



**HAL**  
open science

# Développement d'algorithmes de métrologie dédiés à la caractérisation de nano-objets à partir d'informations hétérogènes

Alexandre Derville

## ► To cite this version:

Alexandre Derville. Développement d'algorithmes de métrologie dédiés à la caractérisation de nano-objets à partir d'informations hétérogènes. Traitement des images [eess.IV]. Université Grenoble Alpes, 2018. Français. NNT : 2018GREAM082 . tel-02102398

**HAL Id: tel-02102398**

**<https://theses.hal.science/tel-02102398v1>**

Submitted on 17 Apr 2019

**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.

## THÈSE

Pour obtenir le grade de

### **DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ DE GRENOBLE ALPES**

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 25 mai 2016

Présentée par

**Alexandre DERVILLÉ**

Thèse dirigée par **Jean-François COEURJOLLY**  
et **Marianne CLAUSEL**

préparée au sein du **Laboratoire Jean Kuntzmann**  
et de l'**École Doctorale Mathématiques, Sciences et Technologies de  
l'Information, Informatique**

## **Développement d'algorithmes de métrologie dédiés à la caractérisa- tion de nano-objets à partir d'infor- mations hétérogènes.**

Thèse soutenue publiquement le **20 Décembre 2018**,  
devant le jury composé de :

**Mme Adeline Leclercq Samson**

Professeure, Université Grenoble Alpes, Présidente

**M. Bruno Galerne**

Professeur, Université Orléans, Rapporteur

**Mme Ségolen Geffray**

Maître de conférences, Université de Strasbourg, Examinatrice

**Mme, Marianne Clausel**

Professeure, Université de Lorraine, Directrice de thèse

**M. Jean-François Coeurjolly**

Professeur, Université de Québec à Montréal, Directeur de thèse



---

**Résumé :** Ces travaux de thèse s'inscrivent dans le contexte technico/économique des nanomatériaux notamment les nanoparticules et les copolymères. Aujourd'hui, une révolution technologique est en cours avec l'introduction de ces matériaux dans des matrices plus ou moins complexes présentes dans notre quotidien (santé, cosmétique, bâtiment, agro-alimentaire...). Ces matériaux confèrent à ces produits des propriétés uniques (mécanique, électrique, chimique, thermique, ...). Cette omniprésence associée aux enjeux économiques engendre deux problématiques liées au contrôle des procédés de fabrication et à la métrologie associée. La première est de garantir une traçabilité de ces nanomatériaux afin de prévenir tout risque sanitaire et environnemental et la seconde est d'optimiser le développement des procédés afin de pérenniser des filières économiques rentables. Pour cela, les deux techniques les plus courantes de métrologie utilisées sont : la microscopie électronique à balayage (MEB) et la microscopie à force atomique (AFM).

Le premier volet des travaux est consacré au développement d'une méthodologie de fusion de données permettant d'analyser automatiquement les données en provenance de chaque microscope et d'utiliser leurs points forts respectifs afin de réduire les incertitudes de mesure en trois dimensions. Une première partie a été consacrée à la correction d'un défaut majeur d'asservissement de l'AFM qui génère des dérives et/ou sauts dans les signaux. Nous présentons une technique dirigée par les données permettant une correction de ces signaux. La méthode présentée a l'avantage de ne pas faire d'hypothèses sur les objets et leurs positions. Elle peut être utilisée en routine automatique pour l'amélioration du signal avant l'analyse des objets.

La deuxième partie est consacrée au développement d'une méthode d'analyse automatique des images de nanoparticules sphériques en provenance d'un AFM ou d'un MEB. Dans le but de développer une traçabilité en 3D, il est nécessaire d'identifier et de mesurer les nanoparticules identiques qui ont été mesurées à la fois sur l'AFM et sur le MEB. Afin d'obtenir deux estimations du diamètre sur la même particule physique, nous avons développé une technique qui permet de mettre en correspondance les particules. Partant des estimations pour les deux types de microscopie, avec des particules présentes dans les deux types d'images ou non, nous présentons une technique qui permet l'agrégation d'estimateurs sur les populations de diamètres afin d'obtenir une valeur plus fiable des propriétés du diamètre des particules.

Le second volet de cette thèse est dédié à l'optimisation d'un procédé de fabrication de copolymères à blocs (structures lamellaires) afin d'exploiter toutes les grandeurs caractéristiques utilisées pour la validation du procédé (largeur de ligne, période, rugosité, taux de défauts) notamment à partir d'images MEB afin de les mettre en correspondance avec un ensemble de paramètres de procédé. En effet, lors du développement d'un nouveau procédé, un plan d'expériences est effectué. L'analyse de ce dernier permet d'estimer manuellement une fenêtre de procédé plus ou moins précise (estimation liée à l'expertise de l'ingénieur matériaux). L'étape est répétée jusqu'à l'obtention des caractéristiques souhaitées. Afin d'accélérer le développement, nous avons étudié une façon de prédire le résultat du procédé de fabrication sur l'espace des paramètres. Pour cela, nous avons étudié différentes techniques de régression que nous présentons afin de proposer une méthodologie automatique d'optimisation des paramètres d'un procédé alimentée par les caractéristiques d'images AFM et/ou MEB. Ces travaux d'agrégations d'estimateurs et d'optimisation de fenêtre de procédés permettent d'envisager le développement d'une standardisation d'analyse automatique de données issues de MEB et d'AFM en vue du développement d'une norme de traçabilité des nanomatériaux.

**Mots clefs :** M-estimateurs, agrégation d'estimateurs, méthodes de régression, microscope à force atomique, microscope électronique à balayage

---

**Abstract :** This thesis is included in the technical and economical context of nanomaterials, more specifically nanoparticles and block copolymer. Today, we observe a technological revolution with the introduction of these materials into matrices more or less complex present in our daily lives (health, cosmetics, buildings, food ...). These materials yield unique properties to these products (mechanical, electrical, chemical, thermal ...). This omnipresence associated with the economic stakes generates two problems related to the process control and associated metrology. The first is to ensure traceability of these nanomaterials in order to prevent any health and environmental risks and the second is to optimize the development of processes in order to sustain profitable economic sectors. For this, the two most common metrology techniques used are : scanning electron microscopy (SEM) and atomic force microscopy (AFM).

The first phase of the work is devoted to the development of a data fusion methodology that automatically analyzes data from each microscope and uses their respective strengths to reduce measurement uncertainties in three dimensions. A first part was dedicated to the correction of a major defect of the AFM which generates drifts and / or jumps in the signals. We present a data-driven methodology, fast to implement and which accurately corrects these deviations. The proposed methodology makes no assumption on the object locations and can therefore be used as an efficient preprocessing routine for signal enhancement before object analysis.

The second part is dedicated to the development of a method for automatic analysis of spherical nanoparticle images coming from an AFM or a SEM. In order to develop 3D traceability, it is mandatory to identify and measure the identical nanoparticles that have been measured on both AFM and SEM. In order to obtain two estimations of the diameter on the same physical particle, we developed a technique that allows to match the particles. Starting from estimates for both types of microscopy, with particles present in both kinds of images or not, we present a technique that allows the aggregation of estimators on diameter populations in order to obtain a more reliable value of properties of the particle diameter. The second phase of this thesis is dedicated to the optimization of a block copolymer process (lamellar structures) in order to capitalize on all the characteristic quantities used for the validation of the process (line width, period, roughness, defects rate) in particular from SEM images for the purpose of matching them with a set of process parameters.

Indeed, during the development of a new process, an experimental plan is carried out. The analysis of the latter makes it possible to manually estimate a more or less precise process window (estimate related to the expertise of the materials engineer). The step is reiterated until the desired characteristics are obtained. In order to accelerate the development, we have studied a way of predicting the result of the process on the parameter space. For this, we studied different regression techniques that we present to propose an automatic methodology for optimizing the parameters of a process powered by AFM and / or SEM image characteristics. This work of estimator aggregation and process window optimization makes it possible to consider the development of a standardization of automatic analysis of SEM and AFM data for the development of a standard for traceability of nanomaterials.

**Keywords :** M-estimators, estimator aggregation, regression techniques, atomic force microscopy, scanning electron microscopy

# Remerciements

La section des remerciements est toujours la partie la plus lu dans un manuscrit de thèse ! Donc sans plus attendre, les nominés sont :

- **Coraline** : sans toi je n'aurais pas pu faire cette thèse. Tout d'abord je tiens à te dire combien je t'admire pour avoir réussi à me supporter tout ce temps et la patience dont tu as fait preuve en relisant cette thèse je ne sais combien de fois ! Tu as toujours été là pour moi et je ne pourrai jamais assez t'en remercier.
- **Mes enfants**. Dans l'ordre des naissances Armand, Bastien puis Corentin. Vous m'avez permis d'avoir une grande résistance au manque de sommeil, cela m'a bien aidé lors de la rédaction de ce manuscrit.
- **Marianne et Jean-François** : Je vous remercie tous les deux pour toutes ces heures passées en vidéo conférence (malgré la connexion souvent mauvaise !). Même si on n'était pas proche physiquement, vous avez toujours su trouver du temps pour moi et je vous en suis reconnaissant ! Techniquement, vous avez su me cadrer quand j'en avais besoin et que je ne me posais pas les bonnes questions, j'ai appris beaucoup à vos côtés.
- **Mes collègues** : Je veux pas faire de jaloux donc dans l'ordre alphabétique : Aurélien, Benoît, Eric, Guilhem, Guillaume, Guy, Johann, Julien, Sergio, Thibault et Yann. Vous m'avez tous impressionné avec votre faculté à supporter le rythme effréné qu'on a depuis maintenant quelques années. Merci également pour ces moments de délires qui permettent de décompresser.

Pour finir à toutes et à tous un grand, un très grand **MERCI**

---

# Table des matières

<b>Introduction et contexte du travail</b>	<b>11</b>
<b>Bibliographie</b>	<b>17</b>
<b>1 Redressement d'image AFM</b>	<b>19</b>
Introduction . . . . .	19
1.1 Les mesures par microscopie à force atomique . . . . .	21
1.1.1 Des mesures sujettes à défauts . . . . .	21
1.1.2 Correction des défauts : logiciels existants . . . . .	24
1.2 Choix et définition des critères de qualité . . . . .	28
1.2.1 Objectifs . . . . .	29
1.2.2 Les différents critères . . . . .	29
1.3 Redressement d'images AFM . . . . .	37
1.3.1 Estimation polynomiale classique et robuste . . . . .	37
1.3.2 Méthodes de redressement . . . . .	44
1.3.3 Utilisation de l'information globale . . . . .	48
1.4 Comparaison des méthodes . . . . .	58
1.4.1 Liste des logiciels et techniques employés . . . . .	58
1.4.2 Liste des critères et explications . . . . .	59
1.4.3 Comparaison des méthodes sur des images types . . . . .	59
1.5 Analyse d'incertitude de la méthode . . . . .	74
1.5.1 Protocole de test . . . . .	74
1.5.2 Résultats . . . . .	75
1.6 Conclusion . . . . .	76
<b>Annexes</b>	<b>77</b>
1.A Description du procédé de mesure par AFM . . . . .	77
1.A.1 Fonctionnement d'un AFM . . . . .	77
1.A.2 Les différents modes de mesure . . . . .	79
<b>Bibliographie</b>	<b>81</b>



## TABLE DES MATIÈRES

---

<b>2 Agrégation d'estimateurs pour l'estimation de diamètres de particules</b>	<b>83</b>
Introduction	83
2.1 Analyse d'images de nanoparticules	85
2.1.1 Analyse d'images MEB	85
2.1.2 Images AFM	90
2.2 Recalage et correspondance des images	93
2.3 Agrégation d'estimateurs	98
2.3.1 Présentation de la méthode	98
2.3.2 Pratique/Implémentation	100
2.3.3 Résultats agrégation	102
<b>Annexes</b>	<b>111</b>
2.A Principe de fonctionnement d'un MEB	111
2.B Importance de la dépendance entre échantillons	113
2.B.1 Présentation des estimateurs étudiés	113
2.B.2 Présentation des variances d'estimateurs étudiées	114
2.B.3 Résultats	115
2.B.4 Conclusion	116
<b>Bibliographie</b>	<b>119</b>
<b>3 Optimisation de paramètres de procédé de fabrication</b>	<b>121</b>
Introduction	121
3.1 Introduction	123
3.1.1 Présentation du lamellaire	123
3.1.2 Objectifs de l'analyse d'image de lamellaire	123
3.1.3 État de l'art	127
3.2 Analyse d'image lamellaire	127
3.2.1 Calcul de la période	128
3.2.2 Calcul de l'orientation	129
3.2.3 Pré-traitement	130
3.2.4 Binarisation	131
3.2.5 Calcul du squelette	132
3.2.6 Extraction des défauts	132
3.2.7 Calcul des dimensions des lignes	133
3.2.8 Conclusion	134
3.3 Méthodes de régression	135
3.3.1 Description des données	135
3.3.2 Régressions linéaires pénalisées	136
3.3.3 Interpolation par pondération inverse à la distance	137

## TABLE DES MATIÈRES

---

3.3.4	Krigeage . . . . .	137
3.3.5	Réseau de neurones . . . . .	139
3.3.6	Forêt aléatoire : Random Forest . . . . .	140
3.4	Résultats . . . . .	142
3.4.1	Critères de performance . . . . .	142
3.4.2	Résultats régression . . . . .	143
3.5	Conclusion et discussion . . . . .	150
	<b>Bibliographie</b>	<b>151</b>
<b>4</b>	<b>Intégration dans les produits Pollen</b>	<b>153</b>
	Introduction . . . . .	153
4.1	Généralités sur le développement logiciel . . . . .	155
4.1.1	La modélisation UML . . . . .	155
4.1.2	Chaîne d'intégration . . . . .	155
4.2	Redressement d'images AFM . . . . .	157
4.3	Algorithme d'analyses de nanoparticules . . . . .	159
4.3.1	Analyse d'images MEB et AFM . . . . .	159
4.3.2	Agrégation d'estimateurs . . . . .	161
4.4	Analyse d'images lamellaire . . . . .	165
	<b>Bibliographie</b>	<b>167</b>

*TABLE DES MATIÈRES*

---

# Introduction et contexte du travail

Les matériaux à hautes performances sont des matériaux présents dans notre quotidien. Ils sont omniprésents dans un grand nombre de domaines industriels du fait de leurs propriétés uniques ou nouvelles (thermiques, dimensionnelles, mécaniques, électriques, masses, ...) que ne peuvent avoir les matériaux traditionnels. Ils font partie intégrante de notre vie quotidienne et on les retrouve dans notre salle de bain, nos ordinateurs et les médicaments que nous prenons. Ce manuscrit s'inscrit dans ce contexte technico/économique. Nous allons nous intéresser à deux domaines particuliers des matériaux hautes performances. La première partie de la thèse s'articule autour des nanoparticules, la seconde porte sur l'analyse de procédés de fabrication dans l'industrie du semi-conducteurs.

Les nanoparticules sont des objets présents dans quantité de produits et dans quasiment tous les domaines industriels mais elles restent méconnues. En effet, nous les trouvons dans des produits du quotidien comme dans les cosmétiques (crème solaire, dentifrice, ...) ou bien les produits alimentaires (bonbons, produit lyophilisés, ...) mais aussi dans les peintures, pneus et bien d'autres encore. Elles sont utilisées pour leurs propriétés uniques. En effet, leur petitesse leur octroie des propriétés physico-chimiques surprenantes et explique l'engouement des industriels pour ces matériaux. Beaucoup de laboratoires pharmaceutiques se penchent sur leur utilisation dans les médicaments de demain. Les nanoparticules sont des objets de tailles nanométriques. Leur définition exacte, décrit dans la norme [ISO/TS 2768, CEN \(2008\)](#), définit des objets dont la plus petite dimension fait moins de 100nm. De nos jours, on entend beaucoup parler de ces objets mais les effets qu'ils ont sur notre santé restent encore incertains. Le manque de métrologie dans ce domaine constitue un frein à l'évaluation de la toxicité des nanoparticules, comme l'illustrent les rapports de l'[Agence française de sécurité sanitaire de l'environnement et du travail \(2006\)](#), de l'[Agence française de sécurité sanitaire de l'environnement et du travail \(2010\)](#) et de l'[European Food Safety](#)

---

[Authority \(2009\)](#).

C'est pourquoi les autorités souhaitent réglementer l'usage des nanoparticules et obligent les industriels à déclarer leur usage. D'abord les autorités française, puis les autorités européennes ont suivi. La mise en place de nouvelles normes est en cours. Comme expliqué dans la publication de [Feltin et al. \(2012\)](#), le comité technique de l'ISO en charge de la normalisation des nanomatériaux (ISO/TC 229) a défini un ensemble de paramètres physico-chimiques déterminants pour l'étude des nano-objets fabriqués par intervention humaine et soumis aux tests toxicologiques. Cet ensemble de dix paramètres comprend :

- la taille ;
- la distribution en taille ;
- la forme ;
- l'état d'agrégation ou d'agglomération ;
- la composition chimique ;
- la structure cristallographique ;
- la surface spécifique ;
- la charge en surface ;
- la chimie de surface ;
- la solubilité/dispersibilité.

Cette obligation de déclaration contraint les industriels à devoir analyser les particules qu'ils utilisent faisant ainsi grandir les besoins en métrologie associés. Les techniques d'analyse traditionnelles, telles que les techniques optiques, ne peuvent s'appliquer simplement à l'analyse des nanoparticules en raison de leur très faibles dimensions. Les techniques utilisées pour analyser ces objets sont les mêmes que dans le milieu du semi-conducteur, une industrie maîtrisant la métrologie à ces dimensions. En France, le laboratoire national de métrologie et d'essais (le LNE) dispose d'une plateforme spécialisée dans la caractérisation de nanomatériaux, présentée dans l'article de [Feltin et al. \(2012\)](#). Cette plateforme comporte les équipements nécessaires à la caractérisation de nanomatériaux selon l'ISO/TC 229. Dans ce manuscrit, nous ne nous intéresserons qu'aux mesures dimensionnelles pour la caractérisation de nanomatériaux.

La première partie de ce manuscrit résume les travaux effectués dans un contrat de collaboration avec le LNE. Son but est de fournir une solution technique pour l'analyse et la fusion de données sortant des différents équipements de la plateforme de caractérisation du LNE, pour la caractérisation de nanoparticules. Cette première partie s'articule autour de l'analyse de données par technique individuelle ainsi que la combinaison des méthodes. Au cours de cette collaboration, nous nous sommes intéressés à deux tech-

niques de caractérisation : la première est la microscopie à force atomique (AFM) et la seconde la microscopie électronique à balayage (MEB). L'objectif était de faire de l'analyse de nanoparticules sphériques sur ces données. Les deux types de microscopes permettent d'obtenir une image de particules pour le MEB et une image de la topologie de surface pour l'AFM et sont respectivement présentées dans les Annexes 2.A et 1.A,

La technique de microscopie à force atomique nécessite une première phase de correction du signal sortant du microscope. En effet, comme nous le décrivons dans le Chapitre 1, les défauts présents rendent l'analyse des objets complexe, c'est pourquoi cette phase de correction est nécessaire. Nous effectuerons en outre un bilan des méthodes actuelles pour la correction de données AFM afin d'en extraire les besoins courants. Nous présenterons alors les approches possibles pour la correction via des méthodes d'estimation robustes et l'utilisation des propriétés du signal AFM. Nous présenterons plusieurs approches pour la correction et, afin de valider nos résultats, effectuerons un comparatif avec les techniques traditionnelles. Le Chapitre 1 présente une étude complète sur la correction d'image AFM.

Une fois le signal amélioré, nous avons développé une technique d'extraction de particules et de leurs propriétés pour les images AFM et MEB, en utilisant des techniques de traitement du signal et de l'image. Le Chapitre 2 présente les méthodes d'extraction de particules pour chacune des techniques. L'objectif de l'extraction de particules est de pouvoir détecter et analyser celles-ci dans les images. L'analyse est faite en deux temps, via l'extraction par des méthodes de traitement du signal et de l'image puis dans un second temps l'analyse dimensionnelle des particules. L'analyse doit être en mesure d'extraire les particules primaires, c'est-à-dire les particules présentes dans les images, soit dans les agrégats de particules, soit des particules isolées.

Le LNE dispose d'un protocole de mesure permettant de scanner la même zone pour les deux types de microscopes. Les méthodes précédentes permettent d'extraire les particules et leurs propriétés pour chacune des deux techniques. Nous avons développé une méthode de mise en correspondance de particules extraites par les deux techniques, décrite dans le Chapitre 2. Nous disposerons alors de différentes estimations pour les diamètres des particules. Nous avons soit deux estimations de dimensions des particules mais sur des sous échantillons distincts, soit une estimation de dimension par les deux techniques de mesures, pour chaque particule. Nous pouvons alors étudier comment combiner les estimations de paramètres tel que le diamètre selon différentes méthodes. Le Chapitre 2 présente une technique générique

---

de combinaison d'estimateurs.

L'industrie du semi-conducteur suit la loi de Moore depuis plus de 50 ans, qui prédit un doublement du nombre de transistors sur un microprocesseur tous les deux ans, mais cette miniaturisation a un coût élevé. En effet, le coût de chaque nouveau nœud technologique croît de manière exponentielle du fait de l'augmentation du nombre et de la complexité des étapes de fabrication. Une solution pour fabriquer des lignes ou des cylindres de quelques nanomètres de large, facilement et à bas coût, est d'utiliser des copolymères à blocs. Ces copolymères, que nous présenterons plus en détails dans le Chapitre 3 sont des matériaux qui s'auto-assemblent pour former des lignes ou des cylindres. Ce chapitre présente l'algorithme développé pour analyser d'image de microscopie électronique de lamellaire, autrement nommé copolymère à blocs en ligne, qui est une nouvelle méthode pour la fabrication de lignes dans le semi-conducteur.

Le développement d'un nouveau procédé de lamellaire est complexe, il faut gérer de nombreux paramètres, comme la formulation chimique, la concentration, la quantité de matériaux à déposer et bien d'autres. La méthode classique de mise au point d'un nouveau procédé est la suivante. L'expert dispose d'un plan d'expérience initial sur les paramètres qu'il veut optimiser puis réalise les expériences associées pour les analyser. Une fois tous les résultats obtenus, il détermine une zone d'intérêt qui est une région dans l'espace des paramètres donnant les performances les plus proches de celles souhaitées pour le matériau. Cette zone d'intérêt s'appelle la fenêtre de procédé. Si la fenêtre n'est pas assez précise, il refait un nouveau plan d'expérience dans la première fenêtre de procédé afin de l'affiner. Ce processus de développement est problématique pour les entreprises car chaque expérience est coûteuse et l'analyse qui suit la réalisation de ces expériences est longue.

Nous nous sommes intéressés à cette problématique pour assister un expert développant un nouveau procédé de fabrication. Pour cela, la première étape est d'estimer le résultat pour des paramètres non présents dans le plan d'expérience initial. Une fois une technique d'estimation obtenue, nous sommes en mesure d'estimer le résultat sur l'espace des paramètres. L'étape suivante est de déterminer dans cet espace, la zone la plus propice aux propriétés souhaitées. Cette zone est appelé la fenêtre de procédé.

Le dernier chapitre, le Chapitre 4 présente quant à lui l'intégration des différents algorithmes développés durant cette thèse dans le produit de POLLEN METROLOGY, le logiciel Platypus<sup>TM</sup>. Ce chapitre décrira les différentes phases entre prototypage, développement et intégration dans un logiciel industriel.

