction de sa taille, a une probabilité importante d'être une partie du bruit ou d'une fissure.

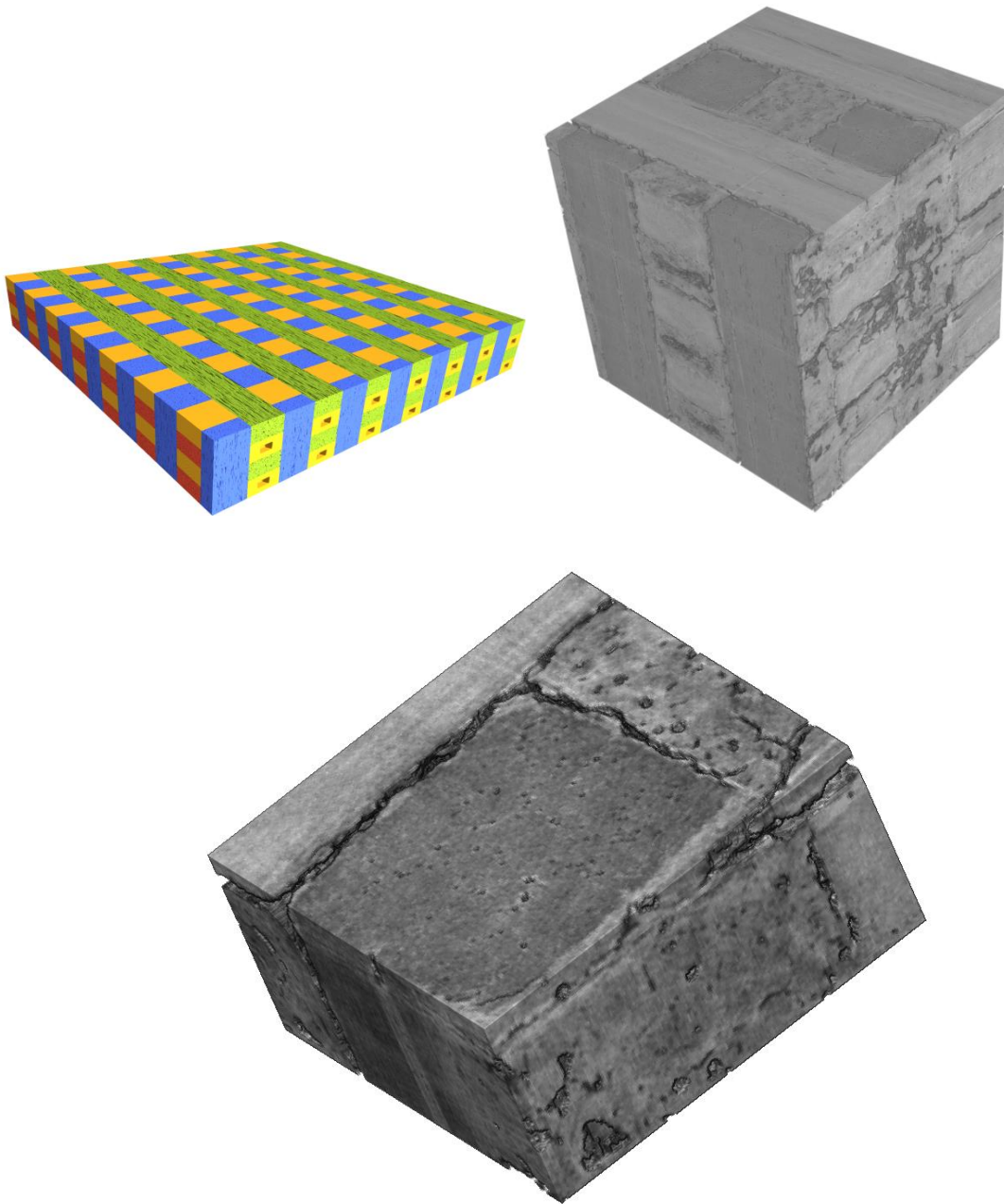


Figure 136 : Images d'un composite à structure de tissage cartésien. En haut à gauche, structure virtuelle du tissage. À droite, image réelle du composite à taille suffisante pour observer le tissage. On recherche à détecter les fissures entre les fils parallélépipédiques.

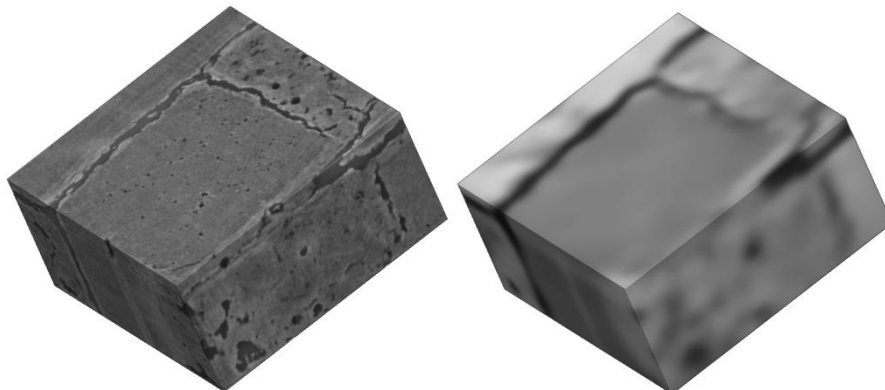


Figure 137 : Vue de la composante contour (à gauche) et de la composante texture (à droite) obtenues avec une méthode variationnelle.

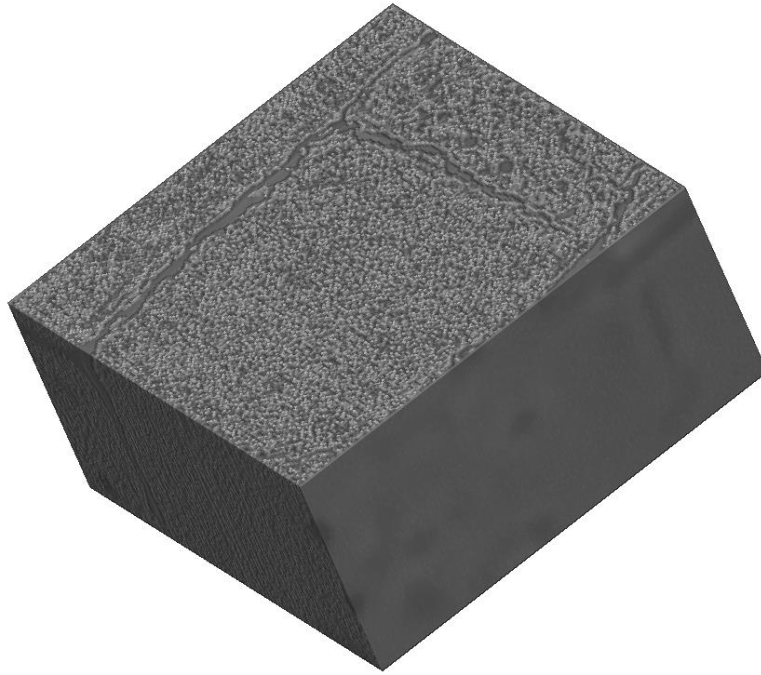


Figure 138 : Composante du bruit obtenue avec une méthode variationnelle. On utilisera cette partie pour extraire les fissures.

5.2. Conclusions

Les travaux présentés dans ce document traitent de la détection et reconnaissance de formes géométriques sur des images 3D de matériaux composites, dans un but ultérieur de modélisation thermique et/ou mécanique. Dans un tel cadre, les images présentent le plus souvent des tailles suffisamment importantes pour rendre difficile l'utilisation de méthodes courantes dans un temps raisonnable. Les méthodes proposées consistent donc à utiliser de manière astucieuse les propriétés des images, et les avantages qui en découlent, afin de proposer des méthodes rapides et adaptées.

Le choix de s'éloigner des techniques analytiques ou probabilistes éprouvées mais coûteuses est donc le fondement du travail présenté ici, qui propose des méthodes et des algorithmes à visée beaucoup moins universelle, mais permettant par là-même, en s'attachant le plus possible au contexte de l'étude, d'atteindre des très bonnes complexités algorithmiques. Les méthodes proposées dérivent directement des propriétés géométriques des objets observés, ce qui permet de créer des algorithmes adaptés, robustes et rapides.