Cette thèse CIFRE s'est déroulée dans une toute jeune société, la start up POLLEN METROLOGY. La thèse a en effet débuté au bout d'une année d'existence de la société qui comportait alors 4 employés. Trois ans après, alors que ce travail de thèse arrive à son terme, nous avons suivi et participé à toutes les étapes du développement de POLLEN METROLOGY, qui compte maintenant 18 employés (en septembre 2018). Nous avons depuis lors participé au développement via du prototypage, de l'étude de faisabilité pour de nouveaux projets et du développement afin d'améliorer le logiciel commercialisé par l'entreprise. Par exemple, le contrat de collaboration avec le LNE comportait des algorithmes d'analyse d'images pour de l'étalonnage. Ces sujets tiers bien qu'intéressants (et chronophages) ne faisaient pas partie du sujet de thèse initial et/ou ne requéraient pas de développement recherche, c'est pourquoi ils n'apparaîtront pas dans ce manuscrit.



---

# Bibliographie

Agence française de sécurité sanitaire de l'environnement et du travail (2006).  
Les nanomatériaux : Effets sur la santé de l'homme et sur l'environnement.

Agence française de sécurité sanitaire de l'environnement et du travail (2010).  
Les nanomatériaux : Évaluation des risques liés aux nanomatériaux pour  
la population générale et pour l'environnement. *La Doc. Fr*, 3 :1–207.

European Food Safety Authority (2009). The potential risks arising from  
nanoscience and nanotechnologies on food and feed safety. *EFSA Journal*,  
7(3) :958.

Feltin, N., Delvallée, A., Ducourtieux, S., Gournay, P., Poyet, B., Trabelsi,  
M., Boukellal, Y., and Piquemal, F. (2012). Carmen : une plateforme de  
caractérisation métrologique dédiée aux nanomatériaux carmen : a metro-  
logical characterization platform for nanomaterials. *Revue Française de  
métrologie*, (31) :41–54.

ISO/TS 2768, CEN (2008). Nanotechnologies—terminology and definitions  
for nano-objects—nanoparticle, nanofiber and nanoplate. Standard, Inter-  
national Organization for Standardization, Geneva, CH.

*BIBLIOGRAPHIE*

---

# Chapitre 1

## Redressement d'image AFM

### Introduction

La microscopie à force atomique (autrement nommée AFM pour Atomique Force Microscopy) est une technique de mesure très répandue dans les laboratoires et entreprises, permettant de caractériser des surfaces ou objets à des échelles nanométriques. Cette technique d'analyse permet de générer des images AFM qui rendent compte de la topologie de la surface. Malheureusement, cette technique de mesure est sujette à certains types de défauts. Les plus fréquents et ceux que nous allons étudier dans ce chapitre sont une dérive sur les mesures ainsi que des sauts d'asservissements. La correction des dérives dans les données de microscopie à force atomique est une problématique récurrente. En effet, afin d'obtenir les mesures les plus précises possibles, le métrologue doit corriger ces défauts. Ce chapitre est dédié à la correction d'image AFM, qui peut également être nommée redressement d'image AFM.

Le chapitre est structuré de la façon suivante, tout d'abord nous décrivons dans la section 1.1 les différents types de défauts qui doivent être corrigés ainsi que les solutions logicielles existantes. Comme nous le verrons dans la section 1.1, les solutions logicielles existantes ne sont pas automatiques et l'objectif de ce chapitre est de présenter une nouvelle technique, dirigée par les données. C'est pourquoi, il nous faut mettre au point une base de comparaison. Nous présentons des critères pour l'évaluation de la qualité d'un algorithme de correction d'image AFM.

Nous verrons ensuite que pour le redressement d'image AFM, il faut estimer la dérive. Nous présenterons alors plusieurs techniques d'interpolation. Nous présenterons les M-estimateurs ainsi qu'une variante, les M-estimateurs

---

asymétriques. Nous appliquerons ensuite ces techniques pour mettre au point différentes stratégies de redressement permettant de répondre à la problématique de manière rapide, automatique et ne nécessitant aucun paramètre.

Enfin, un comparatif des méthodes développées et de différents logiciels commerciaux et libres sera fait sur quatre images représentatives, plus ou moins complexes à analyser.

Nous effectuerons ensuite une étude sur la robustesse à du bruit additionnel pour voir la sensibilité de l’algorithme développé.

Le travail présenté dans ce chapitre fait l’objet de la publication [Dervillé et al. \(2018\)](#).

## Sommaire

---

<b>Introduction</b>	<b>19</b>
<b>1.1 Les mesures par microscopie à force atomique</b>	<b>21</b>
1.1.1 Des mesures sujettes à défauts	21
1.1.2 Correction des défauts : logiciels existants	24
<b>1.2 Choix et définition des critères de qualité</b>	<b>28</b>
1.2.1 Objectifs	29
1.2.2 Les différents critères	29
<b>1.3 Redressement d’images AFM</b>	<b>37</b>
1.3.1 Estimation polynomiale classique et robuste	37
1.3.2 Méthodes de redressement	44
1.3.3 Utilisation de l’information globale	48
<b>1.4 Comparaison des méthodes</b>	<b>58</b>
1.4.1 Liste des logiciels et techniques employés	58
1.4.2 Liste des critères et explications	59
1.4.3 Comparaison des méthodes sur des images types	59
<b>1.5 Analyse d’incertitude de la méthode</b>	<b>74</b>
1.5.1 Protocole de test	74
1.5.2 Résultats	75
<b>1.6 Conclusion</b>	<b>76</b>

---

## 1.1 Les mesures par microscopie à force atomique

Dans cette section, nous allons tout d'abord étudier les défauts de la technique de mesure par microscopie à force atomique (que l'on notera par AFM<sup>1</sup> dans la suite du document). Le fonctionnement d'un AFM est décrit dans l'Annexe 1.A. Ensuite nous présenterons les différentes techniques utilisées actuellement pour la correction de ces défauts.

### 1.1.1 Des mesures sujettes à défauts

Le mode de mesure le plus couramment utilisé lors de la prise de mesure par AFM est le mode de mesure intermittent, décrit dans le paragraphe 1.A.2. Ce mode de mesure, qui est le plus largement utilisé, engendre un certain nombre de défauts. Avant de les décrire plus amplement, voici plusieurs définitions nécessaires :

- **ligne de scan** : l'AFM prend les mesures en faisant des lignes d'une certaine longueur avec un pas régulier. Nous parlerons donc de lignes de scan pour désigner ces lignes de mesures. En réalité, il ne s'agit pas de lignes droites car le trajet suivi par la pointe décrit une sinusoïde du haut en bas de l'image avec une période égale à l'espace entre deux lignes.
- **image AFM** : on parlera d'image pour parler d'une mesure AFM. Comme les lignes de scan font toutes la même longueur, elles forment une matrice qui peut être représentée comme une image. L'AFM retourne une topologie de la surface et nous avons un nuage de points en trois dimensions. Si le mode de prise de mesure de la surface n'était pas en ligne à ligne avec un pas régulier entre les points et entre les lignes nous ne pourrions pas représenter les données sous forme d'image. L'information contenue dans l'image est bien une information 3D mais la représentation graphique est faite sous forme d'image grâce à cette propriété de la prise de mesure.

Les images AFM contiennent principalement deux défauts. Un défaut qui est toujours présent, de manière plus ou moins importante, est la *dérive*. En effet, la dérive implique que la hauteur de la surface n'est pas un plan constant.

Un autre type de défaut qui est moins courant est le *saut d'asservissement*. D'une ligne de scan à une autre on peut observer une « cassure » représentant une perte d'asservissement de la pointe.

---

1. AFM pour Atomic Force Microscope

### La dérive

La dérive est un défaut toujours présent dans une image AFM. La Figure 1.1 montre deux exemples de dérive dans des images.

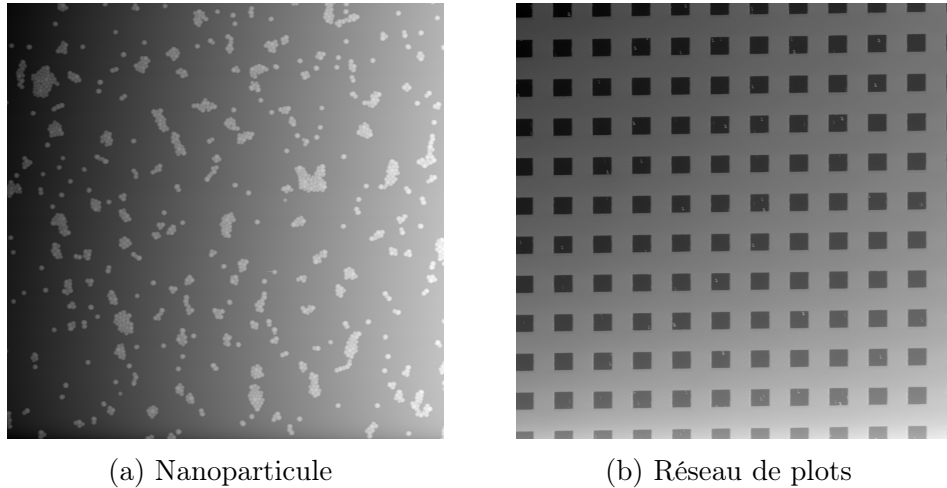


FIGURE 1.1 – Exemples d’images AFM brutes contenant une dérive

Chaque ligne semble subir une dérive linéaire, comme le montre la Figure 1.2. Cette dérive évolue au fur et à mesure des lignes de scan, c’est pourquoi, on peut observer sur l’image complète une dérive qui ne semble pas linéaire sur la globalité de l’image.

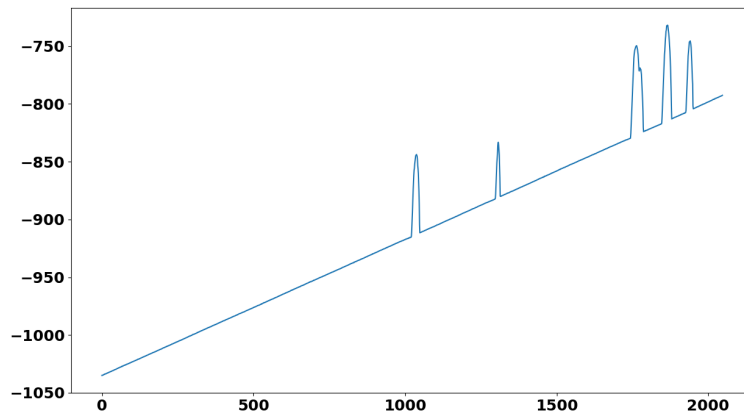


FIGURE 1.2 – Tracé de la ligne 150 de l’image 1.1a.

### Les sauts d'asservissement

Les sauts d'asservissement sont des « ruptures » dans les mesures mais nettes, en une ligne. On peut observer des exemples de sauts d'asservissement sur la Figure 1.3. Les images sont généralement continues, comme observées dans la section précédente. Un saut d'asservissement est une perte de continuité dans l'image, d'une ligne de scan à une autre.

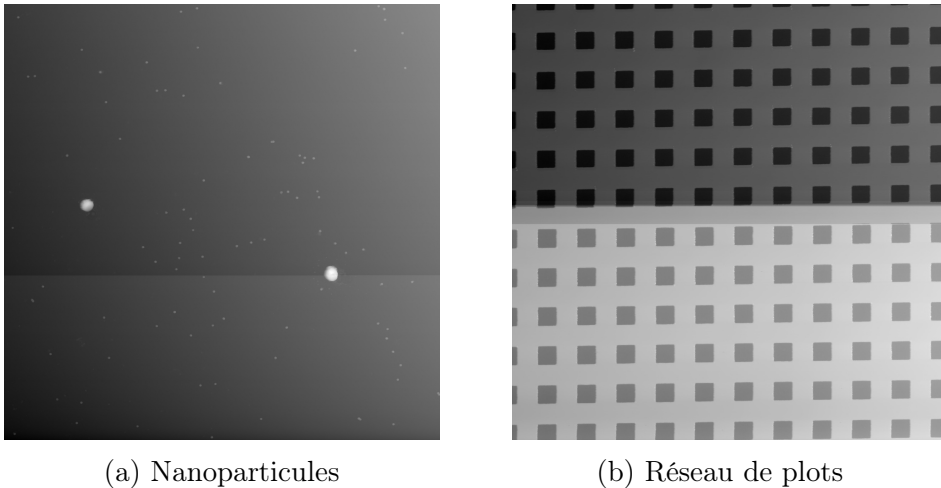


FIGURE 1.3 – Exemples de mesures contenant un saut d'asservissement

Cette perte de continuité dans l'image est due à une discontinuité dans la mesure de la hauteur, généralement lors du passage d'une ligne de scan à une autre. La Figure 1.4 contient la visualisation du saut d'asservissement de l'image 1.3b en 3D.

Les sauts d'asservissement sont généralement dûs à une contamination de la pointe ou bien à un défaut dans la prise des mesures (perte du signal lors du retour par exemple).



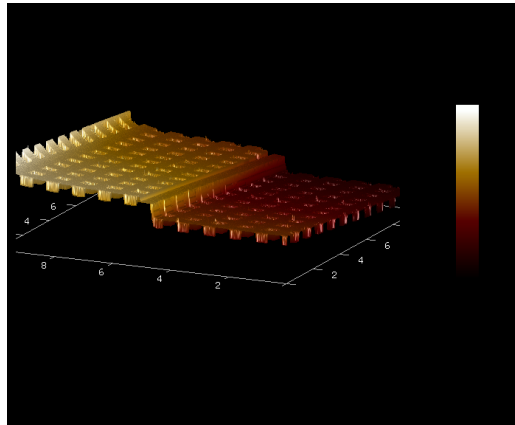


FIGURE 1.4 – Représentation de l’image 1.3b contenant un saut d’asservissement en 3D

### 1.1.2 Correction des défauts : logiciels existants

Dans cette section, nous décrivons brièvement les principaux logiciels utilisés pour le redressement d’images AFM.

#### Les logiciels de fabricants de microscopes AFM

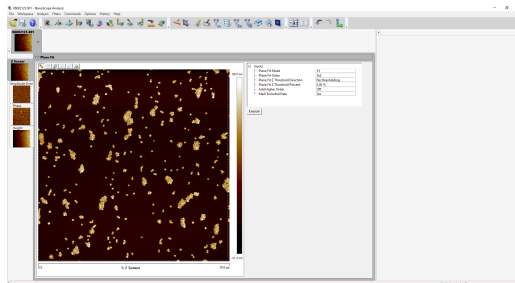
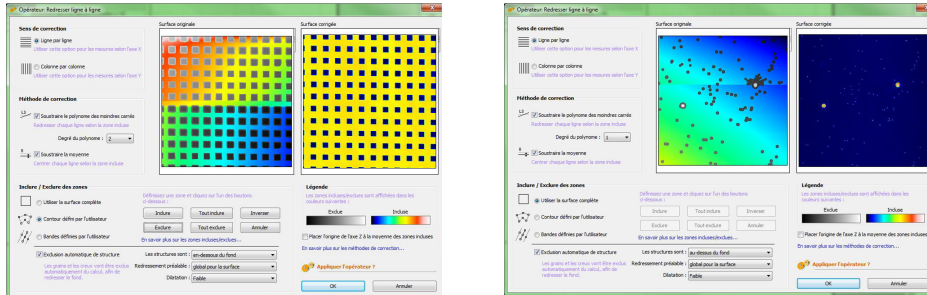


FIGURE 1.5 – Exemple d’une capture de la fenêtre de Nanoscope pour le redressement d’une image AFM

**Nanoscope** Nanoscope est un logiciel de Bruker spécialisé dans l’analyse d’images AFM. Le logiciel permet d’appliquer différents types de redressement : soit ligne à ligne soit par plan. Une capture d’écran est présentée sur la Figure 1.5. Nanoscope permet également de redresser l’image complète en prenant en compte uniquement certaines zones de l’image, définies par l’utilisateur.

Les logiciels commerciaux



(a) Redressement d'une image AFM de réseau de plots

(b) Redressement d'une image AFM de nanoparticules

FIGURE 1.6 – Exemples de captures de la fenêtre de MountainsMap pour le redressement d'images AFM ligne à ligne

**MountainsMap** Le logiciel MountainsMap est une collection de logiciels d'analyse de données météorologiques de la société Digital Surf. Pour le redressement AFM, le logiciel fonctionne de la façon suivante : l'utilisateur charge une image AFM, utilise l'outil de redressement. Dans cet outil, l'utilisateur peut choisir entre effectuer un redressement par plan ou bien ligne à ligne de son image AFM. Des exemples de captures d'écran de l'application de redressement ligne à ligne d'image AFM sont présentés dans la Figure 1.6. Le logiciel détecte les objets qui sont présents dans l'image avant d'effectuer le redressement, afin d'en être indépendant. Il applique une dilatation des zones d'exclusion afin de n'avoir aucun résidu d'objet restant éventuel. Il arrive que le logiciel ne trouve pas tous les objets présents dans l'image et l'utilisateur doit alors exclure les objets non détectés à la main. Une capture d'écran de l'outil d'exclusion des objets est présentée dans la Figure 1.7. Après avoir exclu les objets de manière automatique, en utilisant l'outil de détection d'objets, ou bien manuelle, l'utilisateur peut appliquer un redressement par plan ou bien par ligne en ajustant un polynôme de degré qu'il définit.

**SPIP** SPIP est un logiciel commercial de la société Image Metrology, permettant l'analyse de données météorologiques. Il permet de faire du redressement d'image AFM. Les captures d'écran de l'application de redressement sont présentées dans la Figure 1.8. L'utilisateur dispose d'un menu avec plusieurs options pour le redressement. Il peut choisir un redressement par plan

## 1.1. LES MESURES PAR MICROSCOPIE À FORCE ATOMIQUE

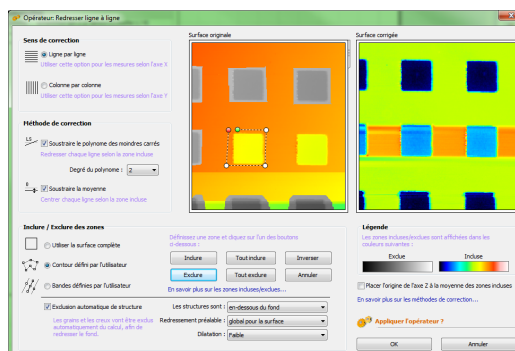
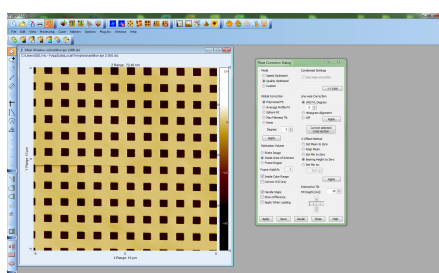
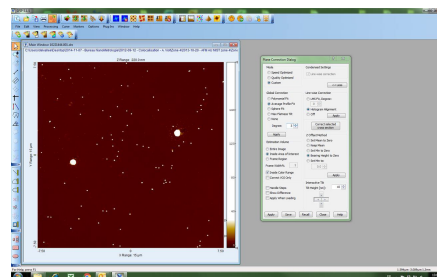


FIGURE 1.7 – Exemples d’une capture de la fenêtre de MountainsMap pour l’exclusion manuelle d’objets dans une image AFM



(a) Redressement d’une image AFM de réseau de plots

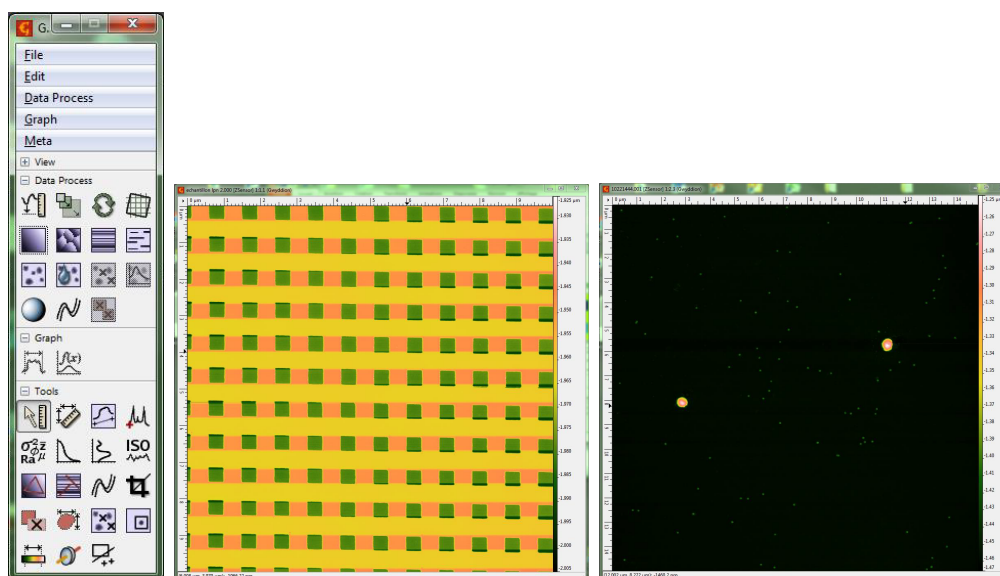


(b) Redressement d’une image AFM de nanoparticules

FIGURE 1.8 – Exemples de captures de la fenêtre de SPIP pour le redressement d’images AFM

ou par ligne en choisissant le type d'interpolation, afin de savoir comment est créée la surface à retirer.

## Les logiciels gratuits



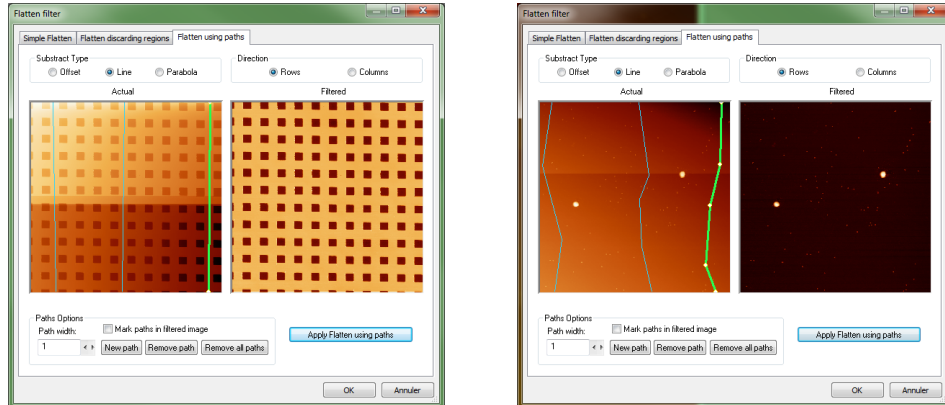
(a) Boîte à outils. (b) Redressement d'une image AFM de réseau de plots. (c) Redressement d'une image AFM de nanoparticules avec saut d'asservissement.

FIGURE 1.9 – Exemples de captures de la fenêtre de gwyddion pour le redressement d'images AFM.

**Gwyddion** Gwyddion est un logiciel libre qui permet d'analyser les données sortant de différents types de microscopes. Le logiciel contient une boîte à outils qui permet d'appliquer des opérations sur les données. Une capture d'écran est présentée sur la Figure 1.9. Pour le redressement, il applique un redressement par polynôme ligne à ligne sur l'image, ou bien par soustraction du plan médian.

**WSxM** WSxM est un logiciel qui permet d'analyser des données de microscopes. Pour le redressement AFM, il faut placer des « chemins » : une suite de segments traversant l'image de manière verticale, représentant les points qui seront pris en compte pour appliquer une interpolation afin de redresser l'image. Une capture d'écran est présentée sur la Figure 1.10.

## 1.2. CHOIX ET DÉFINITION DES CRITÈRES DE QUALITÉ



(a) Redressement d'une image AFM de réseau de plots

(b) Redressement d'une image AFM de nanoparticules

FIGURE 1.10 – Exemples de captures de la fenêtre de WsXM pour le redressement d'images AFM

### Limitations des méthodes actuelles

La limitation principale est dans le ratio entre automatisation et robustesse. Par robuste, nous entendons la stabilité des résultats en fonction des structures qui sont présentes ou encore la stabilité au bruit présent sur les images. Les logiciels qui fonctionnent de manière automatique ne sont pas forcément très robustes tandis que ceux qui le sont nécessitent beaucoup de réglages manuels par l'utilisateur (et donc le temps de traitement est relativement long). Comme par exemple la définition des objets de manière manuelle ou bien en devant tester différents degrés de polynôme avant de trouver le meilleur type de redressement. Il est à noter que les différentes étapes du traitement peuvent varier fortement en fonction du type de l'image, ce qui ne facilite pas le travail.

## 1.2 Choix et définition des critères de qualité

Nous venons de présenter les différentes méthodes classiques de redressement d'image AFM. Afin de comparer les méthodes, nous avons besoin de définir des critères objectifs. Dans cette section, nous allons présenter divers indicateurs permettant de mesurer la performance des procédures de redressement d'images AFM. Pour cela, nous allons d'abord définir les objectifs d'un redressement d'image AFM de qualité puis présenter les façons d'estimer si un redressement AFM respecte les objectifs ou non.

### 1.2.1 Objectifs

Dans le cadre de la correction des dérives d'images AFM, il est nécessaire de pouvoir évaluer et quantifier la qualité des résultats afin de comparer les méthodes et logiciels. Afin de définir des indicateurs pertinents, nous allons tout d'abord établir les caractéristiques attendues d'un redressement d'image AFM réussi.

Comme nous avons vu dans la section précédente, une image AFM n'est pas plane. Elle est composée d'un ensemble de lignes de scan qui contiennent chacune une dérive considérée comme linéaire. Afin de pouvoir mesurer simplement la hauteur des objets qui sont présents sur l'image AFM, le fond de l'image doit être le plus plat possible. En effet, on considère l'image comme un plan, contenant de la rugosité, avec des objets au dessus et/ou en dessous de ce plan.

Les objectifs du redressement d'images AFM sont les suivants :

- le fond de l'image, c'est-à-dire la surface, doit avoir une moyenne proche de 0 avec pour variabilité la rugosité du plan,
- les objets ne doivent pas être modifiés et doivent être au dessus et/ou en dessous du plan.

### 1.2.2 Les différents critères

Un critère de qualité a pour objectif de mettre un indice de confiance/-qualité sur un redressement d'image AFM. En effet, l'œil humain apporte toujours une part de subjectivité mais n'est pas toujours performant pour détecter les défauts. Le critère de qualité doit donc permettre de quantifier à quel point les critères de redressement sont respectés.

Pour cela, nous avons fait le choix de décrire l'image comme un mélange de plusieurs populations :

- La population des pixels liés au fond de l'image. Sur le plan théorique, chacun des pixels du fond suit une loi normale de moyenne nulle et d'écart-type égal à la rugosité,
- Les objets qui peuvent être modélisés suivant les cas par une ou un mélange de plusieurs populations suivant l'hétérogénéité des dimensions et de la position (au dessus ou en dessous du plan).

Ainsi, un premier critère de qualité qui pourrait être défini, après séparation du fond de l'image et des objets, serait l'étude des différentes populations de l'image AFM. Malheureusement, nous ne pouvons dire si les objets ont été modifiés ou non. Nous allons donc plutôt comparer l'image de départ et l'image d'arrivée, ce qui va permettre de ne pas avoir à séparer les objets du

fond de l'image.

### Critères basés sur l'image

L'objectif de ces critères est de tenir compte des différentes populations présentes dans l'image : objets et fond de l'image. Nous allons donc séparer les objets du fond sur une image redressée afin d'étudier ces populations. Nous allons prendre deux types de critère, un premier sur l'information globale de l'image et le second sur une analyse de chaque ligne.

Le principe du premier de ces critères est de segmenter au préalable l'image en considérant différents niveaux de valeurs, puis de calculer la variance de chacun de ces ensembles de niveau. Une solution pour faire la segmentation est d'appliquer un algorithme de K-moyennes, souvent utilisé en traitement d'image. Les Figures 1.11 et 1.12 montrent des résultats de segmentation sur image brute et image redressée.

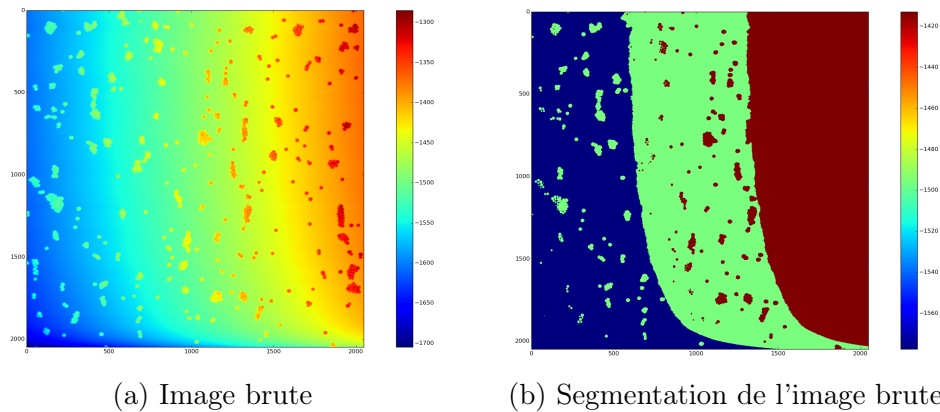


FIGURE 1.11 – Exemple d'image AFM de nanoparticules brute et de sa segmentation faite en appliquant un algorithme de K-moyennes sur les valeurs de hauteur avec 3 classes.

Soulignons que le point essentiel est de séparer au préalable les objets du fond. En effet, si on segmente comme indiqué dans la Figure 1.11b, nous n'obtenons pas le fond d'un côté et les objets de l'autre. Nous avons des portions complètes de l'images dans nos classes.

Dans le cas d'une image redressée et segmentée comme sur l'image 1.12b, la segmentation est représentative de l'image et les objets sont présents dans une classe et le fond dans une autre.

La segmentation est un élément critique. Les Figures 1.13 , 1.14 et 1.11 illustrent des cas où la segmentation faite par l'algorithme des K-moyennes.

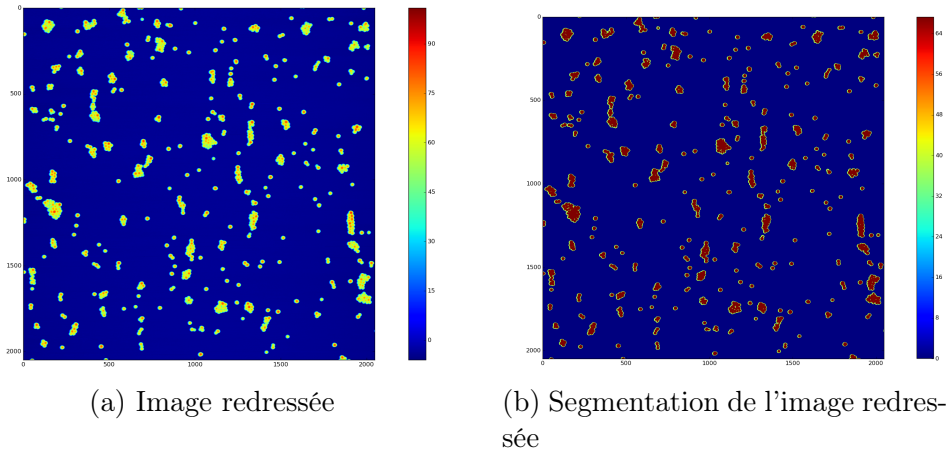


FIGURE 1.12 – Exemples de résultats d'image AFM redressée et de sa segmentation faite en appliquant un algorithme de K-moyennes sur les valeurs de hauteur avec 3 classes.

On voit très nettement sur ces exemples que la méthode est limitée. Ces résultats de segmentation ne permettent pas d'estimer la variance du fond de l'image. L'algorithme des K-moyennes nous permet d'avoir une segmentation dans le cas où l'image est redressée. Or nous souhaitons calculer le masque sur l'image brute, qui servira pour le calcul de critères pour la comparaison des méthodes de redressement.

Une autre solution pour la segmentation est d'utiliser une technique d'extraction d'objets dans une image. Dans notre cas nous nous sommes intéressés aux MSER (voir référence [Nistér and Stewénius \(2008\)](#)). La méthode permet d'extraire les objets qui sont présents. Les MSER permettent d'extraire des zones stables dans l'image, c'est-à-dire des zones caractéristiques importantes. Cela donne, dans le cas d'une image brute, avec ou sans saut d'asservissement, un meilleur résultat pour la création d'un masque, comme l'illustrent les Figures 1.13 et 1.14. Ainsi, nous sommes en mesure d'avoir un masque unique, de qualité, dans le cadre d'une comparaison inter-méthode sur une même image.

Dans le cas des Figures 1.14 et 1.13, nous voyons que la création d'un masque binaire par MSER pour l'étude des variances est plus justifiée qu'une extraction par K-moyennes, qui suppose que le redressement est correct. Le défaut de la technique de l'extraction du masque par MSER est que nous n'avons pas d'information sur les objets par rapport à la surface. Cette technique n'est pas 100% robuste mais permet d'obtenir un masque correct dans la plupart des cas d'utilisation.



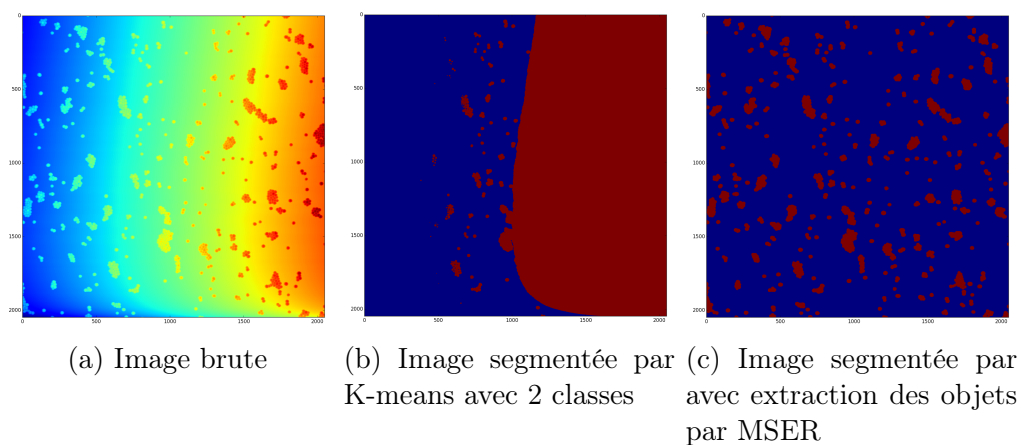


FIGURE 1.13 – Exemples de segmentation sur une image non redressée.

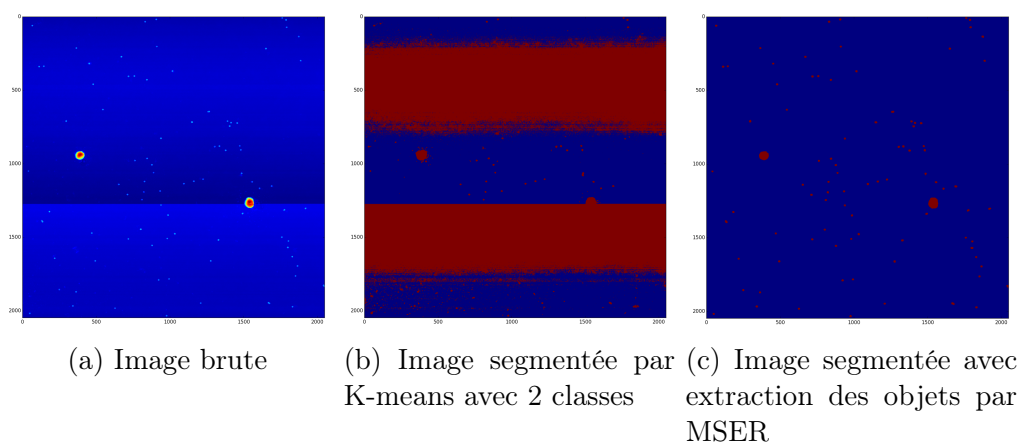


FIGURE 1.14 – Exemples de segmentation sur une image mal redressée contenant un saut d'asservissement.

La variance de la population du fond de l'image  $\sigma_F^2$  nous permet d'avoir une valeur représentative de la planéité de l'image. Cette valeur permet d'établir un premier critère de comparaison, plus la valeur de  $\sigma_F^2$  sera faible, meilleur sera le résultat. La borne inférieure pour  $\sigma_F^2$  est la rugosité du substrat. Donc plus un redressement sera bien réalisé et plus la valeur de  $\sigma_F^2$  convergera vers la valeur de rugosité du substrat. Si quelques objets ne sont pas détectés par l'algorithme des MSER, la valeur de  $\sigma_F^2$  la plus faible continuera de représenter le meilleur résultat.

Le redressement d'image AFM étant souvent effectué en redressant chaque ligne de manière indépendante, nous souhaitons avoir un critère qui vérifierait que les lignes sont bien toutes redressées.

Le critère précédent, ne permet pas d'observer si l'image contient une ligne mal redressée, comme par exemple dans le cas d'un saut d'asservissement. En effet, nous utilisons la globalité des points pour calculer nos valeurs et donc une minorité de points aberrants n'influe pas sur les valeurs.

Afin de combler ce manque, nous allons analyser la variance du fond de chaque ligne. Ainsi nous prendrons la valeur maximale de l'écart à la moyenne des variances du fond de chaque ligne, c'est-à-dire :

$$\sigma_{L_{max}}^2 = \max_l | \sigma_l^2 - \bar{\sigma}_l^2 |$$

où  $\sigma_l^2$  est la variance du fond de la ligne  $l$  et  $\bar{\sigma}_l^2$  la variance moyenne du fond des lignes. Le calcul du fond est fait en utilisant le masque calculé précédemment en utilisant la détection d'objets par MSER. Ainsi, plus cette valeur sera grande, plus le fond d'une ligne sera différent des autres, et montrera qu'une ligne est potentiellement mal redressée.

### Critère pour mettre en évidence la modification des objets

Le redressement ne doit pas affecter les objets : ceux-ci doivent avoir les mêmes dimensions avant et après. Afin d'évaluer si les objets sont modifiés, nous avons fait le choix d'étudier le gradient de l'image. Le redressement a pour objectif d'aligner le fond de l'image sur le plan  $Z = 0$  et ne doit pas modifier la rugosité ni les objets. Dans ce cadre, nous allons étudier la différence entre l'image brute et l'image redressée, que nous nommerons  $D$  par la suite. Comme le montre la Figure 1.15, la matrice  $D$  ne peut pas contenir de valeurs différentes du fond au niveau des objets, elle doit uniquement contenir la dérive de l'image d'origine.

Cependant, nous voulons observer si les objets ont subi des modifications. Pour cela, nous avons fait le choix d'étudier le gradient de l'image  $D$ . Le

## 1.2. CHOIX ET DÉFINITION DES CRITÈRES DE QUALITÉ

---

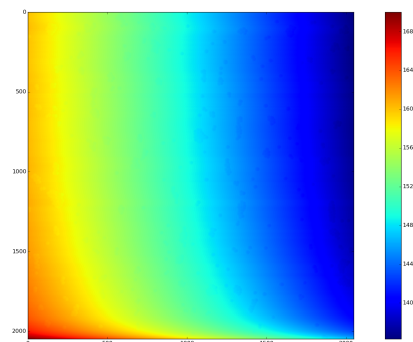


FIGURE 1.15 – Différence entre l'image brute, présentée sur la Figure 1.11a, et l'image redressée, présentée sur la Figure 1.12a.

gradient est une technique utilisée couramment en traitement d'image afin de détecter les contours des objets (voir le livre de Forsyth and Ponce (2002)). On observe sur la Figure 1.16 que le gradient de l'image brute et celui de l'image redressée se ressemblent fortement, la seule différence étant au niveau de la valeur moyenne du fond de l'image, générée par la dérive.

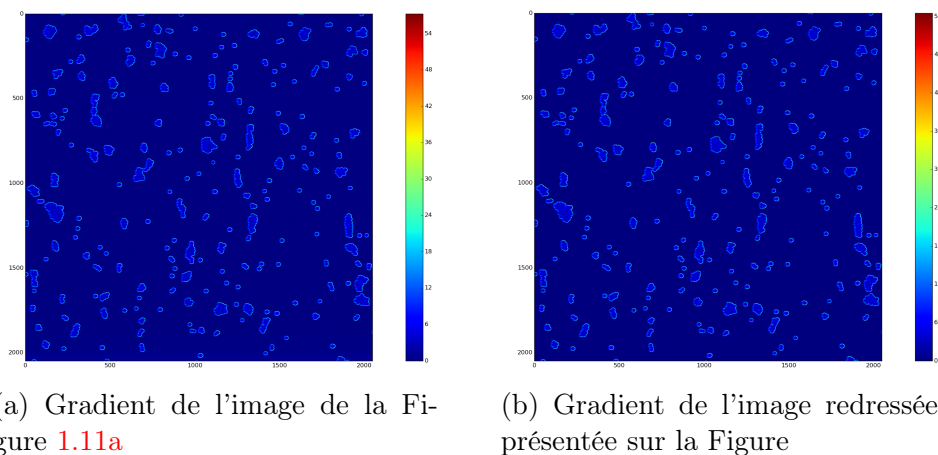


FIGURE 1.16 – Exemples de gradient de la même image AFM brute et redressée.

En effet, le calcul du gradient de l'image  $D$  peut être fait en calculant la différence entre les images des Figures 1.16b et 1.16a ou bien en calculant le gradient de l'image  $D$ , présentée dans la Figure 1.15. Le résultat du gradient de l'image  $D$  est présenté sur la Figure 1.17.

Dans notre cas, l'objectif est la vérification qu'il y a eu une modification

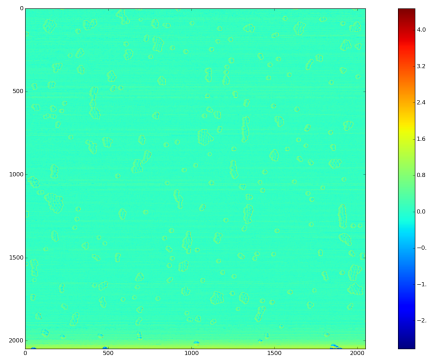


FIGURE 1.17 – Gradient de l'image de la Figure 1.15

globale qui n'impacte pas plus les objets que le fond. Si on analyse les variances entre le fond de l'image et la position des objets sur l'image  $D$ , en utilisant le masque calculé pour les critères précédents, et que nous observons que les variances ne sont pas similaires, cela impliquera une modification des objets. C'est pourquoi, nous définissons le rapport  $\rho$  comme :

$$\rho = \frac{\sigma_{GB}^2}{\sigma_G^2}$$

Où  $\sigma_G^2$  est la variance totale du gradient de  $D$  et  $\sigma_{GB}^2$  est la variance du fond du gradient de  $D$ . La population du fond est extraite en utilisant le masque calculé précédemment. Un exemple de gradient de l'image de différence, est présenté sur la Figure 1.17.

Le redressement ne devrait donc pas affecter plus les objets que le fond de l'image, on s'attend à ce que  $\sigma_{GB}^2$  soit proche de  $\sigma_G^2$ , ce qui revient à un  $\rho \approx 1$ .

### Résumé des critères

Afin de ne pas perdre le lecteur, nous allons faire un bref récapitulatif des différents indicateurs qui ont été sélectionnés, selon deux catégories :

1. La variance du fond de l'image  $\sigma_F^2$  : Pour calculer cette valeur, nous prenons la population qui représente le fond de l'image puis en extrayons sa variance. Cette population est extraite en calculant un masque par la technique MSER.
2. La variance de ligne extrême  $\sigma_{L_{max}}^2$  : représente la distance maximale entre la variance de population du fond de ligne et la moyenne de variance de population du fond de ligne. Nous distinguons la population

## 1.2. CHOIX ET DÉFINITION DES CRITÈRES DE QUALITÉ

---

du fond de ligne via le masque calculé précédemment en utilisant les MSER.

3. Le rapport de variances :

$$\rho = \frac{\sigma_{GB}^2}{\sigma_G^2}$$

Avec  $\sigma_G^2$  la variance totale et  $\sigma_{GB}^2$  variance du fond du gradient de  $D$ , où  $D$  est l'image qui à été retirée à l'image brute pour le redressement, calculée comme la différence entre l'image brute et l'image redressée. La population du fond est extraite par le même masque que pour les autres critères.

## 1.3 Redressement d'images AFM

Dans cette section nous allons présenter comment redresser une image AFM. Nous avons vu que les images AFM peuvent être redressées de façon globale, en estimant un plan polynomial puis en le soustrayant à l'image brute ou bien en estimant chaque ligne par un polynôme. L'estimation doit être faite en considérant uniquement la surface et non les objets qui sont présents dans l'image, qui doivent être considérés comme des valeurs aberrantes. C'est pourquoi nous allons dans un premier temps présenter des techniques d'estimations polynomiale classiques et robustes. Nous présenterons tout d'abord les moindres carrés, puis les M-estimateurs qui permettent une estimation robuste et enfin une variante les M-estimateurs asymétriques.

Dans un second temps, nous présentons comment corriger une image AFM à partir des méthodes d'estimation présentées. Nous présenterons alors le modèle que nous avons choisi pour représenter l'image puis nous expliquerons comment exploiter ces différentes méthodes pour redresser les images.

Le dernier point que nous aborderons sera la technique que nous avons choisi d'utiliser pour corriger la dérive des images AFM. Nous détaillerons comment nous estimons les coefficients puis comment nous corrigeons la dérive et les sauts d'asservissement en exploitant au mieux les caractéristiques des mesures AFM.

### 1.3.1 Estimation polynomiale classique et robuste

Afin d'estimer la dérive plus ou moins complexe à détecter et supprimer, il faut être en mesure d'interpoler les données par un polynôme.

Que le redressement soit en 2D (ajustement d'une droite par ligne de scan) ou bien que le redressement soit fait en 3D (ajustement d'un plan sur l'image globale), les techniques d'estimation que nous allons présenter seront les mêmes.

Nous présentons tout d'abord une méthode classique, les moindres carrés puis nous présenterons des méthodes robustes, les M-estimateurs ainsi qu'une variante, les M-estimateurs asymétriques.

### Estimation par moindres carrés

Soient  $N$  points  $t_i \in \mathbb{R}^p$ , où  $i \in \{1 \cdots N\}$ . L'objectif est de calculer la fonction  $f : \mathbb{R}^p \rightarrow \mathbb{R}$ , qui approche les données scalaires  $f_i$  aux positions  $t_i$ , au sens des moindres carrés, c'est-à-dire en minimisant la fonction  $E(f) = \sum_i \|f(t_i) - f_i\|^2$  par rapport à  $f$ .

On cherche la fonction  $f$  sous la forme d'un polynôme de degré  $d$ . Dans le cas d'un redressement 2D ce polynôme sera un polynôme d'une variable, tandis que dans le cas 3D où pour tout  $i$ ,  $t_i = (x_i, y_i) \in \mathbb{R}^2$ , ce polynôme sera un polynôme de deux variables.

Notons dans tous les cas  $\theta$  le vecteur des coefficients de ce polynôme. Nous sommes donc amenés à résoudre le problème d'optimisation suivant :

$$\hat{\theta} = \operatorname{argmin}_{\theta \in \mathbb{R}^q} \sum_i (f(t_i; \theta) - f_i)^2 \quad (1.1)$$

Par la suite, nous allons poser :  $f(t; \theta) = b(t)^\top \theta$  avec  $b(t) = (b_1(t), \dots, b_q(t))^\top$ , le vecteur de taille  $q$  représentant la base polynomiale de degré  $d$ .  $q$  représente la dimension de l'espace vectoriel des polynômes de degré inférieur ou égal à  $d$ .