Sur des images données de matériaux composites carbone/résine, un algorithme d'association de centres de fibres a été proposé, suivi d'une classification des fibres dans le cas où celles-ci forment des groupes spatialement cohérents. Le protocole construit est fondé sur l'utilisation de nombreux algorithmes élémentaires, permettant d'obtenir le résultat voulu. Le protocole en lui-même est propre à l'image étudiée, cependant, différentes suites d'algorithmes élémentaires donnent des résultats différents, et peuvent par conséquent traiter des images différentes. Plus qu'une unique méthode permettant de traiter un type d'image particulier (les composites carbone-résine), le travail effectué fournit une « boîte à outils » permettant de construire des algorithmes géométriques adaptés à différents matériaux, ainsi qu'un protocole explicitant la réalisation et l'exploitation d'un algorithme détaillé. Moins généraliste que les méthodes analytiques ou probabilistes, le travail proposé ici ne permet pas moins de traiter différents types d'images, qu'elles diffèrent par le matériau représenté, la résolution ou la méthode d'acquisition.

Sur des images de mousses ou de matériaux à inclusions sphériques, la méthode présentée à une visée bien plus générale, adaptée à la plupart des images de matériaux

possibles. En effet, à partir du moment où les structures sphériques ne contiennent qu'une unique phase unie—ce qui est le cas de toutes les mousses, où les bulles sont constituées de vide, la méthode développée est utilisable. L'adaptation aux différentes variétés d'images possibles se fait cette fois sur une étude de diverses méthodes de pré et de post-traitement, dont une a été étudiée plus particulièrement dans le cadre d'images de mousse de silice.

D'un point de vue applicatif, les méthodes développées et les algorithmes construits permettent bien d'obtenir des résultats utilisables dans le cadre de simulations thermiques ou mécaniques, aussi bien sur les images de mousses que sur celles de matériaux fibreux. La détection de fissure, qui a été engagée dans cette thèse (partie 5.1), permettra d'élargir le spectre des matériaux modélisables pour une simulation ultérieure (en particulier dans le domaine mécanique, très sensible aux fissures). Sur les composites carbone-résine, la détection des fils et leur association à un ensemble homogène permet de considérer l'aspect multi-échelle de tels matériaux et par là-même d'en simplifier la modélisation numérique. De même, la connaissance de la répartition de tailles de sphères sur les matériaux de mousses permet aussi de déterminer un volume élémentaire représentatif et de retrouver finalement un matériau homogène équivalent.

Annexe 1 : Comparaison théorique de la PHT (Kiryati & Bruckstein, 1991) et de la RHT (Xu & Oja, 1993)

Pour montrer les avantages comparatifs des deux méthodes, on va considérer une image du même type que la Figure 139, comportant une droite constituée de N points, et M points de bruit. On notera $T = M + N$. On considère de plus que les triplets de points du bruit ne sont jamais colinéaires.

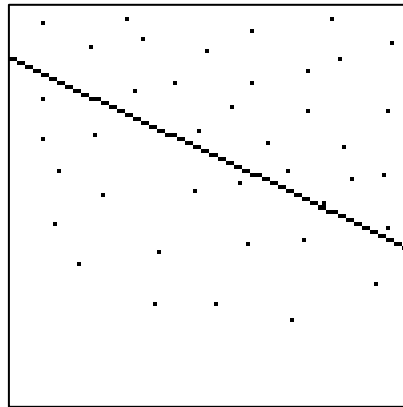


Figure 139 : Type d'image utilisé pour la démonstration des avantages comparatifs de la PHT et de la RHT.

De fait, pour la PHT, 4 points suffisent normalement à caractériser la droite. La probabilité qu'un point x tiré au hasard appartienne à la droite d est de :

$$\mathbb{P}(x \in d) = \frac{N}{T}$$

On notera $p = \frac{N}{T}$ et $q = 1 - p$.

Si l'on considère tous les tirs comme indépendants, alors la probabilité d'avoir au moins 4 tirs sur la droite en effectuant k tirs noté \bar{x} est de :

$$\mathbb{P}(\text{card}(\bar{x} \cap d) \geq 4) = 1 - q^k - \binom{k}{1} q^{k-1} p - \binom{k}{2} q^{k-2} p^2 - \binom{k}{3} q^{k-3} p^3$$

On se donne un ε fixé, et on note $k_\varepsilon^{\text{PHT}}(N, T)$ la valeur de k vérifiant

$$q^k + \binom{k}{1} q^{k-1} p + \binom{k}{2} q^{k-2} p^2 + \binom{k}{3} q^{k-3} p^3 = \varepsilon$$

En effectuant $k_\varepsilon^{PHT}(N, T)$ tirs, on a donc une probabilité $1 - \varepsilon$ de détecter la droite grâce à la PHT. L'accumulateur, dont on notera L^2 les dimensions, est modifié le long d'une sinusoïde pour chaque point. On considèrera en première approximation que le nombre de calculs élémentaires à effectuer est donc de l'ordre de :

$$C^{PHT} \sim L \cdot k_\varepsilon^{PHT}(N, T)$$

Dans le cas de la RHT, on tire des doublets, et de même, il est nécessaire de tirer 4 doublets minimum pour caractériser une droite dans notre cas. La probabilité qu'un doublet w corresponde à la droite d est de :

$$\mathbb{P}(w \in d) = \frac{N(N-1)}{T(T-1)}$$

On notera $r = \frac{N(N-1)}{T(T-1)}$ et $s = 1 - r$.

Si l'on considère tous les tirs comme indépendants, alors la probabilité d'avoir au moins 4 doublets sur la droite en effectuant k tirs de doublets noté \bar{w} est de :

$$\mathbb{P}(\text{card}(\bar{w} \cap d) \geq 4) = 1 - s^k - \binom{k}{1} s^{k-1} r - \binom{k}{2} s^{k-2} r^2 - \binom{k}{3} s^{k-3} r^3$$

En reprenant la même valeur de ε fixée que pour la PHT, on définit $k_\varepsilon^{RHT}(N, T)$ la valeur de k vérifiant

$$s^k + \binom{k}{1} s^{k-1} r + \binom{k}{2} s^{k-2} r^2 + \binom{k}{3} s^{k-3} r^3 = \varepsilon$$

Pour la RHT, un seul point de l'accumulateur est modifié pour chaque doublet choisi. En première approximation, on peut donc estimer la quantité de calculs élémentaires de la RHT par :

$$C^{RHT} \sim k_\varepsilon^{RHT}(N, T)$$

Afin d'étudier le comportement de la PHT et de la RHT, il faut donc comparer les comportements des deux fonctions $(N, T) \mapsto L \cdot k_\varepsilon^{PHT}(N, T)$ et $(N, T) \mapsto k_\varepsilon^{RHT}(N, T)$. On constatera déjà que :

$$k_\varepsilon^{RHT}(N, T) = k_\varepsilon^{PHT}(N(N-1), T(T-1))$$

Pour observer le comportement de k , on va étudier les lignes de niveau de la fonction \mathcal{f} suivante (Figure 140) définie sur $[0; 1] \times \mathbb{R}_+$:

$$(x, k) \mapsto (1-x)^{k-3} \left((1-x)^3 + k(1-x)^2x + \frac{k(k-1)}{2}(1-x)x^2 + \frac{k(k-1)(k-2)}{6}x^3 \right)$$

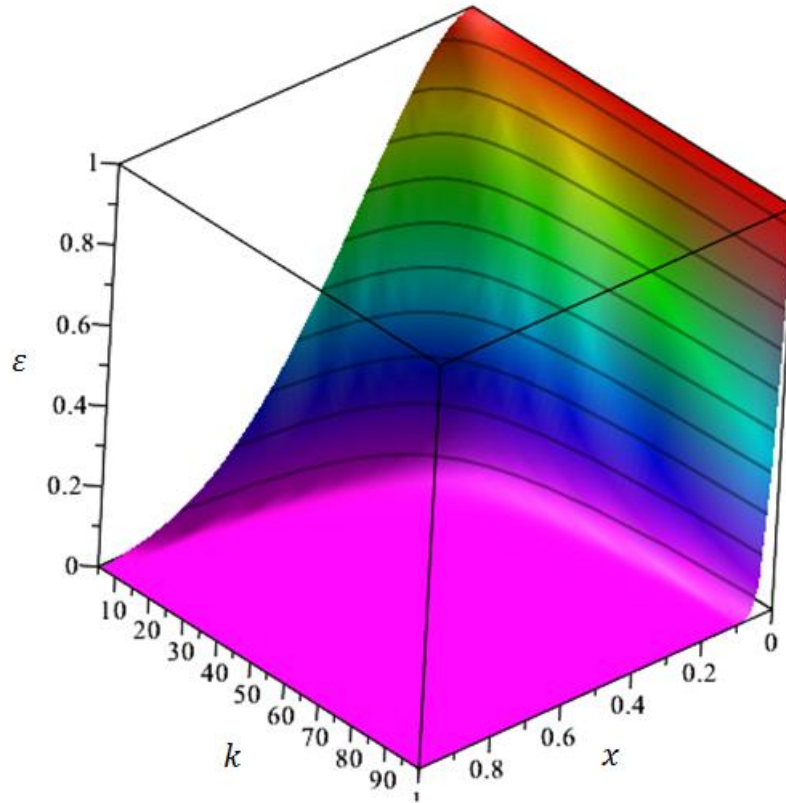


Figure 140 : Graphe de la fonction \mathcal{f} sur $[0; 1] \times [5; 100]$. Des lignes de niveau de la fonction sont représentées en noir.

La ligne de niveau définie par $l(\mathcal{f}, \varepsilon) = \{(x, k) \text{ tel que } \mathcal{f}(x, k) = \varepsilon\}$ permet alors d'étudier le comportement de k (Figure 141), en effet, on a alors les propriétés suivantes :

- $k_\varepsilon^{PHT}(N, T)$ est l'ordonnée du point de la ligne de niveau $l(\mathcal{f}, \varepsilon)$ d'abscisse $x = \frac{N}{T}$.
En effet, $\mathcal{f}\left(\frac{N}{T}, k_\varepsilon^{PHT}(N, T)\right) = \varepsilon$ par définition.
- De même, $k_\varepsilon^{RHT}(N, T)$ est l'ordonnée du point de la ligne de niveau $l(\mathcal{f}, \varepsilon)$ d'abscisse $x = \frac{N(N-1)}{T(T-1)}$.

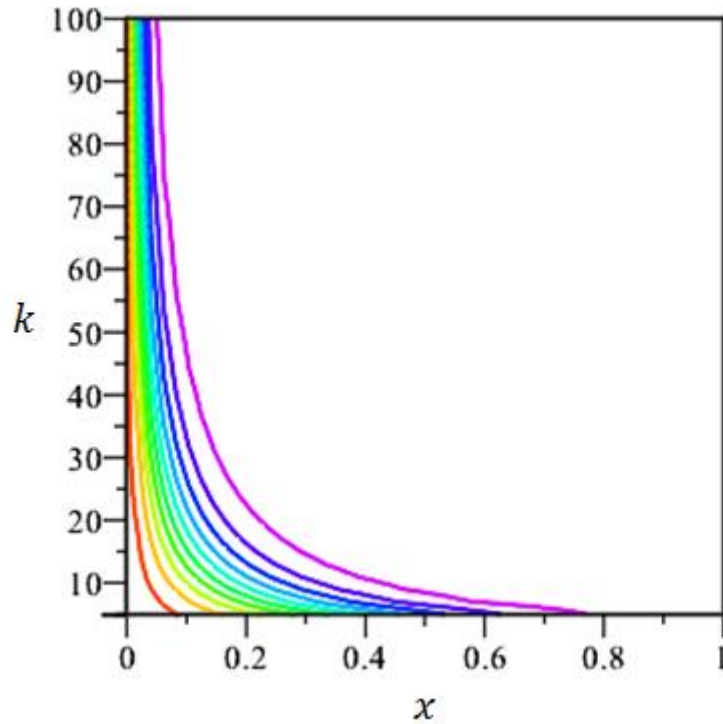


Figure 141 : Graphe des lignes de niveau $l(x, \varepsilon)$, le violet correspondant aux petites valeurs de ε vers les grandes valeurs en rouge. On constatera le comportement décroissant en x des lignes de niveau.

Dans notre cas, on considèrera le nombre de points de la droite N fixé à N_0 , et on fera varier T sur $[N_0; +\infty[$. En effet, on peut considérer que pour tout N , on a

$$k_{\varepsilon}^{PHT}(N, T) = k_{\varepsilon}^{PHT}\left(N_0, \frac{TN_0}{N}\right) \text{ et } k_{\varepsilon}^{RHT}(N, T) = k_{\varepsilon}^{RHT}\left(N_0, \frac{TN_0}{N}\right)$$

(dans le cas de la PHT, il est évident que $\frac{N_0}{\frac{TN_0}{N}} = \frac{N}{T}$, et dans le cas de la RHT, on peut admettre,

pour un nombre de point suffisamment élevé que $\frac{N_0-1}{\frac{TN_0-1}{N}} = \frac{N-1}{T-1}$).

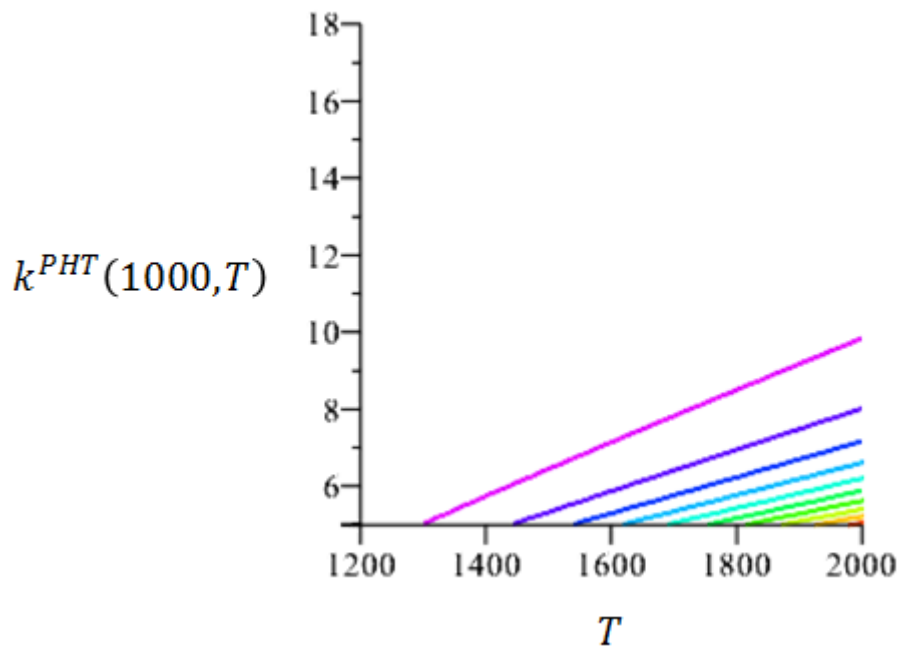


Figure 142 : Evolution de $k^{PHT}(1000, T)$ pour $T \in [1200; 2000]$, c'est-à-dire pour un rapport signal sur bruit compris entre 5 et 1. Les différentes courbes correspondent à différentes valeurs de ε . En particulier, la courbe violette correspond à $\varepsilon = 0,05$, c'est-à-dire à la valeur de k^{PHT} nécessaire pour avoir 95 % de chances de détecter la droite.

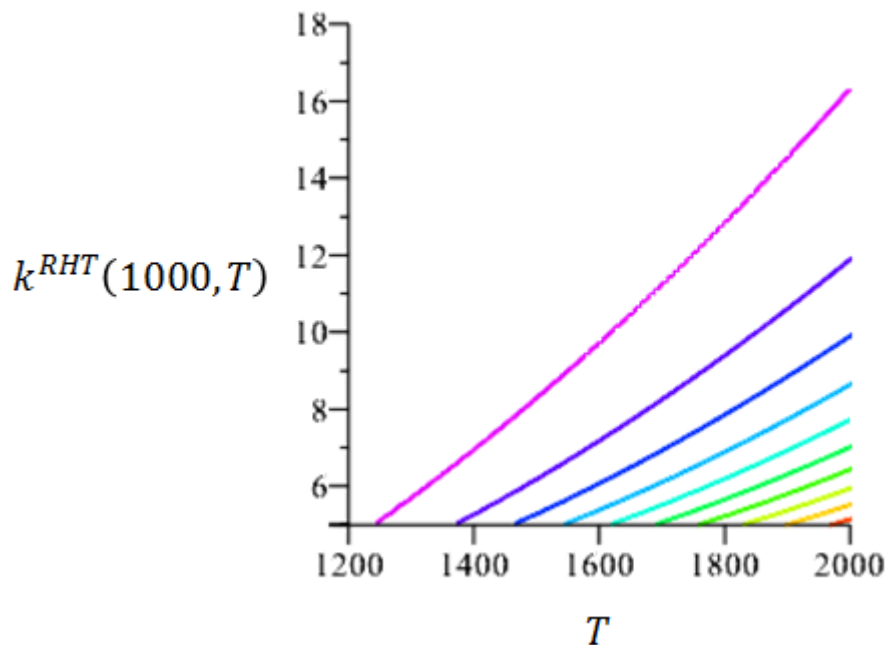


Figure 143 : Evolution de $k^{RHT}(1000, T)$ pour $T \in [1200; 2000]$, c'est-à-dire pour un rapport signal sur bruit compris entre 5 et 1. Les différentes courbes correspondent à différentes valeurs de ε . En particulier, la courbe violette correspond à $\varepsilon = 0,05$, c'est-à-dire à la valeur de k^{RHT} nécessaire pour avoir 95 % de chances de détecter la droite.