Par exemple :

- **cas d'un polynôme de degré 3 dans  $\mathbb{R}$**  : Dans ce cas, nous avons  $p = 1$  et  $d = 3$ . On a alors  $t \in \mathbb{R}$  et  $f(t; \theta) = \theta_0 + \theta_1 \times t + \theta_2 \times t^2 + \theta_3 \times t^3$ . Ce qui nous donne  $q = 3$ ,  $b(t) = (1, t, t^2, t^3)^\top$  et  $\theta = (\theta_0, \theta_1, \theta_2, \theta_3)$
- **cas d'un polynôme de degré 2 dans  $\mathbb{R}^2$**  : Dans ce cas, nous avons  $p = 2$  et  $d = 2$ . Si on pose  $t = (x, y)$  alors  $f(t; \theta) = \theta_0 + \theta_1 \times x + \theta_2 \times y + \theta_3 \times xy + \theta_4 \times x^2 + \theta_5 \times y^2$ . Ce qui nous donne  $q = 5$ ,  $b(t) = (1, x, y, xy, x^2, y^2)^\top$  et  $\theta = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$

Le nombre d'éléments de la base polynomiale, c'est-à-dire la dimension de l'espace vectoriel des polynômes de degré inférieur ou égal à  $d$ , peut être calculé explicitement par  $q = \frac{(p+d)!}{d!p!}$ .

Rappelons que pour résoudre le problème (1.1) il suffit de trouver  $\theta$  tel que la valeur de la différentielle par rapport à  $\theta$  de la fonctionnelle  $\sum_i (f(t_i; \theta) - f_i)^2$  soit nulle.

C'est-à-dire, sous écriture matricielle :

$$\begin{aligned} \sum_i 2b(t_i) (b(t_i)^\top \theta - f_i) &= 0 \\ 2 \sum_i b(t_i) b(t_i)^\top \theta - b(t_i) f_i &= 0 \end{aligned}$$

Si on définit  $B$  comme étant la matrice de taille  $N \times q$ ,  $B = (b_1(t), \dots, b_q(t))^T$  et  $F$  le vecteur de taille  $N$ ,  $F = (f_1, \dots, f_N)^T$  on obtient alors le système :

$$\begin{aligned} BB^T \hat{\theta} - BF &= 0 \\ BB^T \hat{\theta} &= BF \\ \hat{\theta} &= B^\dagger F \end{aligned}$$

avec  $B^\dagger$  qui désigne la matrice pseudo-inverse de  $B$ . On peut ainsi calculer les coefficients du polynôme qui interpole le mieux les données au sens des moindres carrés.

La technique des moindres carrés produit un estimateur intéressant mais peu robuste, comme le montrent les travaux de [Huber \(1981\)](#). Ceci pose problème lorsqu'on souhaite l'appliquer au redressement AFM, car les objets, que l'on ne souhaite pas détecter dans un premier temps, sont considérés comme des valeurs aberrantes de notre modèle de régression.

La section suivante envisage des alternatives robustes, c'est-à-dire dont les résultats ne sont pas affectés par des valeurs aberrantes.

### Estimation par M-estimateurs

Les M-estimateurs ont été introduits par [Huber \(1981\)](#). On trouve également une définition des M-estimateurs dans les travaux de [Hampel et al. \(1986\)](#) et [van der Vaart \(2000\)](#) pour une étude asymptotique.

Dans l'application visée, nous souhaitons interpoler le fond de l'image en essayant autant que possible de ne pas utiliser la position des objets au dessus ou en dessous de la dérive linéaire. Les M-estimateurs sont des outils qui nous permettent d'interpoler les données sans prendre en compte ces outliers ou du moins, dans une bien moindre mesure que les moindres carrés classiques, ce que nous allons décrire plus amplement. Le graphique de la [Figure 1.18](#), montre que l'interpolation par M-estimateurs n'est pas perturbée par la présence d'objets (considérés comme des outliers), contrairement à l'interpolation par moindre carrés, sur une ligne de scan présentant 5 objets au dessus de la dérive linéaire.

La problématique est la suivante : soit  $t_i \in \mathbb{R}^p$ , où  $i \in \{1 \dots N\}$ . L'objectif est de calculer, cette fois ci de manière robuste, la fonction  $f$ , qui approche les données scalaires  $f_i$  aux positions  $t_i$ .

Afin de calculer la fonction  $f(t_i; \theta)$ , qui est ici supposée être une fonction polynomiale de degré  $d$  de coefficients  $\theta$ , nous cherchons à résoudre le



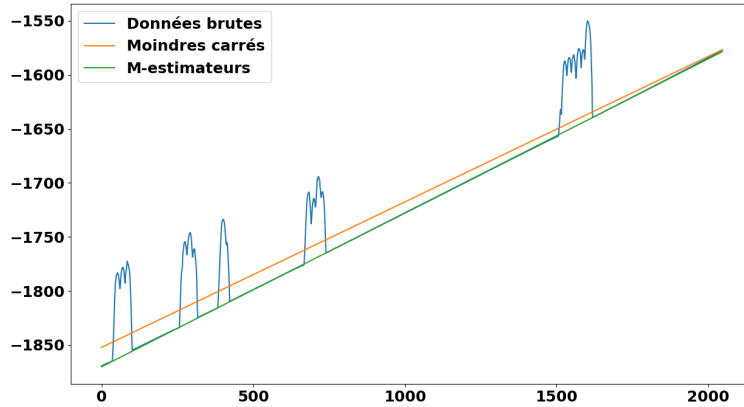


FIGURE 1.18 – Exemple d'interpolation d'une ligne de l'image AFM 1.11a par M-estimateurs et moindre carrés.

problème suivant :

$$\hat{\theta} = \operatorname{argmin}_{\theta \in \mathbb{R}^q} \sum_i \rho(f(t_i; \theta), f_i) \quad (1.2)$$

où  $\rho : \mathbb{R} \rightarrow \mathbb{R}$  est une certaine fonction de coût, supposée  $\mathcal{C}^1$ . On définit ainsi ce qu'on appelle un M-estimateur. Afin de résoudre ce problème, il faut utiliser la fonction  $\psi$  qui est la dérivée de  $\rho$  par rapport à  $\theta$ . Ainsi, pour trouver une solution à l'équation 1.2, il faut résoudre l'équation dite estimante :

$$\sum_i \psi(f(t_i; \theta); f_i) = 0 \quad (1.3)$$

On obtient ainsi toute une classe d'estimateurs dépendant de la fonction  $\psi$  choisie. L'estimateur le plus couramment utilisé est l'estimateur biweight de Tukey pour lequel :

$$\psi(x) = \begin{cases} x \left(1 - \frac{x^2}{c^2}\right)^2 & \text{si } |x| \leq c \\ 0 & \text{sinon} \end{cases} \quad (1.4)$$

Un exemple de cette fonction est tracé sur la Figure 1.19. D'autres estimateurs tel que l'estimateur de Huber ou encore l'estimateur de Andrew sont largement utilisés, voir Andrews et al. (1972). Le choix de la fonction  $\psi$  est souvent calibrée en fonction des propriétés du M-estimateur que l'on souhaite avoir.

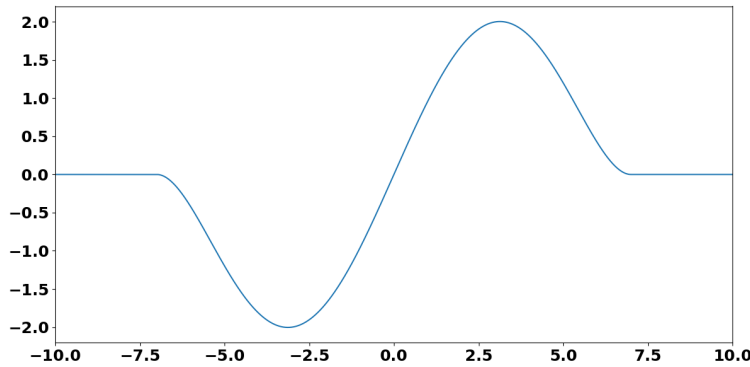


FIGURE 1.19 – Exemple de la fonction  $\psi$  biweight de Tukey (l'équation est présentée en (1.4) avec  $c = 7$ ).

Afin de résoudre ce problème, on reformule l'équation (1.3) en :

$$\sum_i w_i (f(t_i; \theta) - f_i) = 0 \quad \text{avec} \quad w_i = \frac{\psi(f(t_i; \theta); f_i)}{(f(t_i; \theta); f_i)}$$

Une solution approchée de ce problème se trouve par moindres carrés itératifs pondérés<sup>2</sup> (cf les travaux de [Green \(1984\)](#) et ceux de [Wolke \(1992\)](#); [Wolke and Schwetlick \(1988\)](#)). Le principe des IRLS, est de faire une première estimation par moindres carrés. Dans un second temps, on calcule un vecteur de poids, en appliquant la fonction  $\psi$  sur les résidus de l'étape précédente. Nous appliquons ensuite une interpolation par moindres carrés pondérés afin d'estimer les coefficients. On recalcule ensuite les résidus et on recommence les interpolations par moindres carrés pondérés jusqu'à convergence. La convergence peut être mesurée de différentes façons, soit on prend une convergence sur les coefficients calculés soit sur la norme du vecteur de poids.

Afin d'avoir une implémentation générique, nous devons être en mesure de fixer les poids pour la fonction  $\psi$ . Pour cela, nous suivons la recommandation de la fonction `r1m` du paquet MASS de R, décrit dans le livre de [Venables and Ripley \(2002\)](#), qui propose de normaliser les résidus en les divisant par la médiane en valeur absolue divisée par 0,6745. Cette dernière valeur est la médiane en valeur absolue d'une gaussienne de moyenne 0 et d'écart-type 1.

---

2. Iteratively reweighted least squares (IRLS)

### Estimation par M-estimateurs asymétriques

Le principe des M-estimateurs asymétriques est le suivant : il s'agit de modifier la fonction de poids afin qu'elle soit asymétrique pour donner plus d'importance à un côté des données plutôt qu'à l'autre (voir la publication de [Allende et al. \(2006\)](#)).

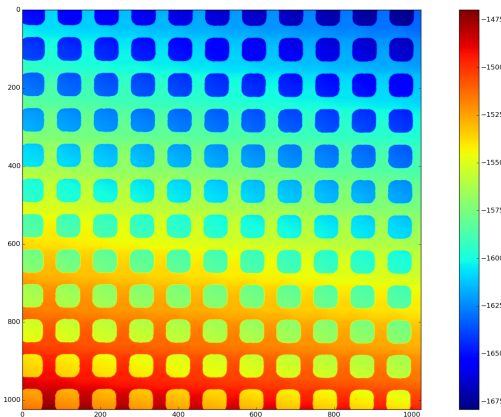


FIGURE 1.20 – Exemples d'image brute AFM de plot.

Si nous prenons l'image 1.20, les objets sont les plots qui sont des trous dans la surface. Si nous prenons une ligne avec des trous, la population de trous étant sensiblement la même que celle de la surface, les M-estimateurs classiques donneront un résultat sensiblement similaire aux moindres carrés, comme le montre la Figure 1.21. Ceci est dû au fait que si les plots sont considérés comme des outliers, ils vont produire des signaux (lignes de scan par exemple) dont les valeurs seront bien inférieures à la dérive linéaire. Les plots (points aberrants) étant répartis d'un unique côté de la surface, si nous appliquons une fonction  $\psi$  asymétrique, nous parvenons à donner plus d'importance à la surface qu'aux trous, et parvenons à interpoler la surface. Nous pouvons observer le résultat sur l'image 1.21.

La version asymétrique des M-estimateurs consiste en l'utilisation d'une fonction  $\psi$  asymétrique, voir les publications de [Allende et al. \(2006\)](#) et [Andrews et al. \(1972\)](#). Par exemple, la fonction de Tukey, présentée en (1.4),

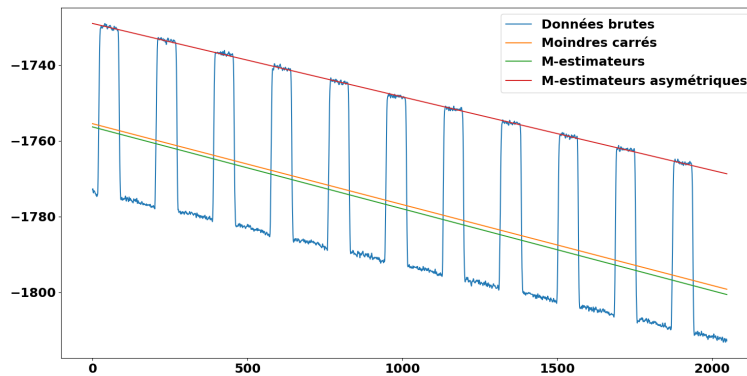


FIGURE 1.21 – Exemple de résultats d'interpolations sur une ligne de l'image 1.20, contenant des plots.

possède la version asymétrique suivante :

$$\psi(x) = \begin{cases} x \left(1 - \frac{x^2}{c_1^2}\right)^2 & \text{si } -c_1 \leq x \leq 0 \\ x \left(1 - \frac{x^2}{c_2^2}\right)^2 & \text{si } 0 \leq x \leq c_2 \\ 0 & \text{sinon} \end{cases} \quad (1.5)$$

Un exemple de tracé d'une fonction de Tukey asymétrique est présent sur la Figure 1.22.

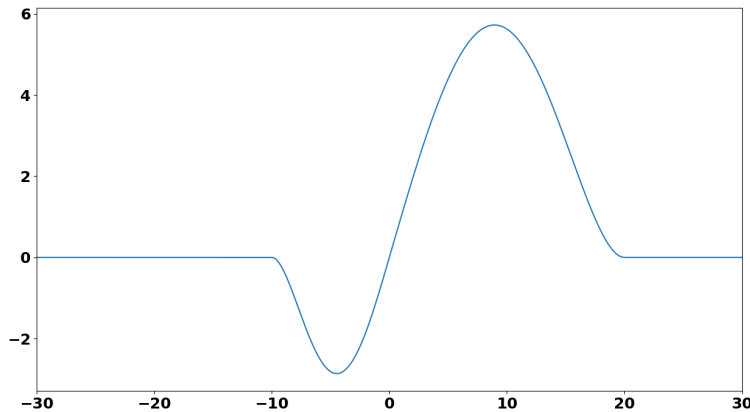


FIGURE 1.22 – Fonction biweight de Tukey asymétrique (équation (1.5) avec  $c_1 = 10$  et  $c_2 = 20$ ).

Dans ce cas, la résolution du problème, en pratique, se fait par IRLS, comme pour les M-estimateurs symétriques. Dans la configuration asymétrique, la difficulté vient de l'estimation de l'asymétrie des poids. Dans notre cas, nous avons appliqué une asymétrie avec un facteur 10 entre les poids, afin de ne pas prendre en compte les points que nous ne souhaitons pas interpoler.

### 1.3.2 Méthodes de redressement

Nous présentons ici l'application des méthodes précédentes pour l'estimation de la dérive. Nous pouvons estimer la dérive soit ligne par ligne soit en estimant la surface par un plan polynomial. Nous présentons ici les modèles mis au point pour les images AFM en fonction du type de redressement.

#### Redressement ligne à ligne

Dans le redressement ligne à ligne, nous considérons chaque ligne de scan comme étant indépendante et contenant une pente linéaire à supprimer. La Figure 1.23 montre un exemple d'une ligne de scan. On voit très nettement que la ligne contient une dérive linéaire.

Dans le cas de ce redressement, le modèle choisi est le suivant :

$$Z_{i,j} - H(x_j, y_i) = \theta_{i,0} + \theta_{i,1}x_j + \varepsilon_{i,j} \quad (1.6)$$

Pour  $i = 1, \dots, n_I$ ,  $j = 1, \dots, n_J$  où  $n_I$  et  $n_J$  sont respectivement le nombre de lignes et le nombre de colonnes de la matrice image.  $Z_{i,j}$  est l'observation de la  $i$ ème ligne et  $j$ ème colonne,  $x_j$  est l'abscisse,  $\theta_{i,0}$  et  $\theta_{i,1}$  sont les deux paramètres réels qui varient d'une ligne à l'autre,  $H(x_j, y_i)$  est la hauteur des objets (positif si les objets sont sur la surface et négatif si les objets sont en dessous) à la position  $(x_j, y_i)$ .

Enfin,  $\varepsilon_{i,j}$  représente les variations aléatoires, qui sont supposées centrées, indépendantes et identiquement distribuées.

Dans la pratique, cela revient à estimer la dérive ligne à ligne par un polynôme de degré 1.

Si pour chacune des lignes de scan de l'image 1.23a, on estime la tendance linéaire en utilisant une méthode robuste, on diminue le saut, comme le montre l'image de la Figure 1.24.

En revanche, si on applique le redressement ligne à ligne mais que les objets sont pris en compte, nous obtenons le résultat présenté dans la Figure 1.25.

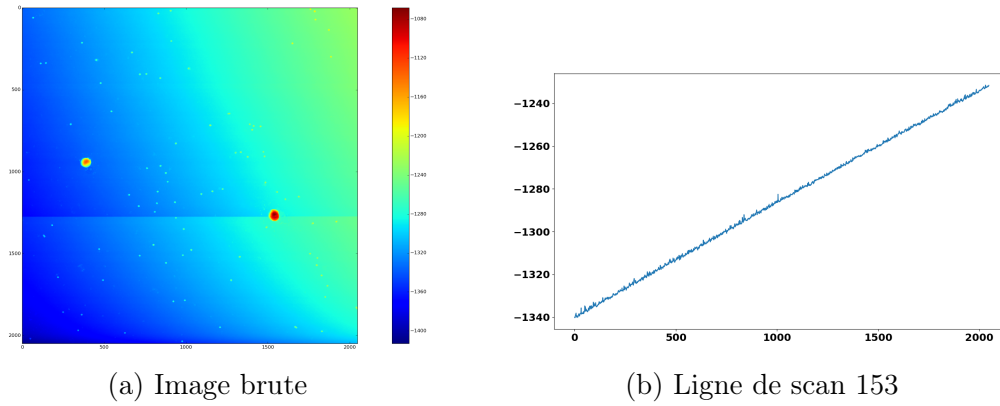


FIGURE 1.23 – Exemple d'une image et d'une ligne de scan de l'image.

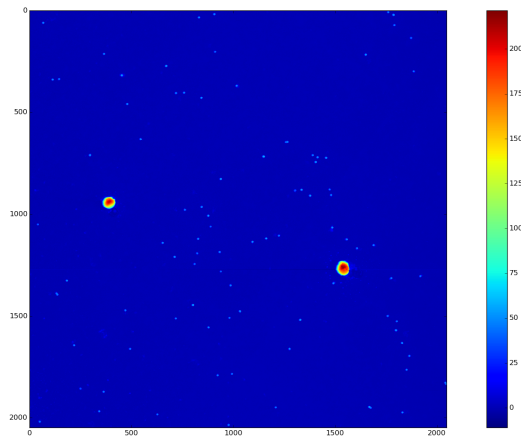


FIGURE 1.24 – Exemple du redressement ligne à ligne par un polynôme de degré 1, estimé par M-estimateur, de l'image 1.23a.

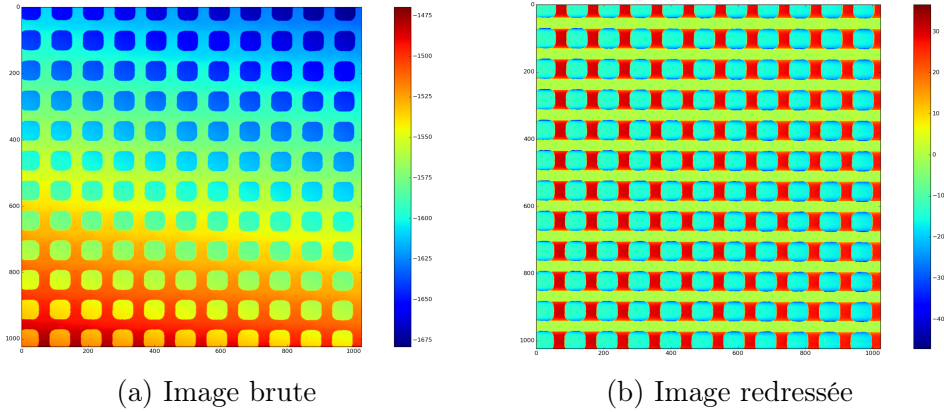


FIGURE 1.25 – Exemple d'un mauvais redressement d'image ligne à ligne par un polynôme de degré 1, estimé par M-estimateur.

Afin de redresser une image qui ne contient pas de saut de ligne, comme par exemple l'image 1.25a, un redressement global peut suffire.

### Redressement global

En suivant la stratégie adoptée par plusieurs logiciels, le redressement global consiste en une interpolation de la surface par un plan polynomial de degré 1 à 3. Comme le montre l'image de la Figure 1.26, on modélise le problème en considérant des objets sur une surface polynomiale. Dans cette approche on ne supprime pas les objets qui sont considérés comme des valeurs aberrantes. Sur des images ne comportant pas de saut d'asservissement, le résultat semble relativement satisfaisant, comme le montrent les images de la Figure 1.27. Si l'on formalise ceci en termes mathématiques, on cherche à estimer notre surface comme une surface polynomiale de degré  $d$  :

$$Z_{i,j} - H(x_j, y_i) = \theta \cdot b(t_{i,j}) + \varepsilon_{i,j} = \sum_{k=0}^q \theta_k b_k(t_{i,j}) + \varepsilon_{i,j}$$

Où  $Z_{i,j}$  sont les observations,  $\theta$  représente les coefficients polynomiaux,  $t_{i,j} = (x_j, y_i)$  où  $x_j$  est la position en abscisse,  $y_i$  la position en ordonnée et  $b(t)$  la base polynomiale définie dans la section 1.3.1.  $H(x_j, y_i)$  est la la hauteur des objets (positif si les objets sont sur la surface et négatif si les objets sont en dessous) à la position  $(x_j, y_i)$ .

Enfin,  $\varepsilon_{i,j}$  représente les variation aléatoires, qui sont supposée centrées, indépendantes et identiquement distribuées.

Si on pose  $\theta_k = \theta_{k_1, k_2}$  et  $b_k(t_{i,j}) = x_j^{k_1} y_i^{k_2}$  où  $k_2$  et  $k_1$  sont respectivement la partie entière et le reste de la division euclidienne de  $k$  par  $d$ . On a alors :

$$Z_{i,j} - H(x_j, y_i) = \sum_{k_1=0}^d \sum_{k_2=k_1}^d \theta_{k_1, k_2} x_j^{k_1} y_i^{k_2} + \varepsilon_{i,j}$$

Où  $\theta_{k_1, k_2}$  sont les coefficients polynomiaux,  $x_j$  est la position en abscisse,  $y_i$  la position en ordonnée.