Les Figure 142 et Figure 143 ci-dessus présentent l'évolution de $k_{\varepsilon}^{PHT}(1000, T)$ et $k_{\varepsilon}^{RHT}(1000, T)$ pour $T \leq 2000$, i.e. pour un rapport signal sur bruit supérieur à 1. Les courbes correspondent aux lignes de niveau respectives des fonctions :

$$(T, k) \mapsto \# \left(\frac{1000}{T}, k \right) \text{ et } (T, k) \mapsto \# \left(\frac{1000 * 999}{T(T-1)}, k \right)$$

On observe que le comportement asymptotique de $k_{\varepsilon}^{PHT}(1000, T)$ et $k_{\varepsilon}^{RHT}(1000, T)$ ne sont pas les mêmes, et on en déduit que $k_{\varepsilon}^{RHT} - k_{\varepsilon}^{PHT} \rightarrow +\infty$. Ainsi, à l'infini, c'est-à-dire quand le rapport signal sur bruit tend vers 0,

$$\frac{C^{RHT}}{C^{PHT}} \rightarrow +\infty$$

Donc la RHT est moins intéressante que la PHT si le bruit est important.

Dans le cas où $T = N$, alors on a $k_{\varepsilon}^{RHT} = k_{\varepsilon}^{PHT}$, donc $C^{RHT} \sim \frac{C^{PHT}}{L}$. On revient donc à la conclusion de la plus grande rapidité de la RHT dans le cas d'images simples et non-bruitées.

Annexe 2 : Estimation de la connexité dans un fil carbone/résine

Pour calculer la complexité de l'algorithme de recherche de droites, il est nécessaire de connaître le nombre de centres qui peuvent être associés entre eux. Un des centres d'une fibre peut être associé par l'algorithme à l'ensemble des centres de l'image reliés de manière connexe au centre considéré :

Soit $I: D(I) \subset \mathbb{N}^3 \rightarrow \{0; 1\}$ l'image binaire de base considérée. On note $D(I_+)$ l'image réciproque de 1 par I .

De même, on note $\partial D(I)$ la frontière de $D(I)$ et $\partial D(I_+) = \partial D(I) \cap D(I_+)$, $\partial D(I_-) = \partial D(I) \setminus D(I_+)$. On munira $\partial D(I_+)$ de la distance euclidienne sur \mathbb{N}^3 et on considèrera la fonction $f: \partial D(I_+) \rightarrow \mathbb{R}_+$ définie par

$$f(x) = \min_{y \in \partial D(I_-)} \|y - x\|$$

On notera $(c_i)_{i < N}$ les points de $\partial D(I_+)$ réalisant les maximums locaux de f . On notera ensuite $(Q_j)_{j < M}$ les classes d'équivalences de $(c_i)_{i < N}$ pour la relation « est relié de manière connexe dans $D(I)$ à », i.e. $c_i, c_{i'}$ appartiennent au même Q_j si et seulement s'ils sont reliés de manière connexe dans $D(I)$.

On recherche ici une estimation moyenne de $\text{card}(Q_j)$, qui nous donnera ensuite un $\Theta(\cdot)$ de la complexité de la recherche de droites.

Commençons par estimer la probabilité de contact d'une fibre proche avec des fibres voisines. On se placera ici dans le plan, et on considèrera le cas d'une fibre encadrée par une fibre supérieure et une inférieure. Les fibres seront considérées comme des suites de points de coordonnées $(n, Y_n)_{n \in \mathbb{N}}$, où les Y_n sont des variables aléatoires à valeurs dans \mathbb{N} . Par abus de notation, on identifiera la suite $(Y_n)_{n \in \mathbb{N}}$ à la fibre y .

On se donne donc trois fibres y, y^+, y^- vérifiant :

$$\begin{cases} \forall n, |Y_n^+ - Y_n^-| = 2a \\ |Y_0 - Y_0^-| = |Y_0 - Y_0^+| = a \end{cases}$$

On supposera de plus que la suite de variables aléatoires (D_n) définie par

$$\begin{cases} m_E(0) = 0 \\ m_E(2a) = 0 \\ 2m_E(x) - m_E(x+1) - m_E(x-1) = \frac{1}{\varepsilon} \text{ pour } x \in [1:2a-1] \end{cases}$$

Par symétrie, on peut affirmer que $m_E(a) = m_E(a-1) + \frac{1}{2\varepsilon} = m_E(a+1) + \frac{1}{2\varepsilon}$. On peut alors montrer que l'on a, pour $x \in [0:2a]$,

$$m_E(x) = xm_E(1) - \frac{x(x-1)}{2\varepsilon}$$

et par symétrie :

$$m_E(2a-1) = (2a-1)m_E(1) - \frac{(2a-1)(2a-2)}{2\varepsilon} = m_E(1)$$

i.e.

$$m_E(1) = \frac{2a-1}{2\varepsilon}$$

Finalement, on a donc

$$m_E(x) = \frac{x(2a-x)}{2\varepsilon}$$

Dans le cas qui nous intéresse, on suppose que la fibre débute à égale distance de ses deux voisines, et on a donc comme espérance du temps d'arrêt T_E :

$$\mathbb{E}(T_E) = \frac{a^2}{2\varepsilon}$$

On va maintenant considérer l'ensemble $E' = \{0\}$, et on va rechercher l'espérance du temps d'arrêt $T_{E'}$, qui représente le temps moyen auquel la fibre, après avoir touché une de ses voisines, va toucher l'autre voisine. De la même manière que précédemment, on peut définir $m_{E'}$, qui est solution de l'équation affine :

$$\begin{cases} m = 1 + Pm & \text{sur } [1:2a] \\ m = 0 & \text{sur } E' \end{cases}$$

ce qui donne cette fois le système suivant :

$$\begin{cases} m_{E'}(0) = 0 \\ m_{E'}(2a) - m_{E'}(2a - 1) = \frac{1}{\varepsilon} \\ 2m_{E'}(x) - m_{E'}(x + 1) - m_{E'}(x - 1) = \frac{1}{\varepsilon} \text{ pour } x \in [1: 2a - 1] \end{cases}$$

On en revient à l'expression suivante

$$m_{E'}(x) = xm_{E'}(1) - \frac{x(x - 1)}{2\varepsilon}$$

et on sait que l'on a en plus $m_{E'}(2a) - m_{E'}(2a - 1) = \frac{1}{\varepsilon}$, ce qui donne :

$$2am_{E'}(1) - \frac{2a(2a - 1)}{2\varepsilon} = (2a - 1)m_{E'}(1) - \frac{(2a - 1)(2a - 2)}{2\varepsilon} + \frac{1}{\varepsilon}$$

ce qui équivaut à

$$m_{E'}(1) = \frac{2a(2a - 1) - (2a - 1)(2a - 2) + 2}{2\varepsilon} = \frac{2a}{\varepsilon}$$

d'où

$$m_{E'}(x) = \frac{x(4a - x + 1)}{2\varepsilon}$$

Ici, on s'intéresse particulièrement au cas où $x = 2a$, et on a donc :

$$\mathbb{E}(T_{E'}) = \frac{2a^2 + a}{\varepsilon}$$

Grâce à l'additivité des temps d'arrêts, on sait que $T_E + T_{E'}$ est un temps d'arrêt, et par conséquent que son espérance est de

$$\mathbb{E}(T_E + T_{E'}) = \frac{5a^2 + 2a}{2\varepsilon}$$

Maintenant, on va uniquement étudier les événements « la fibre touche pour une voisine sur la distance d » et « la fibre touche pour ses deux voisines sur la distance d », que l'on représentera par deux variables aléatoires X_1, X_2 suivant des lois géométriques de paramètres respectifs p_1 et p_2 . Etant donné que l'espérance des lois géométriques est égale à l'inverse du paramètre p de la loi, on peut, en recoupant avec le travail effectué sur la modélisation en chaîne de Markov ci-dessus, poser :

$$p_1 = \frac{2\varepsilon}{a^2} \text{ et } p_2 = \frac{2\varepsilon}{5a^2 + 2a}$$