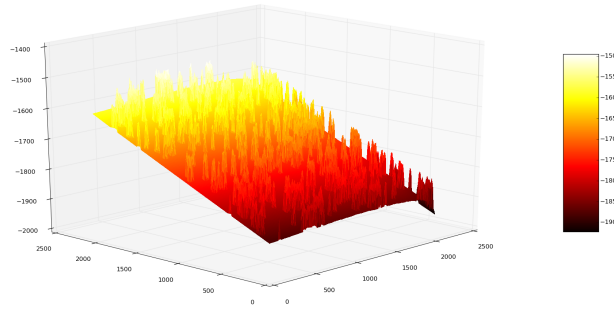
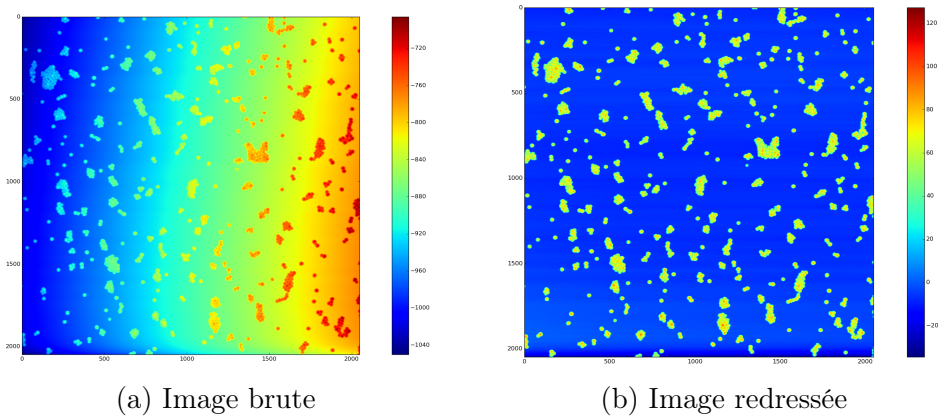


FIGURE 1.26 – Exemple de rendu en 3D de l'image AFM 1.11a.



La technique du redressement global doit être robuste afin de ne pas prendre en compte les objets présents dans l'image. Dans le cas de l'image 1.27c, les objets sont équivalents en taille et répartis de manière homogène sur la surface. C'est pourquoi un redressement polynomial par moindres carrés fonctionne bien, dans ce cas.

Dans le cas de l'image 1.27a, il faut appliquer une technique d'interpolation robuste afin de ne pas prendre les objets en compte. En effet, la non homogénéité des objets, en position et taille, peut avoir une influence sur le



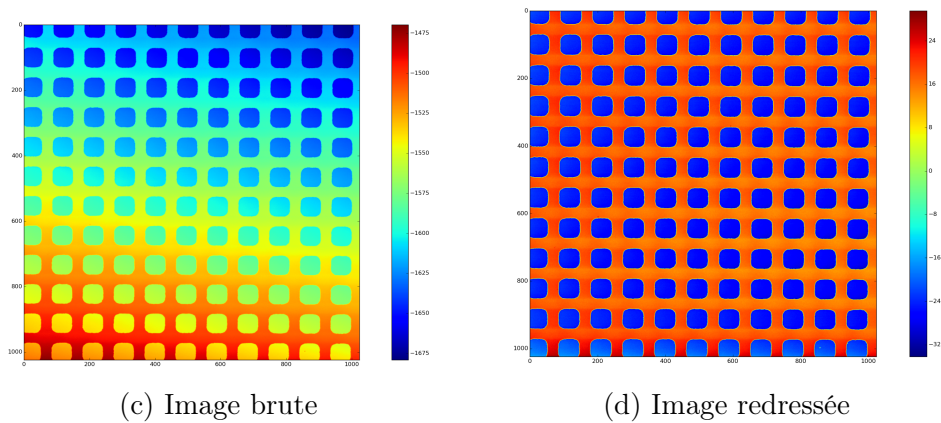


FIGURE 1.27 – Exemples d'un redressement d'image globale par une surface polynomiale de degré 3 estimée par moindre carrés.

redressement alors que l'objectif est d'interpoler uniquement la surface du fond.

### 1.3.3 Utilisation de l'information globale

Dans ce paragraphe nous allons expliquer comment améliorer le redressement d'image AFM ligne à ligne, en utilisant l'information globale.

Dans un premier temps nous étudierons les coefficients polynomiaux extraits en fonction de la ligne afin de montrer que ces derniers suivent une certaine tendance, c'est-à-dire qu'on observe une certaine continuité. Nous avons alors choisi d'appliquer un lissage des coefficients afin que chaque valeur de coefficient ne soit pas totalement indépendant mais tienne compte du voisinage. Pour effectuer ce lissage, nous verrons que les saut d'asservissement sont un obstacle au lissage, c'est pourquoi nous devons dans un premier temps les détecter afin d'effectuer le lissage sans les prendre en compte.

Nous terminerons avec la description d'une technique employée pour automatiser les M-estimateurs asymétriques sur le plan pratique.

#### Étude des coefficients extraits

Dans le cadre du redressement ligne à ligne, nous considérons chaque ligne comme un processus aléatoire indépendant, or ce n'est pas le cas.

En effet, si nous appliquons un redressement linéaire ligne à ligne l'image de la Figure 1.28 tout en conservant les coefficients directeurs et les ordonnées à l'origine pour chaque ligne. Puis que nous les traçons en fonction des lignes, nous obtenons les tracés de la Figure 1.29.

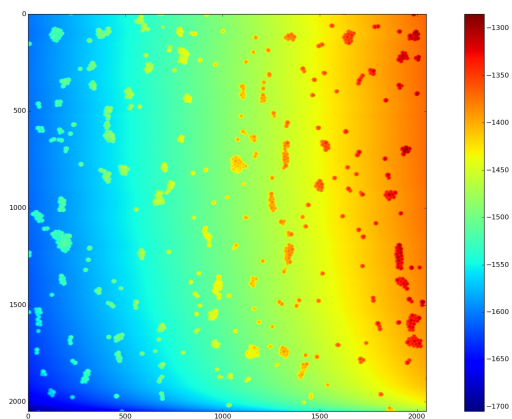
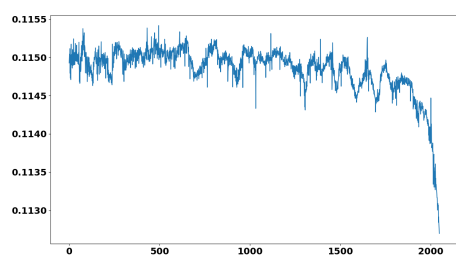
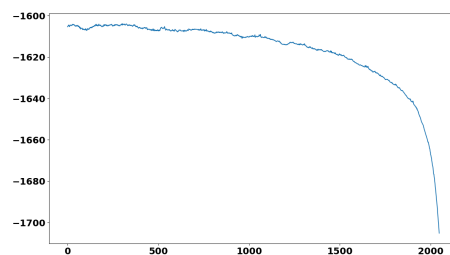


FIGURE 1.28 – Exemple d'image AFM brute de nanoparticules.



(a) Coefficients directeurs



(b) Ordonnées à l'origine

FIGURE 1.29 – Les coefficients polynomiaux, de l'image 1.28 extraits par interpolation robuste puis tracés en fonction des lignes.

### 1.3. REDRESSEMENT D'IMAGES AFM

---

Les coefficients directeurs [1.29a](#) et les ordonnées à l'origine de l'image [1.29b](#) forment des courbes qui semblent continues. Ce phénomène apparaît car la machine balaie la surface et que cette surface est continue. La dérive étant proche sur toute la surface, on observe une certaine dépendance entre les lignes.

Un autre argument plus intuitif est que l'ensemble des lignes de scan forment une image que l'on peut observer. Or, si les lignes étaient réellement indépendantes et les dérives complètement aléatoires d'une ligne à une autre alors l'ensemble des lignes ne formerait pas une surface continue. Cette continuité prouve qu'il existe un lien d'une ligne à l'autre.

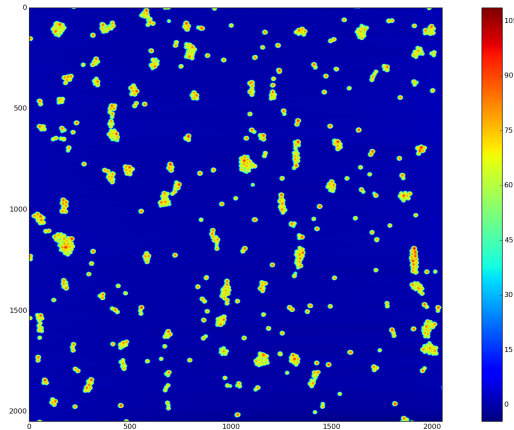


FIGURE 1.30 – Résultat du redressement de l'image [1.28](#).

Afin d'améliorer les résultats, il faudrait appliquer une optimisation des coefficients de la tendance linéaire post-interpolation afin de corriger les défauts de cette dernière. Par exemple imposer une contrainte de continuité ou bien lisser la courbe des coefficients.

Cette méthode pose problème dans le cas où il y a un saut d'asservissement sur la surface. Comme on peut le voir sur la Figure [1.31](#), les coefficients ne sont pas continus. On observe au niveau du saut d'asservissement que la liste des coefficients directeurs, voir image [1.31c](#), ont un pic au niveau du saut et la liste des ordonnées à l'origine, voir image [1.31d](#), ont une rupture, qui représente bien le saut.

La valeur aberrante dans les coefficients directeurs est générée par un saut d'asservissement qui a lieu au cours d'un scan de ligne et non lors d'un retour. Sur l'image résultat, la ligne en question doit donc être supprimée ou

bien remplacée par interpolation des lignes voisines, car la ligne incriminée contient de l'erreur et n'est donc pas pertinente pour la mesure.

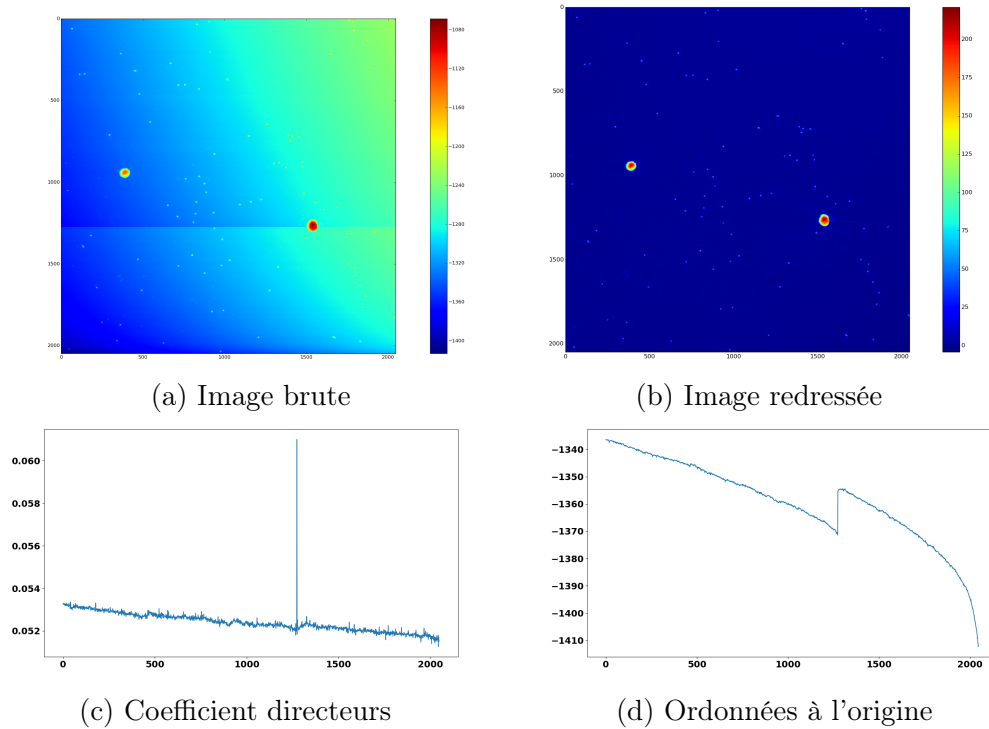


FIGURE 1.31 – Exemple de redressement linéaire ligne à ligne sur une image contenant un saut d'asservissement et tracé des coefficients directeurs et ordonnées à l'origine en fonction des lignes.

L'amélioration de l'interpolation se fait alors par lissage des coefficients directeurs et des ordonnées à l'origine tout en tenant compte du fait qu'il peut y avoir des sauts.

### Lissage des coefficients

La courbe des coefficients suivant une certaine tendance, nous allons essayer d'améliorer l'interpolation en appliquant un lissage des coefficients afin de prendre en compte l'information des lignes voisines, dans le but d'éliminer tout bruit résiduel.

L'étape de lissage des coefficients contient plusieurs étapes. Dans un premier temps, nous devons détecter les sauts. La détection des sauts nous permettra d'obtenir les lignes de l'image AFM d'origine qui contiennent un saut et qui doivent être remplacées. La technique doit également être en mesure de ne pas retourner de saut s'il n'y en a pas.

### 1.3. REDRESSEMENT D'IMAGES AFM

---

Ensuite vient l'étape de lissage des coefficients qui contient trois étapes. Dans un premier temps, supprimer les sauts dans les coefficients, s'il y a présence de saut. Dans un second temps, appliquer un lissage aux coefficients nettoyés des sauts. Puis enfin, remettre le saut en place.

**Détection des sauts d'asservissement** Afin de détecter les sauts d'asservissement, nous passons par l'analyse de la liste des ordonnées à l'origine qui sont extraites. Comme l'illustre la Figure 1.32, nous observons au niveau de la ligne qui contient le saut d'asservissement, que les coefficients directeurs (graphique 1.32b) contiennent un unique pic alors que sur les ordonnées à l'origine (graphique 1.32c) on observe une réelle marche sur les données.

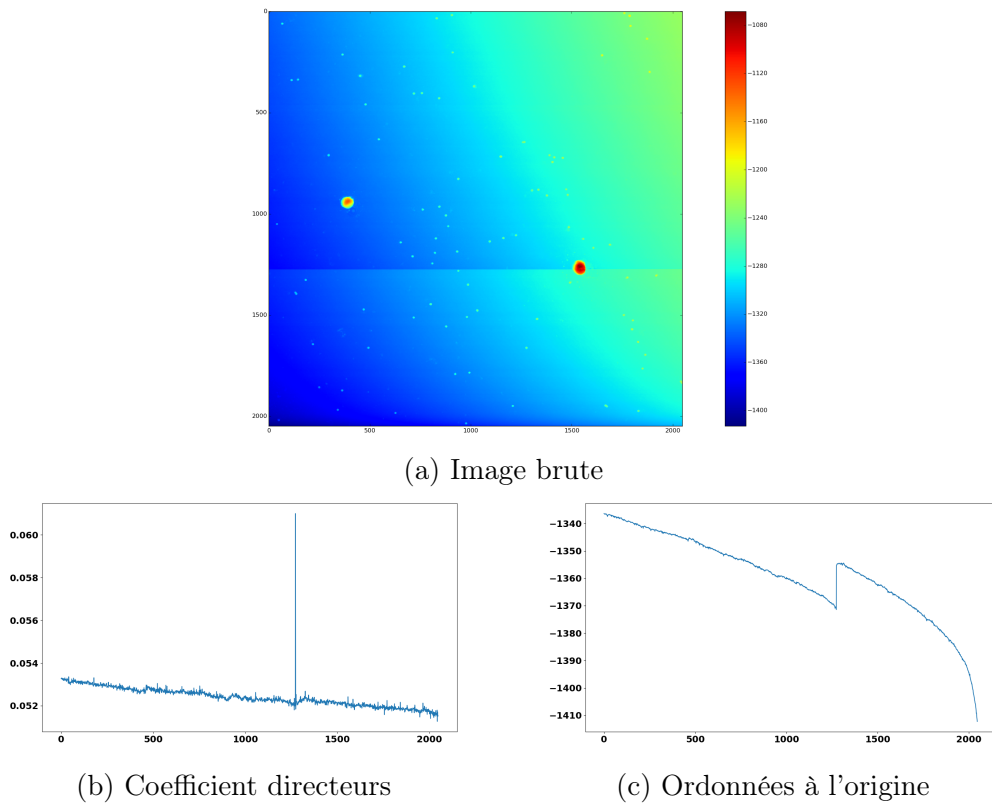


FIGURE 1.32 – Exemple d'image contenant un saut d'asservissement et tracé des coefficients polynomiaux en fonction des lignes.

Afin de détecter le saut, nous nous sommes dans un premier temps intéressés à une recherche de points aberrants sur le gradient des ordonnées à l'origine. Nous supposons les points du gradients suivant une distribution

gaussienne, afin d'écartier les points aberrants. Malheureusement, cette technique ne fonctionna pas car la dérive n'est pas linéaire et donc la liste des valeurs du gradient des coefficients ne sera pas une population qui oscille autour d'une valeur moyenne.

Nous nous sommes ensuite intéressés à l'article de [Ma and Yang \(2011\)](#), décrivant une technique d'estimation de positions de saut dans une courbe en analysant les nœuds d'une interpolation des coefficients par splines.

Dans un premier temps, nous appliquons une interpolation par spline constante ou linéaire par moindres carrés. Nous posons  $\hat{m}_1$  les nœuds de l'interpolation par splines constantes et  $\hat{m}_2$  les nœuds de l'interpolation par splines linéaires.

Pour détecter les sauts dans  $m$ , nous allons poser les hypothèses  $H_0 : m \in \mathcal{C}^{(p)} [0, 1]$  contre  $H_1 : m \notin \mathcal{C} [0, 1]$ , avec  $\mathcal{C}^{(p)}$  qui décrit l'espace des fonctions de régularités  $p = 1, 2$ .

- Pour  $p = 1$ , c'est-à-dire lors d'une interpolation par splines constantes, on définit le test statistique :

$$T_{1,j} = \frac{|\hat{m}_1(j+1) - \hat{m}_1(j)|}{\sigma_1}$$

$\sigma_1^2 = \frac{2\hat{\sigma}^2}{nh}$  où  $n$  est le nombre d'éléments,  $h$  la taille du support de spline et  $\hat{\sigma}^2$  l'erreur quadratique empirique de l'interpolation.

- Pour  $p = 2$ , c'est-à-dire lors d'une interpolation par splines linéaires, on définit le test statistique :

$$T_{2,j} = \frac{|\hat{m}_2(j+1) - \hat{m}_2(j-1) - 2 * \hat{m}_2(j)|}{2\sigma_{2,j}}$$

$$\sigma_{2,j} = \frac{3\hat{\sigma}^2}{8nh} \zeta^\top S_j \zeta$$

avec  $n$  le nombre d'éléments,  $h$  la taille du support de spline,  $\hat{\sigma}^2$  l'erreur

par moindre carré de l'interpolation.  $\zeta = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$  et  $S = V^{-1}$  avec  $V$

la matrice du produit scalaire de la base des splines :

$$V = \begin{pmatrix} 1 & \sqrt{2}/4 & & & & 0 \\ \sqrt{2}/4 & 1 & 1/4 & & & \\ & 1/4 & 1 & \ddots & & \\ & & \ddots & \ddots & 1/4 & \\ & & & 1/4 & 1 & \sqrt{2}/4 \\ 0 & & & & \sqrt{2}/4 & 1 \end{pmatrix}_{(N+2) \times (N+2)}$$

et

$$S_j = \begin{pmatrix} S_{(j-2),(j-2)} & S_{(j-2),(j-1)} & S_{(j-2),(j)} \\ S_{(j-1),(j-2)} & S_{(j-1),(j-1)} & S_{(j-1),(j)} \\ S_{(j),(j-2)} & S_{(j),(j-1)} & S_{(j),(j)} \end{pmatrix}$$

Nous calculons un seuil sur les valeurs des  $T_{p,j}$  en appliquant le test statistique au risque  $\alpha$  sous hypothèse  $H_0$ , en appliquant la formule du théorème 2.1 de l'article de [Ma and Yang \(2011\)](#) :

$$\lim_{n \rightarrow \infty} P \left[ T_{p,j} > \sqrt{2 \log(N - 2p + 2)} d_{N-2p+2}(\alpha) \right] = \alpha, p = 1, 2.$$

Avec :

$$d_N(\alpha) = 1 - \frac{\log\left(-\frac{1}{2} \log(1 - \alpha)\right) + \frac{1}{2} \{\log \log(N) + \log(4\pi)\}}{2 \log(N)}.$$

Ainsi nous avons les positions des nœuds où se situent les éventuels sauts.

Dans l'algorithme, la difficulté est d'obtenir automatiquement la taille de fenêtre pour l'interpolation, ce qui revient à calculer le nombre de nœuds. Le nombre de nœuds, équirépartis sur le support des observations, est calculé comme celui qui minimisera le critère d'information bayésien défini par  $BIC = -2 \ln(\hat{\sigma}) + \frac{N}{n} \cdot \ln(n)$  avec  $\hat{\sigma}$  l'erreur quadratique empirique,  $n$  le nombre d'observations et  $N$  le nombre de nœuds.

La technique retourne les positions de nœuds où se situent les sauts. Chaque nœud représente une portion de courbe. Afin de calculer les positions précises, nous devons étudier plus précisément la portion de courbe. Pour cela, nous calculons le gradient local, qui sera proche d'une dérivée de fonction linéaire et nous supprimons les points aberrants via le seuillage du gradient local en valeur absolue.

**Lissage des coefficients** Dans un premier temps, on supprime les sauts d'asservissement afin de pouvoir lisser les coefficients, pour ensuite remettre le saut uniquement pour les ordonnées à l'origine afin de pouvoir corriger correctement le saut.

La Figure [1.33](#) résume les différentes étapes du lissage des coefficients, qui sont :

1. Suppression des sauts d'asservissement [1.33b](#) : afin de détecter les sauts, nous appliquons la méthode décrite précédemment. Au niveau

de ce saut, on corrige en enlevant la valeur du gradient à partir de la/les lignes en questions. On supprime les positions des sauts sur les coefficients directeurs et les ordonnées à l'origine. En effet, si le saut est en cours de ligne alors le coefficient ne sera pas bon et le lissage affectera son voisinage, ce que nous ne souhaitons pas.

2. Lissage des coefficients **1.33c** : Pour lisser les coefficients, plusieurs type de filtrages peuvent-être appliqués, le but principal étant de prendre en compte le voisinage. Nous pouvons donc faire des interpolations, des filtrages par filtre médian, moyenneur ou gaussien (voir le livre de [Lingrand \(2008\)](#)). Dans notre cas, nous avons choisi d'appliquer un filtre moyenneur avec un masque de taille 5.
3. Réajustement des sauts d'asservissement **1.33d** : Afin que le redressement fonctionne bien, il faut être en mesure de réinsérer les sauts sur la courbe des ordonnées à l'origine. En effet, le redressement s'applique en soustrayant pour chaque ligne la droite affine ayant les coefficients lissés, si les sauts ne sont pas réinsérés, ils ne seront pas corrigés sur l'image.

Les lignes contenant les sauts sont supprimées et remplacées par une interpolation des lignes voisines. En effet, le saut peut arriver au milieu d'une ligne et donc le signal est corrompu pour cette ligne. Si nous avons fait le choix de supprimer la ligne, nous aurions une ligne manquante dans l'image cela aurait complexifié l'analyse qui suivrait le redressement. Nous avons alors choisi l'interpolation car le signal est localement similaire et l'interpolation permet d'estimer le signal non corrompu. Au final, l'estimation locale ne sera pas de moins bonne qualité que la ligne corrompue.

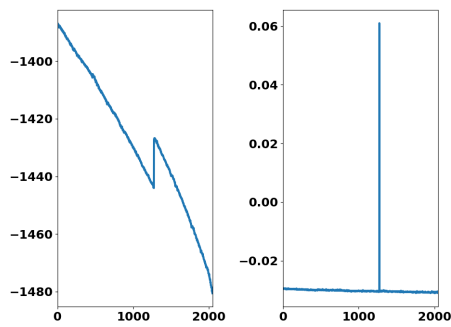
### Extraction automatique du paramètre d'asymétrie

Dans le cadre d'un redressement linéaire ligne à ligne, nous appliquons des M-estimateurs asymétriques. L'asymétrie nous impose de savoir où sont placés les objets par rapport à la surface, afin de connaître la nature de l'asymétrie à appliquer, et ainsi forcer les objets à être « considérés » comme des outliers.