On pourra donc poser dans notre modèle que la probabilité d'une fibre de toucher ses deux voisines sur une distance d est de :

$$\mathbb{P}(X_2 = d) = 1 - \left(1 - \frac{2\varepsilon}{5a^2 + 2a}\right)^d$$

et celle de toucher une seule de ses deux voisines sur une distance d est :

$$\mathbb{P}(X_1 = d, X_2 > d) = \left(1 - \frac{2\varepsilon}{5a^2 + 2a}\right)^d - \left(1 - \frac{2\varepsilon}{a^2}\right)^d$$

Maintenant que l'on a modélisé le contact d'une fibre avec ses voisines dans le plan, il est possible de déterminer la taille des groupes de fibres reliés de manière connexes.

Pour le moment, on considèrera dans un repère orthonormé (O, x, y, z) un fil infini sur les deux directions x et y et défini entre les plans d'équation $z = 0$ et $z = L$. Ce fil sera constitué d'une infinité de fibres de rayon r toutes orientées orthogonalement au plan $z = 0$ (donc elles sont de longueur L), et placées régulièrement selon un pavage hexagonal telles que leurs intersections avec le plan $z = 0$ se fassent aux points de coordonnées $\{(uR, v\sqrt{3}R), u, v \in \mathbb{Z}^2\} \cup \left\{\left(u + \frac{1}{2}\right)R, \left(v + \frac{1}{2}\right)\sqrt{3}R\right\}$, avec $R > r$. On notera pour le moment p_L la probabilité qu'à une fibre de toucher une de ses voisines mais de ne pas toucher la fibre opposée et q_L la probabilité de toucher une voisine et la fibre opposée.

On va ensuite estimer la taille moyenne du groupe de fibres reliées de manière connexe via une méthode de Monte-Carlo. Pour cela, nous allons commencer par nous donner un ensemble de fibres F_0 réduit à une fibre, et A_0 l'ensemble des liaisons entre cette fibre et ses voisines. Pour toute liaison quelconque l entre deux fibres voisines, on notera $\delta(l) \in \{0; 1\}$ l'état du contact entre les deux fibres (0 si elles ne se touchent pas et 1 dans le cas contraire). Ensuite, pour tout n , on effectue les étapes suivantes :

- On regarde chaque liaison $l^n(f, g)$ de A_n . Une liaison de A_n relie deux fibres f, g dont au moins une appartient à F_n . Pour chacune de ces deux fibres, il existe une

liaison opposée à l^n , que l'on notera $\text{Sym}_f(l^n)$. Selon l'état de ces liaisons opposées, on va définir deux valeurs p_f et p_g définie par :

- Si $\text{Sym}_f(l^n) \in A_k, k < n$, alors cela veut dire que la fibre symétrique a déjà été testée, et qu'on lui a associé une valeur $\delta(\text{Sym}_f(l^n))$. On a alors deux cas :
 - $\delta(\text{Sym}_f(l^n)) = 0$, et donc on a

$$p_f = \mathbb{P}(\delta(l^n) = 1 | \delta(\text{Sym}_f(l^n)) = 0)$$

ce qui équivaut, via le théorème de Bayes, à

$$p_f = \frac{\mathbb{P}(\delta(l^n) = 1, \delta(\text{Sym}_f(l^n)) = 0)}{\mathbb{P}(\delta(\text{Sym}_f(l^n)) = 0)}$$

Or on a logiquement

$$\mathbb{P}(\delta(l^n) = 1, \delta(\text{Sym}_f(l^n)) = 0) = \frac{p_L}{2}$$

Et

$$\begin{aligned} \mathbb{P}(\delta(\text{Sym}_f(l^n)) = 0) &= \mathbb{P}(\delta(l^n) = 1, \delta(\text{Sym}_f(l^n)) = 0) \\ &+ \mathbb{P}(\delta(l^n) = 0, \delta(\text{Sym}_f(l^n)) = 0) \end{aligned}$$

D'où

$$\mathbb{P}(\delta(\text{Sym}_f(l^n)) = 0) = 1 - \frac{p_L}{2} - q_L$$

Ce qui nous permet finalement d'obtenir

$$p_f = \frac{p_L}{2 - p_L - 2q_L}$$

- $\delta(\text{Sym}_f(l^n)) = 1$, et donc on a

$$p_f = \mathbb{P}(\delta(l^n) = 1 | \delta(\text{Sym}_f(l^n)) = 1)$$

Via le théorème de Bayes, on a donc

$$p_f = \frac{\mathbb{P}(\delta(l^n) = 1, \delta(\text{Sym}_f(l^n)) = 1)}{\mathbb{P}(\delta(\text{Sym}_f(l^n)) = 1)}$$

Par définition, on a

$$\mathbb{P}(\delta(l^n) = 1, \delta(\text{Sym}_f(l^n)) = 1) = q_L$$

Et

$$\begin{aligned} \mathbb{P}(\delta(\text{Sym}_f(l^n)) = 1) \\ &= \mathbb{P}(\delta(l^n) = 1, \delta(\text{Sym}_f(l^n)) = 1) \\ &+ \mathbb{P}(\delta(l^n) = 0, \delta(\text{Sym}_f(l^n)) = 1) \end{aligned}$$

D'où

$$\mathbb{P}(\delta(\text{Sym}_f(l^n)) = 1) = \frac{p_L}{2} + q_L$$

Ce qui nous permet d'obtenir enfin

$$p_f = \frac{2q_L}{p_L + 2q_L}$$

- Si $\text{Sym}_f(l^n) \notin \cup_{k < n} A_k$, alors la fibre symétrique n'a pas encore été testée, et par conséquent on a

$$\begin{aligned} p_f &= \mathbb{P}(\delta(l^n) = 1) \\ &= \mathbb{P}(\delta(l^n) = 1, \delta(\text{Sym}_f(l^n)) = 0) \\ &+ \mathbb{P}(\delta(l^n) = 1, \delta(\text{Sym}_f(l^n)) = 1) \end{aligned}$$

Donc on a finalement

$$p_f = q_L + \frac{p_L}{2}$$

On calcule de la même manière p_g , puis on pose finalement $\mathbb{P}(\delta(l^n) = 1) = \frac{p_f + p_g}{2}$.

On tire alors une valeur au hasard entre 0 et 1, et si cette valeur est inférieure à $\mathbb{P}(\delta(l^n) = 1)$, alors on pose $\delta(l^n) = 1$.

- On définit alors F_{n+1} comme l'ensemble des fibres f n'appartenant pas à $\cup_{k \leq n} F_k$ et telles qu'il existe une fibre $g \in \cup_{k \leq n} F_k$ voisine de f pour laquelle la liaison $l(f, g)$ vérifie $\delta(l(f, g)) = 1$ (i.e. l'ensemble des fibres encore non-considérées qui ont une liaison avec une fibre déjà considérée)
- On définit ensuite $A_{n+1} = \{l(f, g) : f \in F_{n+1}, g \notin \cup_{k \leq n} F_k\}$, que l'on peut aussi écrire $A_{n+1} = \{l(f, g) : f \in F_{n+1}\} \cap \overline{\cup_{k \leq n} A_k}$, c'est-à-dire comme l'ensemble des liaisons attachées aux fibres de F_{n+1} encore non-considérées précédemment.
- Si $A_{n+1} = \emptyset$, alors la région a atteint sa taille maximale, qui est de

$$N = \sum_{k=0}^{n+1} \text{card } F_k$$

On effectue ensuite un très grand nombre d'essais pour chaque jeu de valeur (p_L, q_L) , qui nous donnent donc des ensembles $(N_k^{(p_L, q_L)}(\omega))_{k < K}$ correspondant à des réalisations d'une suite de variables aléatoires indépendantes, identiquement distribuées et intégrables. D'après la loi forte des grands nombres, on a donc finalement :

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K N_k^{(p_L, q_L)}(\omega) = \mathbb{E}(N_1^{(p_L, q_L)})$$

En observant le comportement de $\mathbb{E}(N_1^{(p_L, q_L)})$, son graphe et en l'approchant par une surface analytique, on peut en déduire que $\mathbb{E}(N_1^{(p_L, q_L)})$ est approximable par une fonction du type $(p, q) \mapsto \exp\left(\frac{1}{f(p, q) - f_0}\right)$

Bibliographie

- Ballard, D. H., 1981. Generalizing the Hough Transform to detect arbitrary shapes. *Pattern Recognition*, 13 (2), pp. 111-122.
- Ben-Hur, A., Horn, D., Siegelmann, H. T. & Vapnik, V., 2001. Support vector clustering. *J. Machine Learning Research*, 2, pp. 125-137.
- Bergounioux, M., 2014. Second order variational models for image texture analysis. In: *Advances in Imaging and Electron Physics*, pp. 35-124.
- Besag, J., 1974. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Statistical Society (Series B)*, 36, pp. 192-236.
- Besag, J., 1975. Statistical analysis of non-lattice data. *The Statistician*, 24 (3), pp. 179-195.
- Besag, J., 1977. Efficiency of pseudo-likelihood estimation for simple Gaussian fields. *Biometrika*, 64, pp. 616-618.
- Bigun, J., Bigun, T. & Nilsson, K., 2004. Recognition by symmetry derivatives and the generalized structure tensor. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 26 (12), pp. 1590-1605.
- Bigun, J. & Granlund, G. H., 1987. Optimal orientation detection of linear symmetry, *Proc. of the 1st Int. Conf. on Computer Vision*, pp. 433-438.
- Borgefors, G., 1984. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27, pp. 321-345.
- Borgefors, G., 1986. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34, pp. 344-371.
- Bradley, P. S. & Fayyad, U., 1998. Refining initial points for K-means clustering. *Proc. 15th International Conf. Machine Learning*, pp. 91-99.
- Breu, H., Gil, J., Kirkpatrick, D. & Werman, M., 1995. Linear time euclidean distance transform algorithm. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 17 (5), pp. 529-533.
- Camstra, F. & Verri, A., 2005. A novel kernel method for clustering. *IEEE Trans Pattern Anal. Mach. Intell.*, 27 (5), pp. 801-805.
- Carathéodory, C., 1907. Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen. *Math. Ann.*, 64, pp. 95-115.
- Celeux, G., Forbes, F. & Peyrand, N., 2003. EM procedures using mean field-like approximations for Markov model-based image segmentation. *Pattern Recognition*, 36 (1), pp. 131-144.

- Chang, F., Chen, C. J. & Lu, C. J., 2004. A linear-time component-labeling algorithm using contour tracing technique. *Comput. Vision Image Understanding*, 93, pp. 206-220.
- Cohen, M. & Toussaint, G. T., 1977. On the detection of structures in noisy pictures. *Pattern Recognition*, 9, pp. 95-98.
- Cyganski, D., Noel, W. F. & Orr, J. A., 1990. The analytical Hough transform. *SPIE, Sensing and Reconstruction of Three-Dimensional Objects and Scenes*, 1260, pp. 148-159.
- Danielsson, P.-E., 1980. Euclidean distance mapping. *Computer Vision, Graphics, and Image Processing*, 14, pp. 227-248.
- Deans, S. R., 1981. Hough transform from the Radon transform. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2, pp. 185-188.
- Deans, S. R., 2007. *The Radon transform and some of its applications*. Courier Dover Publications.
- Delaunay, B., 1934. *Sur la sphère vide*. *Izvestia Akad. Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*.
- Derin, H. & Elliott, H., 1987. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9 (1), pp. 39-55.
- Dorst, L. & Smeulders, A. W. M., 1984. Discrete representation of straight lines. *IEEE Trans. Pattern Anal. Machine Intell.*, 6, pp. 450-463.
- Drineas, P., Frieze, A., Kannan, R., Vempala, S., & Vinay, V., 1999. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56, pp. 9-33.
- Duda, P. E. & Hart, R. D., 1972. Use of the Hough Transform to detect lines and curves in pictures. *Communications ACM*, 15, pp. 11-15.
- Fortune, S., 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, Volume 2, pp. 153-174.
- Galambos, C., Kittler, J. & Matas, J., 2001. Gradient based progressive probabilistic Hough transform. *IEE Proceedings on Vision, Image and Signal Processing*, 148 (3), pp. 158-165.
- Galambos, C., Matas, J. & Kittler, J., 1999. *Progressive Probabilistic Hough Transform for line detection*.
- Geman, D., 1991. Random fields and inverse problems in imaging. *Lecture Notes Math.*, 1427, pp. 113-193.
- Geman, S. & Geman, D., 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-6 (6), pp. 721-741.

- Geman, S. & Graffigne, C., 1987. *Markov random field image models and their applications to computer vision*. Berkeley, pp. 1496-1517.
- Graham, R. L., 1972. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1, pp. 132-133.
- Gray, A., Abbena, E. & Salamon, S., 2006. *Modern differential geometry of curves and surfaces with mathematica*. CRC Press.
- Green, P. J. & Sibson, R., 1977. Computing Dirichlet tessellations in the plane. *Comput. Journal*, 21, pp. 168-173.
- Hammersley, J. M. & Clifford, P., 1971. Markov fields on finite graphs and lattices. *Unpublished Manuscript*.
- Han, J. H., Koczy, L. & Poston, T., 1994. Fuzzy Hough transform. *Pattern Recognition Letters*, 15 (7), pp. 649-658.
- Haralick, R. M., 1981. Some neighborhood operations. *Real Time/Parallel Computing Image Analysis*. New York: Plenum Press, pp. 11-35.
- Hastings, W. K., 1970. Monte Carlo sampling methods using Markov chains, and their applications. *Biometrika*, 57, pp. 97-109.
- He, L., Chao, Y., Suzuki, K. & Wu, K., 2009. Fast connected-component labeling. *Pattern Recognition*, 42, pp. 1977-1987.
- Hernandez Hoyos, M. et al., 2000. A deformable vessel model with single point initialization for segmentation, quantification and visualization of blood vessels in 3D MRA. *Lecture Notes in Computer Science*, 1935-MICCAI, pp. 735-745.
- Hough, P. V. C., 1962. *Method and means for recognizing complex patterns*. US, Patent No. 3069654.
- Huang, C. T. & Mitchell, O. R., 1994. A Euclidean distance transformation using grayscale morphology decomposition. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 16, pp. 443-448.
- Illingworth, J. & Kittler, J., 1988. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44, pp. 87-116.
- Ising, E., 1925. Beitrag zur theorie des ferromagnetisms. *Zeitschrift für Physik*, 31, pp. 253-258.
- Jain, A. K., 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31, pp. 651-666.
- Jain, A. K. & Dubes, R. C., 1998. *Algorithms for clustering data*. Prentice Hall.

- Jarvis, R. A., 1973. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2, pp. 18-21.
- Jeng, S. & Tsai, W., 1991. Scale and orientation-invariant generalized Hough Transform—a new approach. *Pattern Recognition*, 24 (11), pp. 1037-1051.
- Jiang, L., 2012. Efficient randomized Hough transform for circle detection using novel probability sampling and feature points. *Optik*, 123, pp. 1834-1840.
- Kälviäinen, H. & Hirvonen, P., 1997. An extension to the Randomized Hough Transform exploiting connectivity. *Pattern Recognition Letters*, 18, pp. 77-85.
- Kälviäinen, H., Hirvonen, P., Xu, L. & Oja, E., 1995. Probabilistic and non-probabilistic Hough transforms-overview and comparisons. *Image and Vision Computing*, 13 (4), pp. 239-252.
- Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R. & Wu, A. Y., 2002. An efficient k-Means clustering algorithm: analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24 (7), pp. 881-892.
- Kass, M. & Witkin, A., 1987. Analyzing oriented patterns. *Computer Vision, Graphics and Image Processing*, 37, pp. 362-385.
- Khoselham, K., 2007. Extending generalized Hough transform to detect 3D objects in laser range data, *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*.
- Kirkpatrick, S., Gellatt, C. D. & Vecchi, M. P., 1982. *Optimization by simulated annealing*, New York: IBM Thomas J. Watson Research Center, Yorktown Heights.
- Kiryati, N. & Bruckstein, A. M., 1991. A probabilistic Hough Transform. *Pattern Recognition*, 24 (4), pp. 303-316.
- Kiryati, N., Kälviäinen, H. & Alaoutinen, S., 2000. Randomized or probabilistic Hough transform: unified performance evaluation. *Pattern Recognition Letters*, 21 (13), pp. 1157-1164.
- Kiryati, N., Lindenbaum, M. & Bruckstein, A. M., 1991. Digital or analog Hough transform?. *Pattern Recognition Letters*, 12 (5), pp. 291-297.
- Klette, R. & Rosenfeld, A., 2004. Digital straightness—a review. *Discrete Applied Mathematics*, 139, pp. 197-230.
- Knutsson, H., 1985. Producing a continuous and distance preserving 5-d vector representation of 3-d orientation, *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management*. pp. 175-182.
- Knutsson, H., 1989. Representing local structure using tensor, *Proc. of the 6th Scandinavian Conf. on Image Analysis*. pp. 244-251.