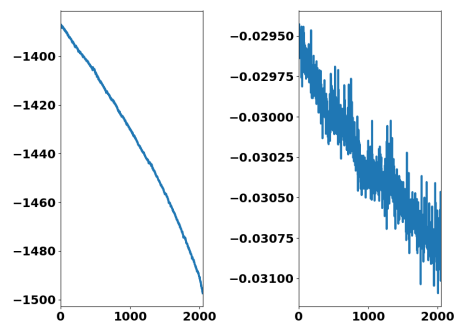
Pour cela, nous effectuons deux redressements par M-estimateurs asymétriques :

- Une interpolation en supposant que les objets sont dessus, c'est-à-dire en appliquant une asymétrie qui donne essentiellement de l'information à la partie basse.
- Une interpolation en supposant que les objets sont dessous, c'est-à-dire en appliquant une asymétrie qui donne essentiellement de l'informa-

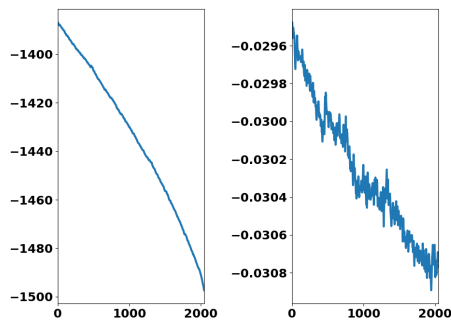




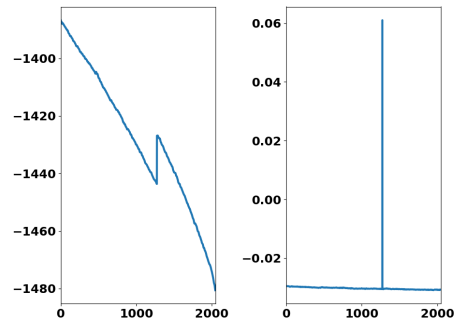
(a) Coefficients bruts extraits par M-estimateurs asymétriques



(b) Coefficients 1.33a après correction du saut d'asservissement



(c) Coefficients 1.33b après lissage



(d) Coefficients 1.33c après réinsertion du saut d'asservissement

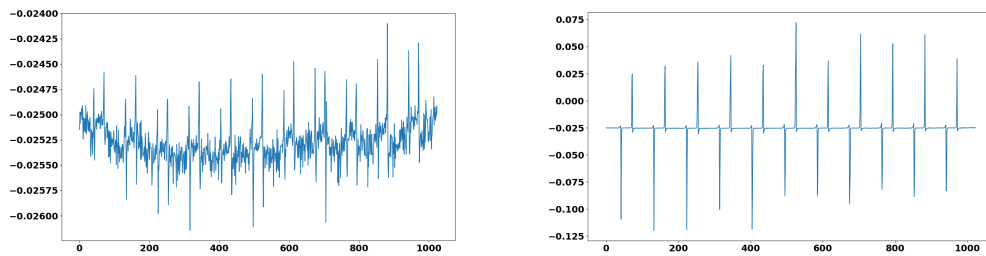
FIGURE 1.33 – Exemple des étapes de lissage des coefficients polynomiaux extraits par M-estimateur asymétriques en fonction des lignes pour l'image 1.32a.

tion à la partie haute.

Nous comparons les coefficients directeurs de chacun des résultats afin de déterminer quel modèle d'asymétrie correspond à la réalité de l'image. Pour cela, nous étudions la variation. La variance empirique la moins forte est celle qui correspond à la position réelle des objets.

La Figure 1.34, représente les coefficients directeurs obtenus suite à l'interpolation par M-estimateurs asymétriques. Le graphique 1.34a contient un tracé des coefficients en ayant donné de l'importance à la partie haute (c'est-à-dire en supposant que les objets sont dessous). Le graphique 1.34b contient un tracé des coefficients en donnant de l'importance à la partie basse des données (c'est-à-dire en supposant que les objets sont dessus).

On observe que les coefficients varient beaucoup plus dans le cas où nous avons supposé que les objets étaient au dessus de la surface. On observe très clairement que la courbe de gauche oscille beaucoup moins, la plage des valeurs étant 5 fois plus petite. Nous pouvons donc en conclure que les objets sont en dessous de la surface.



(a) Objet au dessous, on donne de l'importance au dessus

(b) Objet au dessus, on donne de l'importance au dessous

FIGURE 1.34 – Exemple de courbe de coefficients directeurs, extraits par M-estimateurs asymétriques, en fonction du numéro de la ligne pour l'image 1.27c

En pratique, nous nous concentrons sur la valeur de l'écart-type empirique calculé sur la population des coefficients directeur extraits par ligne. La population donnant le plus petit écart-type sera celle qui correspondra le mieux à la réalité. Nous aurions également pu comparer les redressements résultants des interpolations, comme par exemple le gradient de la différence. Cependant, nous avons mentionné précédemment que chaque ligne n'était pas indépendante des lignes voisines. C'est pourquoi la variation des coefficients est un bon indicateur sur la qualité du redressement et donc sur la position des objets.

## 1.4 Comparaison des méthodes

*Dans cette section, nous allons effectuer une comparaison des résultats de redressement. Pour cela, nous évaluerons les critères présentés dans la section 1.2 sur différents résultats provenant de différentes techniques et logiciels.*

*Nous détaillerons ensuite les différences entre les techniques internes et les techniques commerciales.*

### 1.4.1 Liste des logiciels et techniques employés

- **Gwyddion** : logiciel libre
- **Mountains Map** : logiciel commercial
- **SPIP** : logiciel commercial
- **WsXM** : logiciel gratuit
- **Nanoscope** : logiciel commercial de Bruker, fabricant de microscopes AFM. Le redressement est fait par un plan polynomial d'ordre 3.
- **Global** : redressement en interpolant toute la surface par un polynôme d'ordre 3 par moindres carrés.
- **Global par M-estimateurs symétriques** : redressement en interpolant toute la surface par un polynôme d'ordre 2 par M-estimateurs utilisant une  $\psi$  fonction biweight de Tukey symétrique. Nous avons choisi un degré 2 plutôt qu'un degré 3 car cette interpolation semble produire de meilleurs résultats sur les images testées.
- **Global par M-estimateurs asymétriques** : redressement en interpolant toute la surface par un polynôme d'ordre 2 par M-estimateurs asymétriques utilisant une  $\psi$  fonction biweight de Tukey asymétrique.
- **Ligne à ligne avec exclusion des objets** : premier redressement en interpolant la surface par un polynôme d'ordre 3 par moindres carrés puis exclusion des objets via détection par MSER<sup>3</sup> puis redressement linéaire ligne à ligne par moindres carrés.
- **Ligne à ligne par M-estimateurs symétriques** : redressement linéaire ligne à ligne par M-estimateurs utilisant une  $\psi$  fonction biweight de Tukey symétrique.
- **Ligne à ligne par M-estimateurs asymétriques** : redressement linéaire ligne à ligne par M-estimateurs asymétriques utilisant une  $\psi$  fonction biweight de Tukey asymétrique.
- **Ligne à ligne par M-estimateurs asymétriques avec lissage des coefficients** : redressement décrit précédemment avec un lissage des coefficients ainsi que détection des sauts d'asservissement. La ligne

---

3. Maximally Stable Extremal Regions, voir la publication de [Nistér and Stewénius \(2008\)](#)

contenant le saut est remplacée par une interpolation des deux lignes qui l'entourent.

### 1.4.2 Liste des critères et explications

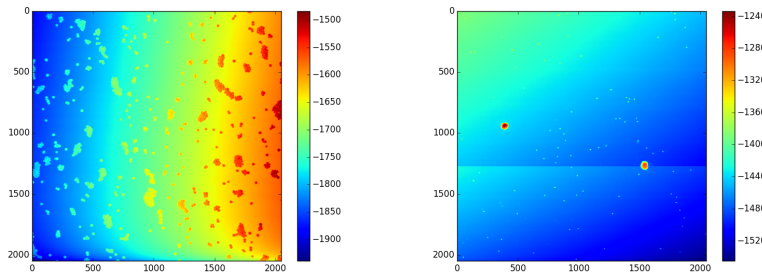
Les critères sont ceux décrits dans la section 1.2 et résumés dans la sous-section 1.2.2. Nous les rappelons ici avec leur interprétation.

- Variance du fond de l'image  $\sigma_F^2$  : la valeur est la variance brute du fond de l'image. On souhaite que cette valeur soit la plus faible possible car elle est sensée représenter le carré de la rugosité. Le fond de l'image est déterminé par segmentation en utilisant les MSER. Cette valeur permet de quantifier la planéité de la surface dans le sens où elle doit être la plus proche du carré de la rugosité, qui est la borne inférieure pour cette valeur.
- La variance de ligne extrême  $\sigma_{L_{max}}^2$  : représente la distance maximale entre la variance de population du fond de ligne et la moyenne de variance de population du fond de ligne.
- Le rapport de variances :

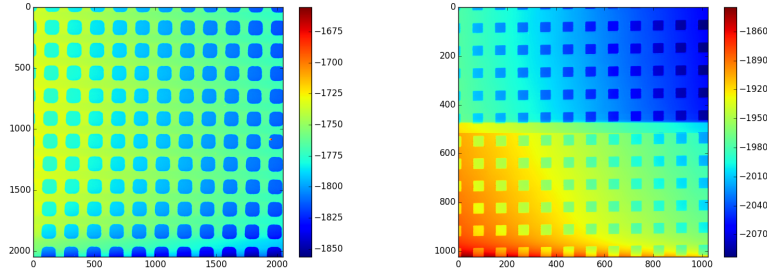
$$\rho = \frac{\sigma_{GB}^2}{\sigma_G^2}$$

Avec  $\sigma_G^2$  la variance totale et  $\sigma_{GB}^2$  variance du fond du gradient de  $D$ , où  $D$  est l'image qui a été retirée à l'image brute pour le redressement, calculée comme la différence entre l'image brute et l'image redressée. La population du fond est extraite par le même masque que pour les autres critères.

### 1.4.3 Comparaison des méthodes sur des images types



(a) Image AFM de nanoparticule sans saut d'asservissement      (b) Image AFM de nanoparticule avec saut d'asservissement



(c) Image AFM de réseau de plots sans saut d'asservissement (d) Image AFM de réseau de plots avec sauts d'asservissements

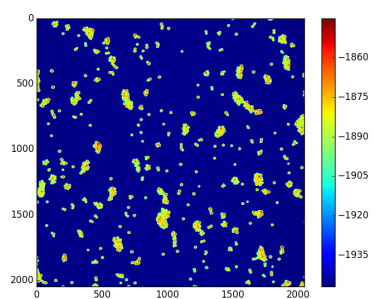
FIGURE 1.35 – Images AFM brutes qui seront la base de la comparaison des différentes techniques et logiciels.

Les logiciels/méthodologies et les critères rappelés dans la section précédente seront évalués sur les 4 images suivantes qui nous semblent représenter une certaine exhaustivité :

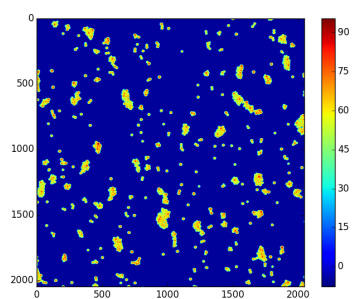
1. Image de nanoparticules sans saut d'asservissement : dans la sous-section 1.4.3, nous analyserons les résultats des critères sur les différents redressements sur l'image AFM 1.35a
2. Image de nanoparticules avec saut d'asservissement : dans la sous-section 1.4.3, nous analyserons les résultats des critères sur les différents redressements sur l'image AFM 1.35b
3. Image de réseau de plots sans saut d'asservissement : dans la sous-section 1.4.3, nous analyserons les résultats des critères sur les différents redressements sur l'image AFM 1.35c
4. Image de réseau de plots avec saut d'asservissement : dans la sous-section 1.4.3, nous analyserons les résultats des critères sur les différents redressements sur l'image AFM 1.35d

### Image de nanoparticules sans saut d'asservissement

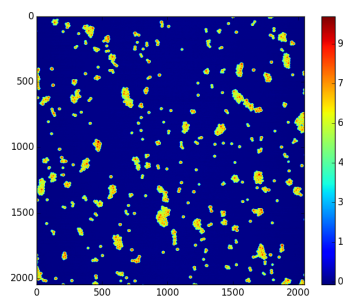
1. Les techniques Nanoscope et les méthodes de redressement global n'ont pas bien « aplani » le bas de l'image (comme on peut l'observer sur les Figures 1.37 et 1.36b), ce qui se traduit par une valeur de  $\sigma_F^2$  important en comparaison aux autres méthodes.
2. Les valeurs de  $\sigma_F^2$  et de  $\rho$  des techniques commerciales sont faibles en comparaison aux valeurs des autres méthodes. Ces deux critères



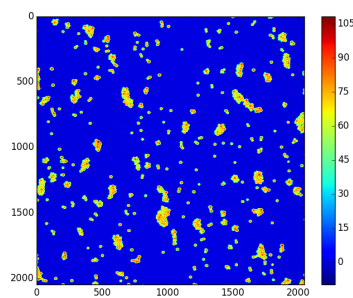
(a) Gwyddion



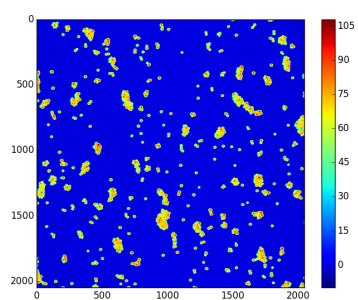
(b) MountainsMap



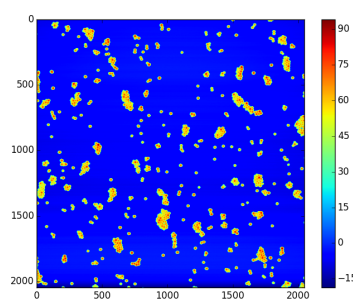
(c) SPIP



(d) WsXM avec path 1



(e) WsXM avec path 2

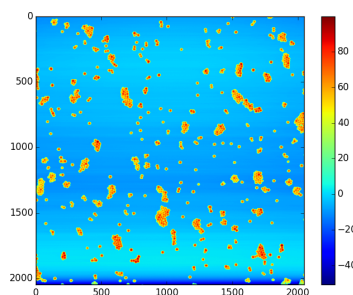


(f) Nanoscope

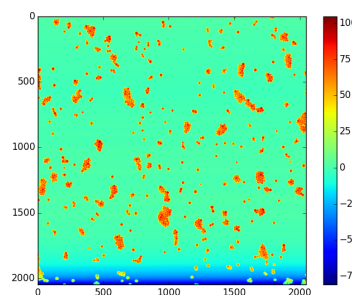
FIGURE 1.36 – Résultats de redressement par logiciels commerciaux de l'image AFM 1.35a.

#### 1.4. COMPARAISON DES MÉTHODES

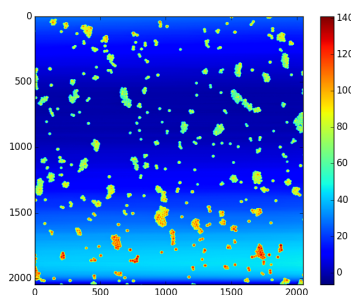
---



(a) Global

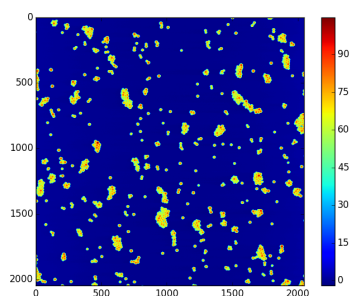


(b) Global par M-estimateurs symétriques

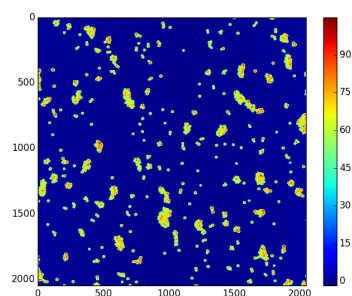


(c) Global par M-estimateurs asymétriques

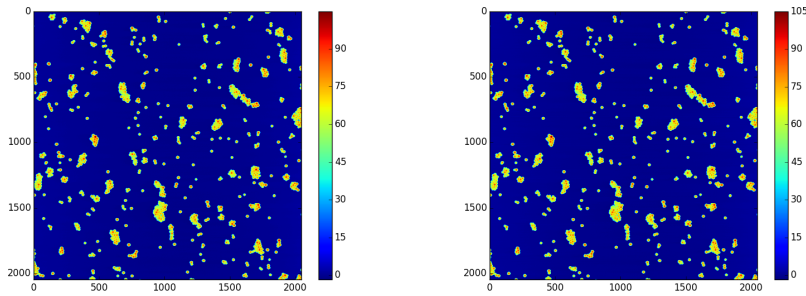
FIGURE 1.37 – Résultats de redressements par techniques de redressement global de l'image AFM 1.35a.



(a) Ligne à ligne par M-estimateurs



(b) Ligne à ligne avec exclusion des objets



(c) Ligne à ligne par M-estimateurs asymétriques (d) Ligne à ligne par M-estimateurs asymétriques avec lissage des coefficients

FIGURE 1.38 – Résultats de redressements par techniques de redressement ligne à ligne de l'image AFM 1.35a.

Méthode	Image		Gradient
	$\sigma_F^2$	$\sigma_{L_{max}}^2$	$\rho$
Gwyddion	0.070	24.207	0.527
Mountains Map	0.028	24.430	0.536
SPIP	0.071	24.208	0.527
WsXM_1	0.137	27.962	0.608
WsXM_2	0.126	27.990	0.627
Nanoscope	3.638	23.779	0.447
Global	21.301	27.038	0.656
Global par M-estimateurs symétriques	87.814	28.825	0.575
Global par M-estimateurs asymétriques	211.705	25.318	0.731
Ligne à ligne avec exclusion objets	0.249	25.379	0.853
Ligne à ligne par M-estimateurs symétriques	0.257	25.379	0.859
Ligne à ligne par M-estimateurs asymétriques	0.250	25.375	0.858
Ligne à ligne par M-estimateurs asymétriques avec lissage coefficients	0.255	25.371	0.866

TABLE 1.1 – Évaluation des valeurs des critères de qualité pour les résultats de redressement de l'image de la Figure 1.35b.

Les valeurs des critères représentant la qualité du redressement présents sur la Table 1.1, de comparaison des méthodes de redressement permettent de conclure :



#### 1.4. COMPARAISON DES MÉTHODES

---

laissent supposer que les images ont subi un filtrage. Le filtrage va entraîner une variance plus faible sur le fond de l'image ainsi qu'une modification sur le contour des objets, ce qui implique le  $\rho$  plus faible.

D'après la table 1.1, on observe que les méthodes globales donnent de mauvais résultats. En effet, la première colonne contient principalement des petites valeurs, donc on a des images plutôt planes sauf pour les méthodes globales. Nous pouvons donc conclure que ces méthodes sont mal adaptées pour cette image.

Les valeurs de  $\rho$  montrent que les objets sont modifiés dans les techniques commerciales. On peut en conclure que les deux meilleures méthodes sont « ligne à ligne avec exclusion des objets » et « ligne à ligne par M-estimateurs asymétriques avec lissage des coefficients ».

#### Image de nanoparticules avec saut d'asservissement

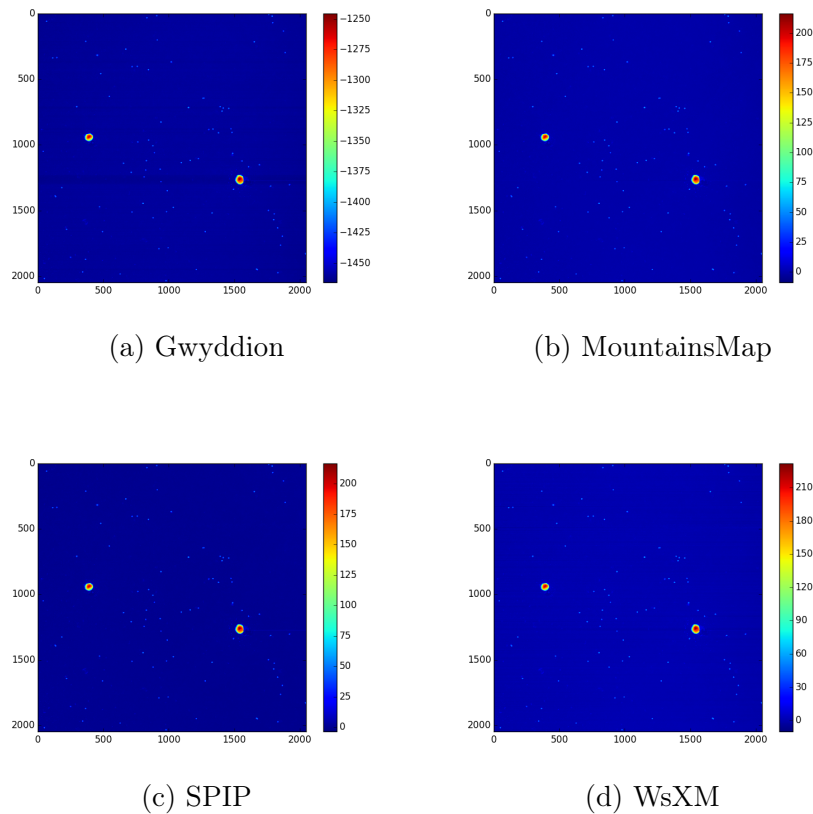


FIGURE 1.39 – Résultats de redressement par logiciels commerciaux de l'image AFM 1.35b.

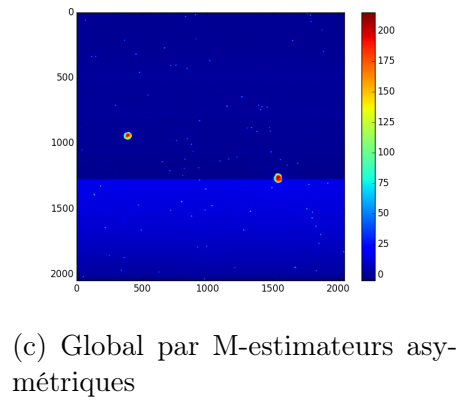
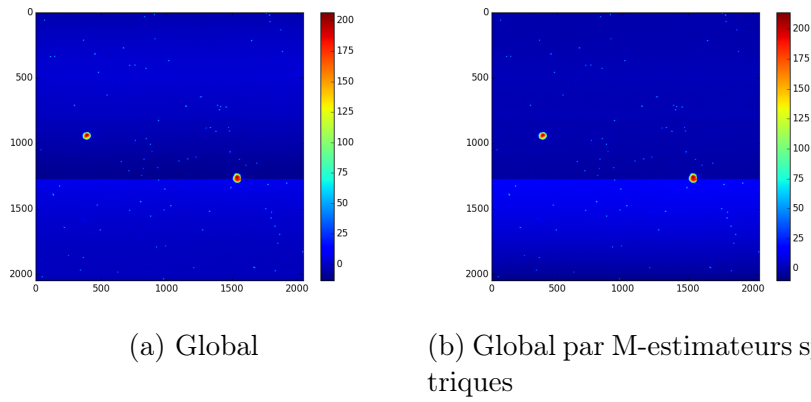
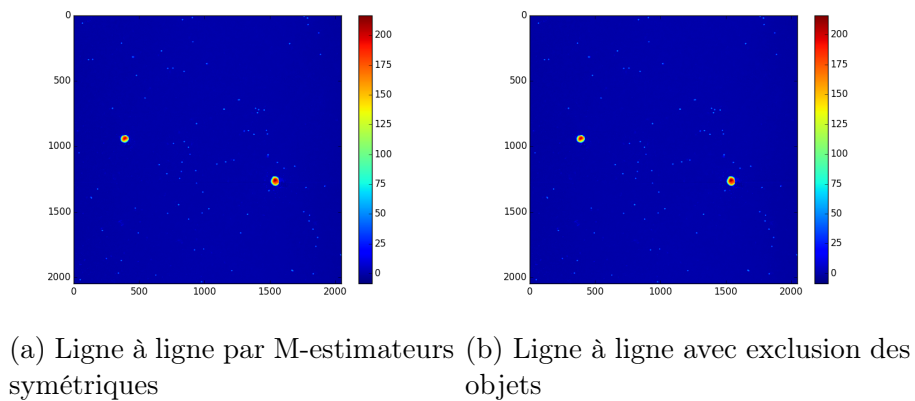


FIGURE 1.40 – Résultats de redressements par techniques de redressement global de l'image AFM 1.35b.



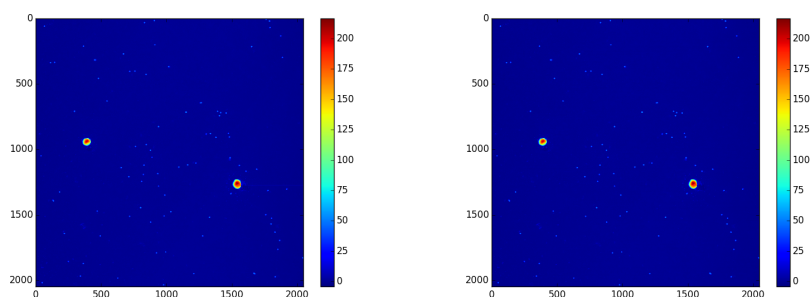
#### 1.4. COMPARAISON DES MÉTHODES

---

Méthode	Image		Gradient
	$\sigma_F^2$	$\sigma_{L_{max}}^2$	$\rho$
Gwyddion	0.856	46.446	0.982
Mountains Map	0.669	18.100	0.981
SPIP	0.531	50.235	0.980
WsXM	0.930	20.934	0.939
Global	14.071	51.144	1.000
Global par M-estimateurs symétriques	24.700	51.244	1.000
Global par M-estimateurs asymétriques	31.322	51.179	1.000
Ligne à ligne avec exclusion objets	0.664	18.015	0.980
Ligne à ligne par M-estimateurs symétriques	0.666	18.012	0.980
Ligne à ligne par M-estimateurs asymétriques	0.683	42.595	0.980
Ligne à ligne par M-estimateurs asymétriques avec lissage coefficients	0.668	12.555	0.956

TABLE 1.2 – Évaluation des valeurs des critères de qualité pour les résultats de redressement de l’image 1.35b.

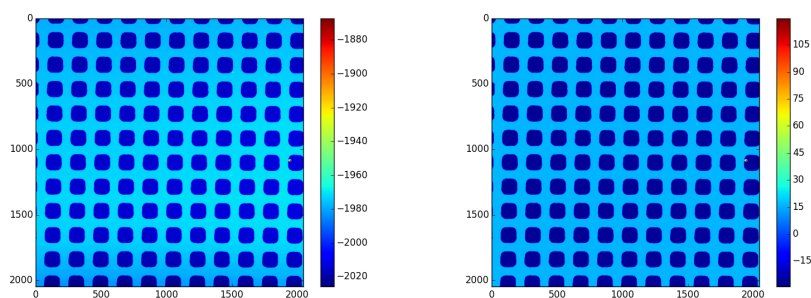
Au vu des deux premières colonnes de la table 1.2, on observe que les techniques de redressement global, ne donnent pas de bons résultats tandis que les autres techniques sont comparables. La variance de ligne montre que la technique de redressement par M-estimateurs asymétriques avec lissage des coefficients vient corriger le problème de saut, ou plus exactement, la ligne du saut. Les valeurs de  $\rho$  sont sensiblement les mêmes pour les techniques qui redressent correctement l’image. Nous pouvons donc conclure que la technique qui corrige le mieux l’image est la technique « ligne à ligne par M-estimateurs asymétriques avec lissage coefficients » .



(c) Ligne à ligne par M-estimateurs asymétriques (d) Ligne à ligne par M-estimateurs asymétriques avec lissage des coefficients

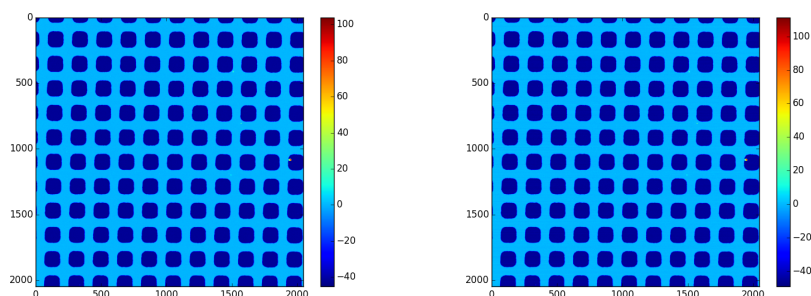
FIGURE 1.41 – Résultats de redressements par techniques de redressement ligne à ligne de l'image AFM 1.35b.

### Image de réseau de plots sans saut d'asservissement



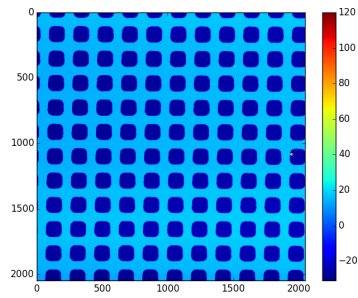
(a) Gwyddion

(b) Mountains Map



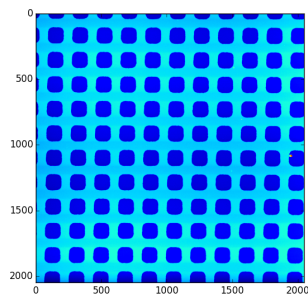
(c) SPIP

(d) WsXM

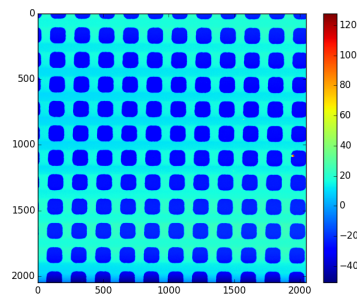


(e) Nanoscope

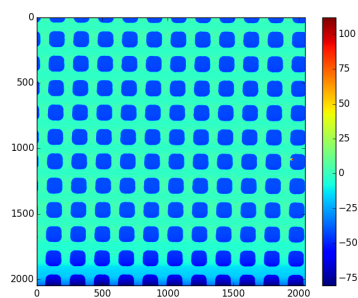
FIGURE 1.42 – Résultats de redressements par logiciels commerciaux de l'image AFM 1.35c.



(a) Global

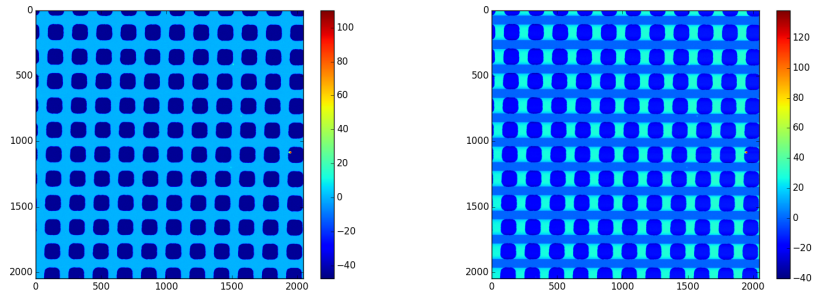


(b) Global par M-estimateurs symétriques

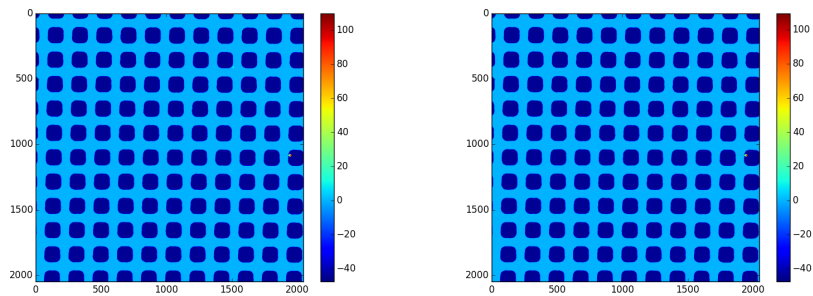


(c) Global par M-estimateurs asymétriques

FIGURE 1.43 – Résultats de redressements par techniques de redressement global de l'image AFM 1.35c.



(a) Ligne à ligne avec exclusion des objets (b) Ligne à ligne par M-estimateurs symétriques



(c) Ligne à ligne par M-estimateurs asymétriques (d) Ligne à ligne par M-estimateurs asymétriques avec lissage des coefficients

FIGURE 1.44 – Résultats de redressements par techniques de redressement ligne à ligne de l'image AFM 1.35c.

#### 1.4. COMPARAISON DES MÉTHODES

---

Méthode	Image		Gradient
	$\sigma_F^2$	$\sigma_{L_{max}}^2$	$\rho$
Gwyddion	6.195	14.337	0.414
Mountains Map	0.351	13.491	0.492
SPIP	0.351	13.476	0.434
WsXM	0.691	15.576	0.538
Nanoscope	1.912	14.613	0.419
Global	8.676	20.240	0.976
Global par M-estimateurs symétriques	13.450	17.009	0.997
Global par M-estimateurs asymétriques	27.614	15.804	0.999
Ligne à ligne avec exclusion objets	0.402	15.620	0.804
Ligne à ligne par M-estimateurs symétriques	160.237	32.654	0.896
Ligne à ligne par M-estimateurs asymétriques	0.404	15.627	0.799
Ligne à ligne par M-estimateurs asymétriques avec lissage coefficients	0.404	15.617	0.800

TABLE 1.3 – Évaluation des valeurs des critères de qualité pour les résultats de redressement de l'image 1.35c.

D'après la table 1.3, on peut faire une analyse similaire à la table 1.1 on observe que la technique qui correspond le mieux est la méthode « ligne à ligne par M-estimateurs asymétriques avec lissage coefficients ».

Image de réseau de plots avec saut d'asservissement

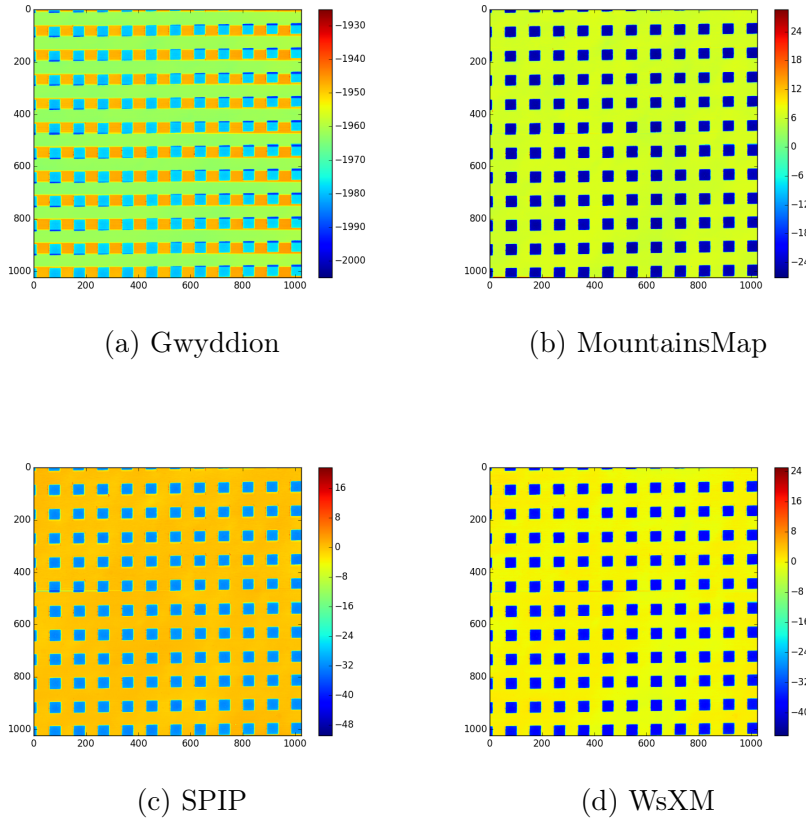
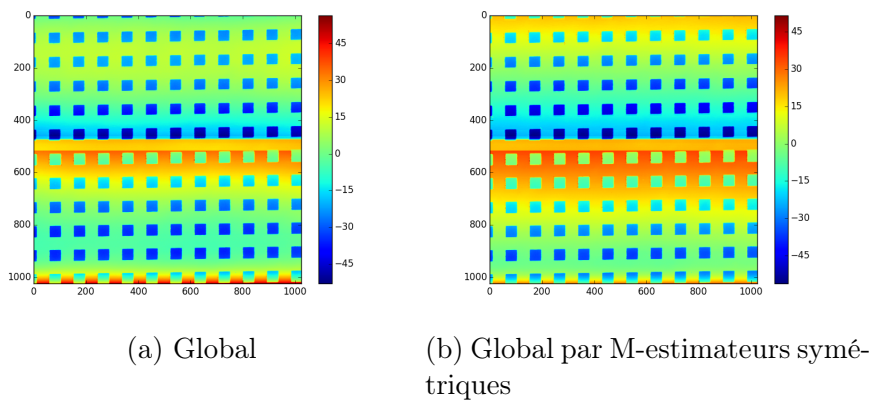
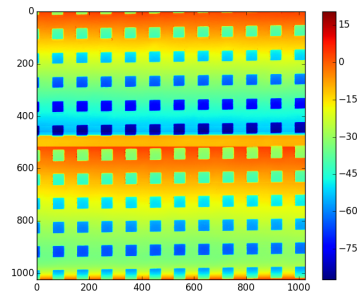


FIGURE 1.45 – Résultats de redressement par logiciels commerciaux de l'image AFM 1.35d.

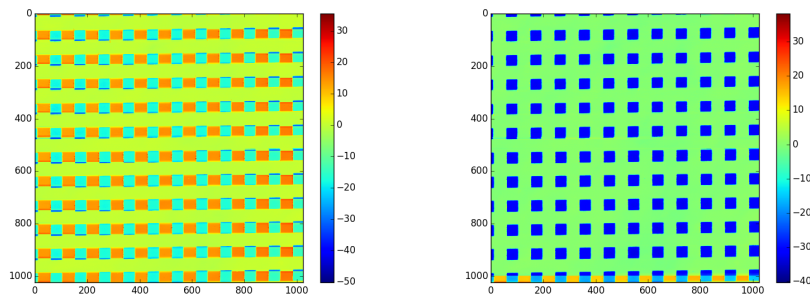




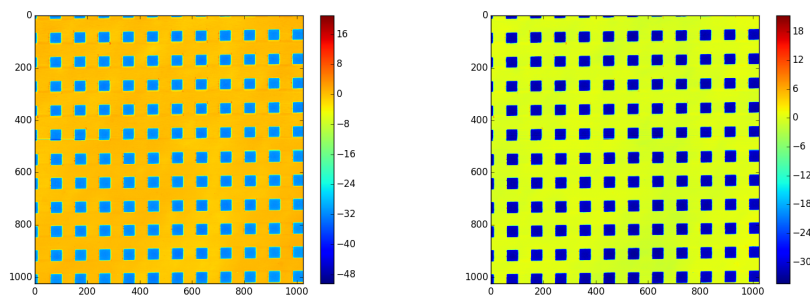


(c) Global par M-estimateurs asymétriques

FIGURE 1.46 – Résultats de redressements par techniques de redressement global de l'image AFM 1.35d.



(a) Ligne à ligne par M-estimateurs symétriques (b) Ligne à ligne avec exclusion des objets



(c) Ligne à ligne par M-estimateurs asymétriques (d) Ligne à ligne par M-estimateurs asymétriques avec lissage des coefficients

FIGURE 1.47 – Résultats de redressements par techniques de redressement ligne à ligne de l'image AFM 1.35d.

Méthode	Image		Gradient
	$\sigma_F^2$	$\sigma_{L_{max}}^2$	$\rho$
Gwyddion	40.807	62.488	0.855
Mountains Map	0.326	62.457	0.966
SPIP	0.319	69.050	0.983
WsXM	0.633	81.334	0.687
Global	124.957	61.729	0.934
Global par M-estimateurs symétriques	194.597	64.616	0.931
Global par M-estimateurs asymétriques	209.200	66.987	0.942
Ligne à ligne avec exclusion objets	2.659	63.127	0.531
Ligne à ligne par M-estimateurs symétriques	38.675	55.534	0.745
Ligne à ligne par M-estimateurs asymétriques	0.342	63.264	0.973
Ligne à ligne par M-estimateurs asymétriques avec lissage coefficients	0.323	63.434	0.875

TABLE 1.4 – Évaluation des valeurs des critères de qualité pour les résultats de redressement de l'image 1.35d.

D'après la table 1.4, les techniques globales ne sont pas bonnes et il faut nous concentrer uniquement sur les techniques ligne à ligne. Quatre techniques semblent comparables : Mountains Map, SPIP, M-estimateurs asymétriques et M-estimateurs asymétriques avec lissage des coefficients. Les différences sont dues à la gestion des sauts d'asservissement qui sont présents dans l'image. Au vu des résultats empiriques, la meilleure technique est celle de Mountains Map suivie de près par les techniques de ligne à ligne par M-estimateurs asymétriques avec ou sans lissage des coefficients.

### Synthèse des résultats empiriques

À la vue des tables présentées précédemment, nous constatons que la technique de redressement de ligne à ligne par M-estimateurs asymétriques avec lissage des coefficients est la technique qui offre le meilleur compromis entre utilisabilité et performance. En effet dans la plupart des cas, la technique est la meilleure ou bien comparable aux meilleures. La technique de référence dans le domaine est Mountains Map, qui fournit de bons résultats dans tous les exemples présentés précédemment. La technique que nous avons développée possède des performances comparables à Mountains Map d'après les critères. D'autre part, la technique que nous avons développée est une méthode dirigée par les données, elle prend uniquement l'image AFM en entrée et redresse l'image automatiquement sans intervention humaine.

## 1.5 Analyse d'incertitude de la méthode

Dans la section précédente, nous avons comparé nos méthodes avec les méthodes couramment utilisées pour le redressement AFM. Il en est ressorti que la méthode que nous devons appliquer est la méthode de redressement ligne à ligne avec estimation des coefficients par M-estimateurs asymétriques, avec lissage des coefficients. Nous allons maintenant analyser la sensibilité de cette méthode. Dans un premier temps nous présenterons le protocole de test que nous avons mis en place puis dans un second temps nous allons faire l'analyse des résultats pour tirer des conclusions sur la robustesse de notre méthode.

### 1.5.1 Protocole de test

Afin d'analyser la robustesse au bruit de notre algorithme, nous avons choisi de faire une étude Monte Carlo. Pour cela, nous avons sélectionné l'image 1.35b, qui contient des particules déposées de manière aléatoire et un saut d'asservissement. Cette image est donc une image représentative et nous pouvons nous en servir comme base pour notre analyse de robustesse au bruit.

Le protocole de l'étude Monte Carlo est le suivant :

1. Ajouter un bruit blanc gaussien à l'image originale. Nous avons choisi de prendre  $\sigma = 3$  car la rugosité du substrat est proche de 1. Cette rugosité est estimée de la même manière que  $\sigma_F$  dans la section précédente.

2. Utilisation de l'algorithme de redressement sur l'image bruitée. Puis sauvegarde des coefficients extraits ainsi que de la position du saut.
3. Répéter 500 fois les étapes 1 et 2

Une fois les 500 jeux de coefficients obtenus, nous pouvons les étudier pour caractériser la robustesse de notre algorithme au bruit. La variance des coefficients ainsi que celle de la position du saut nous indiquera la robustesse de l'algorithme.

### 1.5.2 Résultats

Nous avons dans un premier temps calculé le coefficient de variation des différents coefficients extraits. Rappelons que nous avons 500 fois deux vecteurs de coefficients de taille 2048, soit le nombre de ligne de l'image. Un vecteur pour les coefficients directeurs et un vecteur pour les ordonnées à l'origine. Le coefficient de variation est calculé comme l'écart-type, normalisé par la moyenne. Nous avons alors un coefficient de variation de valeur moyenne sur les 2048 lignes de l'image de 0,22% pour les coefficients directeurs et de 0,0057% pour les ordonnées à l'origine.

Pour chaque image simulée, l'algorithme retourne toujours un seul et unique saut toujours positionné au même endroit.

La Figure 1.48 présente la matrice de l'écart-type de chaque pixel pour les 500 images redressées lors de l'étude Monte Carlo. On observera sur l'image de la Figure 1.48 que la moyenne de cette image est égale à  $\frac{\sqrt{3}}{500}$  ce qui revient à l'erreur théorique attendue. D'autre part, nous noterons que l'écart-type de cette matrice est 1,45 fois plus grande que la moyenne, ce qui reste très faible.

On peut conclure de cette étude que notre algorithme est robuste à du bruit supplémentaire.

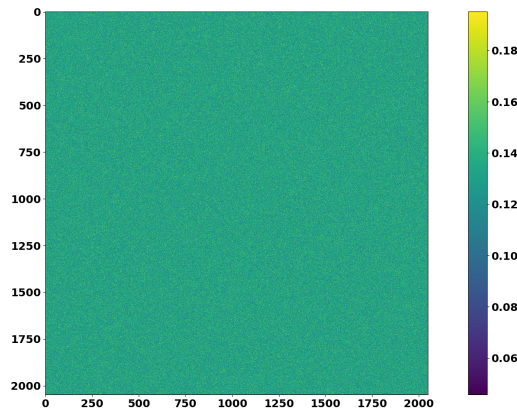


FIGURE 1.48 – Image de l'écart-type de chaque pixel des 500 images redressées lors de l'étude Monte Carlo.

## 1.6 Conclusion

Dans ce chapitre, nous avons dans un premier temps analysé la problématique à laquelle nous devons répondre, à savoir être en mesure de redresser des images AFM. Pour cela, la première partie fut tournée vers l'étude de l'origine des défauts et de leurs caractéristiques ainsi que l'analyse des techniques existantes pour savoir en quoi nous pouvions contribuer à l'existant.

Afin de pouvoir comparer les méthodes, nous avons étudié des façons de comparer les méthodes, via des critères qui mettent en avant les différents objectifs d'un redressement d'image AFM réussi.

Nous avons alors réalisé différents algorithmes de redressement, toutes dirigés par les données, en utilisant différentes méthodes pour l'estimation de la dérive via des méthodes classiques ou robustes. Puis comparé les méthodes par rapport aux méthodes existantes. Nous avons sélectionné la méthode la plus efficace au vue des chiffres puis avons étudié la robustesse de cette méthode face à du bruit additionnel.

En conclusion, nous avons développé une méthode dirigée par les données, ne nécessitant de par le fait aucune intervention humaine, qui permet de redresser une images AFM dans un temps d'exécution respectable et qui de plus est robuste au bruit. Ce travail est en cours de publication. Cette contribution est également intégrée dans les produits de Pollen Metrology.

# Annexes

## 1.A Description du procédé de mesure par AFM

Un microscope à force atomique (ou AFM pour Atomic Force Microscope) est un microscope qui fait parti de la famille des microscopes à champ proche. Ce type d'appareil a été introduit en 1985 par G.Bining, C.F. Quate et C.Gerber. Son fonctionnement est basé sur le concept de microscopie à effet tunnel, permettant l'étude de surface de matériaux isolants à l'échelle atomique, décrit dans le livre de Landis (2010).

### 1.A.1 Fonctionnement d'un AFM

Pour la description du fonctionnement d'un AFM, nous nous sommes inspirés du livre de Yablon (2013), que l'on pourra consulter pour plus de détails sur le sujet. La Figure 1.A.1 montre le principe de fonctionnement d'un AFM.