- Koller, M., Gerig, G., Székely, G. & Dettwiler, D., 1995. Multiscale detection of curvilinear structures in 2-D and 3-D image data, *Int. Conf. on Computer Vision*. pp. 864-869.
- Kolountzakis, M. N. & Kutulakos, K. N., 1992. Fast computation of the Euclidean distance maps for binary images. *Information Processing Letters*, 43 (4), pp. 181-184.
- Krause, M. et al., 2010. Determination of the fibre orientation in composites using the structure tensor and local X-ray transform. *J. Mater. Sci.*, 45, pp. 888-896.
- Leavers, V. F., 1990. Active intelligent vision using the dynamic generalized Hough Transform, *Proc. of the British Machine Vision Conference*. pp. 49-54.
- Lee, D. T. & Schachter, B. J., 1980. Two algorithms for constructing a Delaunay triangulation. *Inter. Journal of Computer and Information Sciences*, 9 (3), pp. 219-242.
- Li, H. & Lavin, M. A., 1986. Fast Hough transform based on the bintree data structure. *IEEE CVPR Conference*, pp. 640-642.
- Li, H., Lavin, M. A. & LeMaster, R. J., 1986. Fast Hough transform: a hierarchical approach. *Comput. Vis. Graph. Image Process.*, 36 (2), pp. 139-161.
- Lindenbaum, M., Koplowitz, J. & Bruckstein, A. M., 1988. On the number of digital straight lines on a NxN grid, *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*. Ann Arbor, pp. 610-615.
- Li, S., 2009. *Markov random field modeling in image analysis*. 3rd ed. London: Springer-Verlag.
- Lloyd, S. P., 1982. Least squares quantization in PCM. *IEEE Trans. Information Theory*, 28, pp. 129-137.
- Lumia, R., 1983. A new three-dimensional connected components algorithm. *Comput. Vision Graphics Image Process.*, 23 (2), pp. 207-217.
- MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations, *Fifth Berkeley Symp. Math. Statistics and Probability*. 1, pp. 281-296.
- Maurer, C. R., Qi, R. & Raghavan, V., 2003. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 25 (2), pp. 265-270.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E., 1953. Equation of state calculations by fast computing machine. *J. Chemical Physics*, 21, pp. 1087-1091.
- Monga, O. & Benayoun, S., 1995. Using partial derivatives of 3D images to extract typical surface features. *Comp. Vision and Image Understanding*, 61 (2), pp. 171-189.

- Muammar, H. & Nixon, M., 1989. Approaches to extending the Hough Transform. *IEEE Inter. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1556-1559.
- Mukhopadhyay, P. & Chaudhuri, B. B., 2015. A survey of Hough Transform. *Pattern Recognition*, 48, pp. 993-1010.
- Mulat, C. et al., 2007. Axis detection method for cylindrical objects. *SPIE, Proc. of 8th Int. Conf. on Quality Control by Artificial Vision*, 6356.
- Natterer, J., 1986. *The Mathematics of Computerized Tomography*.
- Nesterov, Y., 2005. Smooth minimization of non-smooth functions. *Mathematic Programming*, 103 (1), pp. 127-153.
- Ohya, T., Iri, M. & Murota, K., 1984. Improvements of the incremental method for the Voronoi diagrams. *Journal Oper. Res. Soc. Japan*, 27, pp. 306-336.
- Paglieroni, D. W., 1992. A Unified distance transform algorithm and architecture. *Machine Vision Application*, 5, pp. 47-55.
- Paglieroni, D. W., 1992. Distance transforms: properties and machine vision applications. *CVGIP: Graphical Models and Image Processing*, 54, pp. 56-74.
- Palmer, P. L., Kittler, J. & Petrou, M., 1994. Using focus of attention with the Hough transform for accurate line parameter estimation. *Pattern Recognition*, 27 (9), pp. 1127-1134.
- Powell, M. J. D., 1969. A method for nonlinear constraints in minimization problems. *Optimization*. London: R. Fletcher, Academic Press.
- Powell, M. J. D., 1970. A survey of numerical methods for unconstrained optimization. *SIAM Review*, 12 (1), pp. 79-97.
- Princen, J., Illingworth, J. & Kittler, J., 1992. A formal definition of the Hough Transform: properties and relationships. *J. Math. Imaging Vis.*, 1 (2), pp. 153-168.
- Rabin, J., Delon, J. & Gousseau, Y., 2009. A statistical approach to the matching of local features. *SIAM J. Imaging Sciences*, 2 (3), pp. 931-958.
- Radon, J., 1994. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten (On the determination of functions by their integral values along certain manifolds), 75 years of Radon Transform (Vienna, 1992). *Conference Proceedings and Lecture Notes in Mathematical Physics, no. 4*. Cambridge, MA: International Press, pp. 324-339.
- Ragnemalm, I., 1993. The Euclidean distance transform in arbitrary dimensions. *Pattern Recognition Letters*, 14, pp. 883-888.

- Rieger, B. & van Vliet, L. J., 2003. Representing orientation in n-dimensional spaces. *Computer Analysis of Image and Patterns*. N. Petkov & M. A. Westenberg, eds. pp. 17-24.
- Rosenfeld, A., 1969. *Picture processing by computer*. New York/London: Academic Press.
- Rosenfeld, A., 1970. Connectivity in digital pictures. *Journal Assoc. Comput. Machine*, 17 (1), pp. 146-160.
- Rosenfeld, A., 1974. Digital straight lines segments. *IEEE Trans. on Computers*, c-23, pp. 1264-1269.
- Rosenfeld, A. & Pfaltz, J. L., 1966. Sequential operations in digital picture processing. *J. Assoc. Comput. Mach.*, 13, pp. 471-494.
- Rosenfeld, A. & Pfaltz, J. L., 1968. Distance functions on digital pictures. *Pattern Recognition*, 1, pp. 33-61.
- Rudin, L. I., Osher, S. & Fatemi, E., 1991. Nonlinear total variation based noise removal algorithms. *Physica D*, 60, pp. 259-268.
- Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G. & Kikinis, R., 1998. Three-dimensional multiscale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2, pp. 143-168.
- Scott, D. W., 1992. *Multivariate density estimation*. New York: Wiley.
- Serra, J., 1982. *Image analysis and mathematical morphology*. New York: Academic Press.
- Sethian, J. A., 1996. A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci.*, 93 (4), pp. 1591-1595.
- Shaked, D., Yaron, O. & Kiryati, N., 1996. Deriving stopping rules for the probabilistic Hough transform by sequential-analysis. *Comput. Vision Image Underst.*, 63 (3), pp. 512-526.
- Shamos, M. I. & Hoey, D., 1975. Closest-point problems. *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, pp. 151-162.
- Shapiro, S. D., 1974. Detection of lines in noisy pictures using clustering. *2nd IJCPR*. Copenhagen.
- Shen, Y. & Barner, K. E., 2004. Fuzzy vector median-based surface smoothing. *IEEE Trans. Visualization Comput. Graphics*, 10 (3), pp. 252-265.
- Shirai, Y., 1987. Labeling connected regions. *Three-Dimensional Computer Vision*. Berlin: Springer, pp. 86-89.
- Silberberg, T. M., Davis, L. & Harwood, D., 1984. An iterative Hough procedure for three-dimensional object recognition. *Pattern Recognition*, 17 (6), pp. 621-629.

- Tibshirani, R., Walther, G. & Hastie, T., 2001. Estimating the number of clusters in a data set via the gap statistic. *J. Royal Statist. Soc. B*, pp. 411-423.
- Toft, P. A. & Sørensen, J. A., 1996. *The Radon transform-theory and implementation (PhD dissertation)*. Technical University of Denmark (Danmarks Tekniske Universitet), Center for Bachelor of Engineering Studies (Center for Diplomingeniørruddannelse), Center for Information Technology and Electronics (Center for Informationsteknologi og Elektronik).
- Tombari, F. & Stefano, L. D., 2010. Object recognition in 3D scenes with occlusions and clutter by Hough voting, *The 4th Pacific-Rim Symposium on Image and Video Technology (PSIVT)*. pp. 349-355.
- Tong, C. & Kamata, S., 2010. 3D object matching based on spherical Hilbert scanning, *Proc. of the 17th Inter. Conf. on Image Process. (ICIP)*. pp. 2941-2944.
- Tsai, D. M., 1997. An improved generalized Hough transform for the recognition of overlapping objects. *Image Vision Computing*, 15 (12), pp. 877-888.
- Van Veen, T. M. & Groen, F. C. A., 1981. Discretization errors in the Hough transform. *Pattern Recognition*, 14, pp. 137-145.
- Verdonck, B., Bloch, I. & Maître, H., 1996. Accurate segmentation of blood vessels from 3D medical images, *Proc. of the IEEE Int. Conf. on Image Processing*. pp. 311-314.
- Vincent, L., 1991. Exact euclidean distance function by chain propagations. *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, pp. 520-525.
- Voronoi, G., 1908. *Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire: Recherche sur les paralléloèdres primitifs*. J. Reine Angew. Math.
- Wallace, R. S., 1985. A modified Hough transform for lines, *IEEE CVPR Conference*. San Francisco.
- Walsh, D. & Raftery, A. E., 2002. Accurate and efficient curve detection in images: the importance sampling Hough Transform. *Pattern Recognition*, 35, pp. 1421-1431.
- Wang, L., Liu, J. & Li, S., 2000. MRF parameter estimation by MCMC method. *Pattern Recognition*, 33, pp. 1919-1925.
- Wu, F. Y., 1982. The Potts model. *Reviews of Modern Physics*, 54 (1), pp. 235-267.
- Xu, L. & Oja, E., 1990. A new curve detection method: randomized Hough transform (RHT). *Pattern Recognition Letters*, 11 (5), pp. 331-338.
- Xu, L. & Oja, E., 1993. Randomized Hough transform (RHT): Basic mechanism, algorithm and computational complexities. *CVGIP, Image Understanding*, 57, pp. 131-154.

Yla-Jaaski, A. & Kiryati, N., 1994. Adaptive termination of voting in the probabilistic circular Hough transform. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16 (9), pp. 911-915.

Yuen, H. K., Illingworth, J. & Kittler, J., 1989. Detecting partially occluded ellipses using the Hough Transform. *Image Vision Computing*, 7 (1), pp. 31-37.

Yuen, H. K., Princen, J., Illingworth, J. & Kittler, J., 1990. Comparative study of Hough Transform methods for circle finding. *Image and Vision Computing*, 8 (1), pp. 71-77.

Table des matières

1.	Introduction.....	5
1.1.	Problématique de la thèse	5
1.2.	Etat de l'art	10
1.2.1.	Introduction.....	10
1.2.2.	Formalisme	11
1.2.3.	Transformée de Hough	12
1.2.4.	Tenseur de Structure.....	24
1.3.	Résultats significatifs obtenus	38
1.3.1.	Matériaux fibreux.....	38
1.3.2.	Mousses et matériaux assimilés	43
2.	Recherche de droites : application à des composites Carbone/Résine.....	47
2.1.	Introduction et propriétés des images utilisées.....	47
2.2.	Contraintes et facilités accordées par le contexte et les images	51
2.3.	Avantages et Inconvénients d'une méthode géométrique. Comparaison théorique des méthodes.....	54
2.4.	Algorithme développé	63
2.4.1.	Champs de Markov.....	63
2.4.2.	Détection de Lignes.....	72
2.4.3.	Association des fibres.....	97
2.5.	Découpage et suivi.....	113
2.6.	Tests et Résultats.....	118
2.6.1.	Image de fibres sans contacts	118
2.6.2.	Images de fibres avec contacts, excepté sur les faces	121
2.6.3.	Images de fibres avec contacts	124
2.6.4.	Images de fibres cohérentes spatialement	125
3.	Recherche de sphères : applications à des mousses de silice et polyuréthane	129
3.1.	Introduction et propriétés des images utilisées.....	129
3.2.	Contraintes et facilités accordées par le contexte et les images	132
3.3.	Avantages et Inconvénients des méthodes.....	135
3.4.	Algorithme développé	145
3.4.1.	Pré-Traitement : Transformée de Hough Circulaire	145

3.4.2.	Détection de sphères	153
3.4.3.	Traitement : Classification des maximums de la carte de distance	156
3.4.4.	Post-Traitement : Vérification des sphères de manière locale	160
3.5.	Tests et Résultats	174
3.5.1.	Images de sphères sans contacts	174
3.5.2.	Images de sphères avec contacts et sans interpénétrations	176
3.5.3.	Images de sphères avec interpénétrations	177
3.5.4.	Images de sphères incomplètes avec interpénétrations	178
4.	Applications et utilisations des résultats	182
4.1.	Matériaux composites carbone résine	182
4.2.	Mousses de polyuréthane	204
4.3.	Mousses de silice	213
5.	Conclusions et perspectives	223
5.1.	Perspectives et améliorations	223
5.2.	Conclusions	230
Annexe 1 : Comparaison théorique de la PHT et de la RHT		230
Annexe 2 : Estimation de la connexité dans un fil carbone/résine		238
Bibliographie		246

**Analyse rapide d'images 3D de matériaux hétérogènes :
identification de la structure des milieux et application à leur
caractérisation multi-échelle**

La simulation numérique est un outil de plus en plus utilisé pour la conception et le choix de matériaux composites. Celle-ci permet en effet de générer et tester numériquement des structures très diverses plus facilement et plus rapidement qu'avec des procédés de fabrication et de tests réels. Suite au choix d'un matériau virtuel et sa fabrication tangible, un retour sur expérience est nécessaire afin de valider simultanément la simulation et le procédé de fabrication. Pour cela, la numérisation des matériaux fabriqués permet de renvoyer une modélisation comparable aux structures virtuelles générées. Il devient possible d'appliquer les mêmes algorithmes de simulation et de vérifier les prévisions.

Le sujet de cette thèse consiste donc en la modélisation de matériaux composites réels à partir d'images 3D, afin d'y retrouver le matériau virtuel originel. Des méthodes de traitement d'images sont appliquées aux images afin d'en extraire les informations sur la structure du matériau, c'est-à-dire la localisation de chaque constituant et, s'il y a lieu, de leurs orientations. Ces connaissances permettent théoriquement de simuler le comportement thermique et mécanique de structures constituées du matériau étudié.

Cependant, en pratique, représenter fidèlement les composites demande de prendre une discrétisation très fine. Par conséquent, une structure macroscopique demande beaucoup trop de points de discrétisation et donc de temps de calcul et de mémoire pour simuler son comportement. Un aspect de la thèse consiste donc aussi en la détermination d'un matériau homogène équivalent, permettant d'alléger la charge de calcul pour les algorithmes de simulation.

Mots clés : Traitement d'Image, Matériaux Composites, Images 3D

**Fast analysis of heterogeneous materials 3D-images: mediums
structure identification and application to their multiscale
characterization**

Digital simulation is a wide-spreading tool for composite materials design and choice. Indeed it allows to generate and test digitally various structures more easily and quickly than with real manufacturing and tests processes. A feedback is needed following the choice and the fabrication of a virtual material in order to simultaneously validate the simulation and the fabrication process. With this aim, models similar to generated virtual structures are obtained by digitization of manufacturing materials. The same simulation algorithms can then be applied and allow to verify the forecasts.

This thesis is also about the modelling of composite materials from 3D images, in order to rediscover in them the original virtual material. Image processing methods are applied to images to extract material structure data, i.e. each constituent localization, and orientation if applicable. These knowledge theoretically allow to simulate thermal and mechanical behavior of structures constituted of studied material.

However to accurately represent composites requires practically a very small discretization step. Therefore behavior simulation of a macroscopic structure needs too much discretization points, and then time and memory. Hence a part of this thesis focuses also on determination of equivalent homogeneous material problem, which allows, when resolved, to lighten calculation time for simulation algorithms.

Keywords : Image Processing, Composite Materials, 3D Images