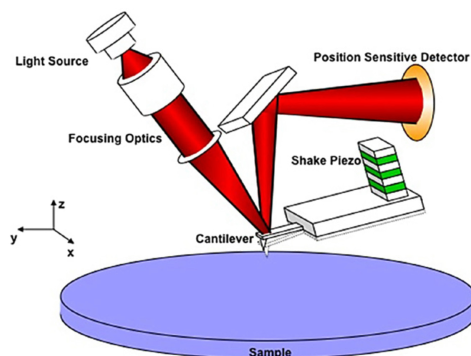


FIGURE 1.A.1 – Schéma du fonctionnement d'un AFM

L'AFM fonctionne sur le principe de l'interaction avec la matière. On vient mesurer l'interaction d'une pointe AFM, voir Figure 1.A.1, avec la

## 1.A. DESCRIPTION DU PROCÉDÉ DE MESURE PAR AFM

---

surface à scanner. Il existe différentes géométries de pointes AFM possibles, un exemple est présenté sur la Figure 1.A.2. Le type de pointe utilisée dépend des modes de mesure (ceux-ci sont présentés dans la section 1.A.2).

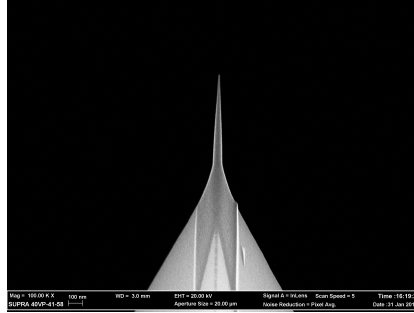


FIGURE 1.A.2 – Image d’une pointe AFM conique, obtenue par microscopie électronique à balayage

La pointe AFM est une pièce, généralement en carbone, qui permet de se rapprocher de la surface, suffisamment pour interagir avec la matière.

Cette pointe est placée au bout d’un cantilever, voir Figure 1.A.3. Le cantilever est la pièce maîtresse d’un AFM. Ce sont les mouvements de ce cantilever que l’on observe afin d’analyser l’interaction de la pointe avec la matière.

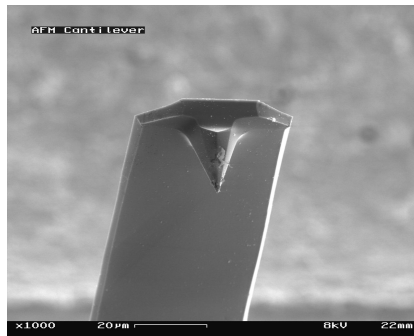


FIGURE 1.A.3 – Image obtenue par microscopie électronique à balayage d’un cantilever AFM (grossissement  $\times 1000$ )

Le cantilever est placé sur un tube piézoélectrique qui permet de le déplacer. Un tube piézoélectrique est un tube qui se déforme en fonction de l’application d’une différence de potentiel. Le tube peut être déformé selon

les trois dimensions horizontale, longitudinale et verticale que l'on notera  $X$ ,  $Y$  et  $Z$ . Il permet de faire vibrer le cantilever et de le déplacer en  $X$ ,  $Y$  et  $Z$ . C'est lui qui rapproche la pointe de la matière.

Afin de mesurer les variations de déplacement du cantilever, on envoie un faisceau laser sur le bout du cantilever. Ce faisceau est réfléchi sur une matrice de photodiodes. À partir de ce signal et connaissant la géométrie de la pointe AFM utilisée, la surface analysée peut être ainsi reconstruite.

## 1.A.2 Les différents modes de mesure

Pour la description des différents modes de mesure, nous nous sommes inspirés du livre de [Bowen and Hilal \(2009\)](#). Il existe trois modes de mesure : le mode contact, le mode contact intermittent et le mode non contact. Ce sont des modes qui mesurent différents types d'interaction avec la matière, comme le montre le diagramme de la Figure 1.A.4. Chacun des modes permet d'obtenir des mesures précises pour certaines applications.

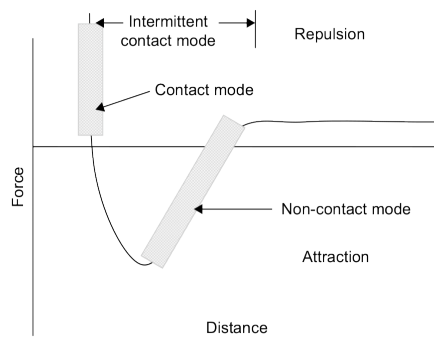


FIGURE 1.A.4 – Diagramme des forces mesurées en fonction des modes de mesure.

**Le mode contact** Le mode contact est historiquement le premier mode qui a été développé. Dans ce mode de mesure, la pointe est en contact avec la matière. En réalité, la pointe ne touche pas la matière mais est tellement proche qu'elle est repoussée par la surface. Les forces mesurées sont donc les forces répulsives.

Dans ce mode, l'utilisateur de l'AFM définit une valeur nommée « *setpoint* » qui correspond à une intensité de répulsion. L'AFM scanne la surface en essayant de maintenir l'interaction à la valeur de *setpoint* afin de reconstruire la topologie de la surface.



## 1.A. DESCRIPTION DU PROCÉDÉ DE MESURE PAR AFM

---

**Le mode contact intermittent** Ce mode est également appelé le mode *tapping*. C'est le mode le plus utilisé pour un AFM. Dans ce mode, la pointe subit une oscillation au dessus de la surface avec une fréquence proche de sa fréquence de résonance. La pointe est ensuite descendue et lorsqu'elle est proche de la surface, la pointe va, de manière récurrente, être attirée puis repoussée par la surface, ce qui modifie l'amplitude de l'oscillation. Lorsque la surface est scannée, les amplitudes des oscillations sont mesurées, ce qui permet de reconstruire la topologie de la surface.

**Le mode non contact** Dans le mode non contact, la pointe oscille comme pour le mode tapping mais avec une amplitude beaucoup plus faible. En s'approchant de la surface les atomes de la pointe et ceux de la surface vont entrer en interactions longues portées, comme les forces de Van Der Waals et forces électrostatiques. Ce sont ces changements dans l'amplitude de l'oscillation de la pointe qui sont mesurés. La pointe n'entre jamais en contact avec la surface lors du mode sans contact.

# Bibliographie

- Allende, H., Frery, A. C., Galbiati, J., and Pizarro, L. (2006). M-estimators with asymmetric influence functions : the distribution case. *Journal of Statistical Computation and Simulation*, 76(11) :941–956.
- Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., and Tukey, J. W. (1972). *Robust Estimates of Location : Survey and Advances*. Princeton University Press, Princeton, New Jersey.
- Bowen, W. and Hilal, N. (2009). *Atomic Force Microscopy in Process Engineering : An Introduction to AFM for Improved Processes and Products*. Butterworth-Heinemann/ICChemE series. Elsevier Science.
- Dervillé, A., Clausel, M., and Coeurjolly, J.-F. (2018). Asymmetric m - estimation and change-point analysis applied to atomic force microscopy data enhancement. *Metrologia*, 55(5) :637–644.
- Forsyth, D. A. and Ponce, J. (2002). *Computer Vision : A Modern Approach*. Prentice Hall Professional Technical Reference.
- Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2) :149–192.
- Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics - The Approach Based on Influence Functions*. Wiley.
- Huber, P. J. (1981). *Robust Statistics*. Wiley.
- Landis, S. (2010). *La nanolithographie*. Traité EGEM. : Série électronique et micro-électronique. Hermes Science Publications.
- Lingrand, D. (2008). *Introduction au traitement d'images*. Multimédia, graphisme. Vuibert, Paris.

## BIBLIOGRAPHIE

---

- Ma, S. and Yang, L. (2011). A jump-detecting procedure based on spline estimation. *Journal of Nonparametric Statistics*, 23(1) :67–81.
- Nistér, D. and Stewénus, H. (2008). Linear time maximally stable extremal regions. In *Lecture Notes in Computer Science*, pages 183–196. Springer Science + Business Media.
- van der Vaart, A. (2000). *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.
- Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.
- Wolke, R. (1992). Iteratively reweighted least squares : A comparison of several single step algorithms for linear models. *BIT*, 32(3) :506–524.
- Wolke, R. and Schwetlick, H. (1988). Iteratively reweighted least squares : Algorithms, convergence analysis, and numerical comparisons. *SIAM Journal on Scientific and Statistical Computing*, 9(5) :907–921.
- Yablon, D. (2013). *Scanning Probe Microscopy for Industrial Applications : Nanomechanical Characterization*. Wiley.

## Chapitre 2

# Agrégation d'estimateurs pour l'estimation de diamètres de particules

### Introduction

Le LNE a mis au point un protocole de mesure avec des repères sur les échantillons afin d'être en mesure de scanner la même zone avec les deux techniques de caractérisation, la microscopie à force atomique (AFM) et de microscopie électronique à balayage (MEB). La technique de microscopie MEB est présentée en Annexe [2.A](#). Ce chapitre a pour objectif de présenter le travail effectué autour de l'analyse simultanée d'images AFM et MEB.

Dans un premier temps, nous présenterons des méthodes d'extraction des nanoparticules pour chacune des deux techniques d'analyse. Nous présenterons pour la technique MEB comment nous arrivons à extraire des particules sphériques et distinguer les particules isolées des particules accumulées en agrégats. Une fois les particules extraites, nous présenterons également comment extraire le diamètre des particules.

Dans un second temps, nous allons présenter une technique de mise en correspondance entre image AFM et MEB afin d'exploiter au mieux cette fusion de données. Nous avons alors besoin de mettre en correspondance les particules qui sont alors co-localisées, via un identifiant. La technique que nous devons développer doit être robuste car les deux techniques ne représentent pas les particules exactement de la même manière, comme nous le verrons dans la suite de ce chapitre.

---

Une fois les diamètres des particules extraits par les deux techniques de mesure, avec ou sans correspondance des particules, nous obtenons alors deux estimations d'une même grandeur. Nous souhaitons alors combiner les deux estimations. La section 2.3 présente l'approche choisie pour l'agrégation d'estimateurs. Nous ferons tout d'abord une présentation du principe général de la méthode pour ensuite décrire son utilisation dans notre contexte applicatif.

Enfin, nous finirons avec une présentation de nos résultats issus de l'agrégation d'estimateurs de caractéristiques de particules provenant d'images MEB et d'AFM. Dans le cas de populations de particules extraites sur des images non co-localisées puis dans le cas d'images co-localisées.

## Sommaire

---

<b>Introduction</b> . . . . .	<b>83</b>
<b>2.1 Analyse d'images de nanoparticules</b> . . . . .	<b>85</b>
2.1.1 Analyse d'images MEB . . . . .	85
2.1.2 Images AFM . . . . .	90
<b>2.2 Recalage et correspondance des images</b> . . . . .	<b>93</b>
<b>2.3 Agrégation d'estimateurs</b> . . . . .	<b>98</b>
2.3.1 Présentation de la méthode . . . . .	98
2.3.2 Pratique/Implémentation . . . . .	100
2.3.3 Résultats agrégation . . . . .	102

---

## 2.1 Analyse d'images de nanoparticules

Nous souhaitons analyser les particules pour chacune des images AFM et MEB, afin d'en extraire différentes caractéristiques, qui serviront de base pour notre procédure d'estimation. Cette section présente les algorithmes développés pour l'extraction de nanoparticules dans les images et l'extraction de leur diamètre.

### 2.1.1 Analyse d'images MEB

Pour analyser les images MEB, nous avons implémenté deux modules :

1. Extraction d'objets dans l'image : on extrait les composantes connexes de l'image. Nous allons extraire les objets distincts de l'image, qui sont soit des agrégats de particules soit des particules isolées. Pour ce mode, on estime ensuite les caractéristiques des particules isolées. En effet, les agrégats étant considérés comme un seul objet et non comme un ensemble de particules primaires, nous ne pouvons en extraire de l'information.
2. Extraction des particules primaires : on décompose les objets en particules primaires. Les particules primaires sont les particules élémentaires/unitaires. Elles peuvent être seules, ce qui donne des particules isolées, soit se coller et former des agrégats. Une fois ces particules primaires extraites on estime les caractéristiques de chacune des particules.

Une étape de pré-traitement est à chaque fois nécessaire. En premier lieu, on enlève les lignes qui contiennent éventuellement une bannière afin qu'elles ne polluent pas l'analyse. En effet, les images MEB sont parfois annotées et on a alors un bandeau en bas de l'image. Le bandeau ou bannière contient des informations sur l'échelle et d'autres paramètres de la prise d'image.

Une fois la bannière retirée, nous appliquons un filtrage gaussien de taille  $3 \times 3$  afin de diminuer le bruit, permettant de supprimer une partie du bruit.

### Extraction des objets de l'image

Le premier algorithme que nous présentons permet d'extraire tous les objets présents dans l'image, c'est-à-dire à la fois les particules isolées et les agrégats. Ce sont les composantes connexes de l'image. Les principales étapes de l'algorithme, présentées sur le schéma de la Figure 2.1 sont :

1. **Binarisation** : on applique une binarisation par la méthode d' [Otsu \(1979\)](#) pour distinguer le fond de l'image des objets. La méthode

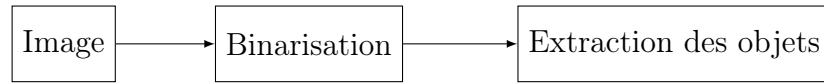
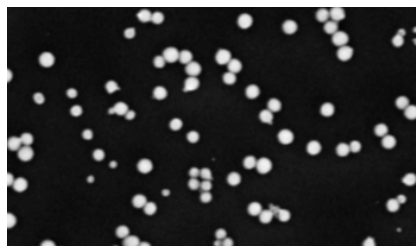


FIGURE 2.1 – Schéma de l'algorithme d'extraction des objets de l'image.

d'Otsu est toute désignée face à ces images car les objets sont distincts du fond et le fond est uniforme. Cela engendre un histogramme des niveaux de gris bimodal. La technique d'Otsu permet alors de bien séparer les deux classes et de trouver le seuil qui permet de binariser l'image de manière optimale.

2. **Nettoyage** : on applique une ouverture morphologique qui est une érosion morphologique suivie d'une dilatation morphologique, décrite dans le livre de [Najman and Talbot \(2008\)](#). L'ouverture morphologique va permettre de supprimer les éventuels pixels de bruit restants. L'image [2.2b](#) est le résultat de la binarisation nettoyée de l'image précédente [2.2a](#).
3. **Extraction des objets** : on applique une labellisation en composantes connexes, technique décrite dans le livre de [Haralick and Shapiro \(1993\)](#). Pour effectuer la labellisation en composantes connexes, on parcourt l'image binaire et lorsque le pixel courant est blanc, on applique un algorithme de remplissage de l'objet en partant du pixel courant. Un algorithme de remplissage est une technique qui permet de parcourir une composante connexe et de stocker les pixels parcourus et de labelliser cette composante. On pourra consulter le livre de [Soille \(2003\)](#) afin d'approfondir le sujet. À la fin de l'algorithme, chaque composante connexe est labellisée avec un identifiant unique et la liste des pixels de chaque composante contenue est sauvegardée.



(a) Image MEB filtrée



(b) Image binaire nettoyée

FIGURE 2.2 – Exemples de résultat de la binarisation par méthode d'Otsu d'une image MEB.

Ces différentes étapes nous permettent de récupérer les objets. Nous pou-

vons extraire ensuite les caractéristiques des objets identifiés. Nous avons choisi néanmoins de nous intéresser uniquement aux particules isolées. Les particules agrégées, sont en effet plus difficiles à analyser car elles peuvent se chevaucher. De plus, les agrégats sont considérés comme un unique objet pour l'instant, ce qui entraîne le besoin d'une analyse plus poussée pour extraire les particules qui le composent.

### Extraction des particules primaires

Le second algorithme que nous présentons permet d'extraire toutes les particules primaires présentes dans l'image. Le schéma de la Figure 2.3 décrit les principales étapes de cet algorithme.

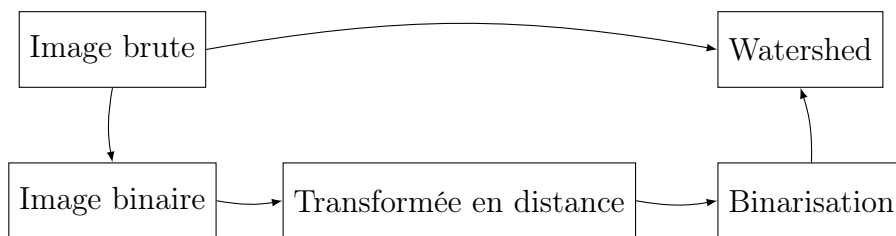


FIGURE 2.3 – Schéma de l'algorithme d'extraction des particules primaires de l'image.

Explications du schéma de la Figure 2.3 :

1. **Binarisation de l'image** : on applique le même algorithme que dans l'extraction des objets. Cette étape prend l'image 2.4a et retourne l'image 2.4b de la Figure 2.4.
2. **Transformée en distance et binarisation** : on applique une transformée en distance, voir l'article de [Borgefors \(1986\)](#), sur l'image binaire obtenue à l'étape précédente (image 2.4c). La transformée en distance transforme chaque pixel blanc en la distance discrète à l'ensemble complémentaire (c'est-à-dire l'ensemble des pixels noirs). En d'autres termes la transformée en distance va donner pour chaque pixel blanc la distance discrète à l'extérieur de l'objet le plus proche. On binarise le résultat de la transformée en distance en effectuant pour chaque composante connexe de l'image une binarisation avec un seuil à 70% de la valeur maximale de la composante (image 2.4d). Cette étape prend l'image 2.4b, calcule 2.4c et retourne 2.4d. Cette étape va permettre de séparer les agrégats en plus petits objets que sont les particules. Elles sont sous segmentées, c'est à dire que le masque binaire est plus petit que les particules réelles. Nous étiquetons en composante

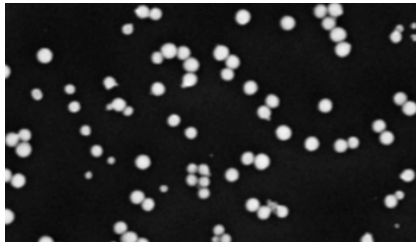


## 2.1. ANALYSE D'IMAGES DE NANOPARTICULES

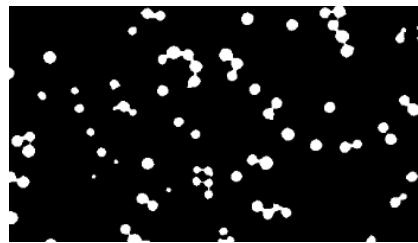
---

connexe l'image binaire résultante. Ainsi chaque composante servira de source pour l'étape suivante.

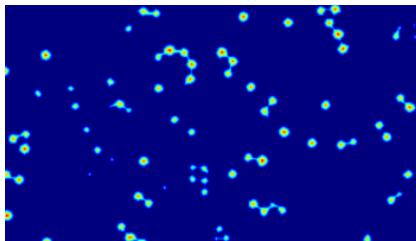
3. **Watershed** : On prend l'image d'origine et l'image étiquetée de la transformée en distance pour initialiser un algorithme de segmentation par ligne de partage des eaux couramment appelé watershed, présenté dans l'article de [Beucher and Meyer \(1992\)](#). Le principe de cet algorithme est de considérer l'intensité des pixels comme une information de hauteur. En utilisant des marqueurs couramment appelés sources, l'algorithme va remplir progressivement d'eau chaque bassin du terrain jusqu'à que le terrain soit complètement rempli en enregistrant les zones de séparation où les eaux se rencontrent. Chaque bassin représente une région et nous obtenons une image segmentée. Dans notre cas, l'algorithme va utiliser l'image étiquetée de la transformée en distance puis compléter les objets dans l'image. Cette étape prend [2.4a](#) et [2.4d](#) et retourne [2.4e](#). L'image [2.4e](#) représente l'étiquetage sortant de l'algorithme de watershed, si on prend le contour des composantes, nous obtenons l'image [2.4f](#).



(a) Image brute



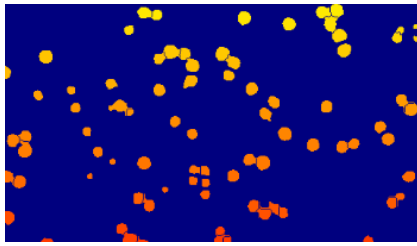
(b) Image binaire érodée



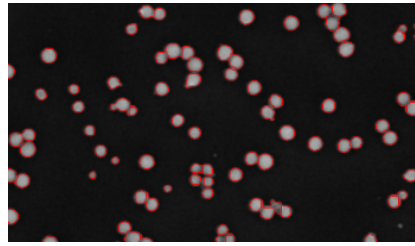
(c) Résultat de la transformée en distance de l'image [2.4b](#)



(d) Transformée en distance [2.4c](#) binarisée



(e) Résultat de la segmentation par algorithme de watershed



(f) Résultat contours watershed

FIGURE 2.4 – Exemples de résultat de la segmentation d'images MEB par algorithme de watershed

Pour l'extraction des particules primaires dans l'image, il faut lire l'image de sortie des watershed, ce qui donne les points de chacune des composantes (particules primaires). Nous pouvons ensuite analyser les caractéristiques des particules primaires.

### Extraction du diamètre des particules

L'étape précédente nous permet d'avoir la position des particules par une segmentation de l'image en composantes connexes. Maintenant nous devons extraire une mesure du diamètre de chaque nanoparticule de cette segmentation. Pour cela, nous prenons la composante connexe de chaque particule, localement, nous analysons le signal. Nous allons extraire le signal à mi-hauteur de la nanoparticule, comme indiqué dans l'article de [Delvallée et al. \(2015\)](#), qui permet d'obtenir la mesure qui est la meilleure estimation du diamètre.

Pour cela, nous prenons le signal local, calculons la hauteur de la particule, via le masque de segmentation. Nous prenons ensuite la hauteur moyenne du fond de l'image. Nous calculons alors la mi-hauteur de la nanoparticule courante via les deux grandeurs extraites précédemment.

Une fois le signal local binarisé via ce nouveau seuil, nous calculons le diamètre des particules en prenant le rayon du cercle correspondant à l'aire discrète obtenue par la binarisation. Pour cela, nous utilisons des distances discrètes, présentées dans l'ouvrage de [Coeurjolly et al. \(2007\)](#). Nous avons alors un estimateur du diamètre de la particule. Cette version n'est valable que pour des particules sphériques, il existe beaucoup d'autre façon de caractériser les dimensions d'une nanoparticule comme par exemple le diamètre de Feret. Dans notre cas, nous nous concentrerons uniquement sur des particules sphériques car notre objectif est de pouvoir combiner deux techniques, l'AFM et le MEB et nous verrons dans la section suivante que la dimension

## 2.1. ANALYSE D'IMAGES DE NANOPARTICULES

---

que l'on peut extraire par AFM est la hauteur de la particule. Nous pouvons alors comparer le diamètre obtenu par MEB à la hauteur obtenue par AFM du fait de la sphéricité des particules.

### 2.1.2 Images AFM

Dans le cas des images AFM, nous pouvons extraire la hauteur des particules. Pour cela, nous devons dans un premier temps extraire les objets, puis lire la hauteur. La hauteur des objets doit être calculée en soustrayant la hauteur du substrat, avec comme incertitude la rugosité de ce dernier. Pour cela, nous avons mis en place l'algorithme d'analyse présenté sur la Figure 2.5.

#### Extraction des particules

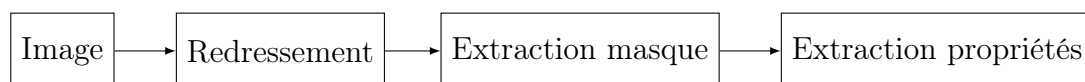


FIGURE 2.5 – Schéma de l'algorithme d'extraction des objets d'une image AFM.

- **Redressement** : application de l'algorithme décrit dans le Chapitre 1. On peut observer un exemple d'image de nanoparticules redressée sur la Figure 2.6.

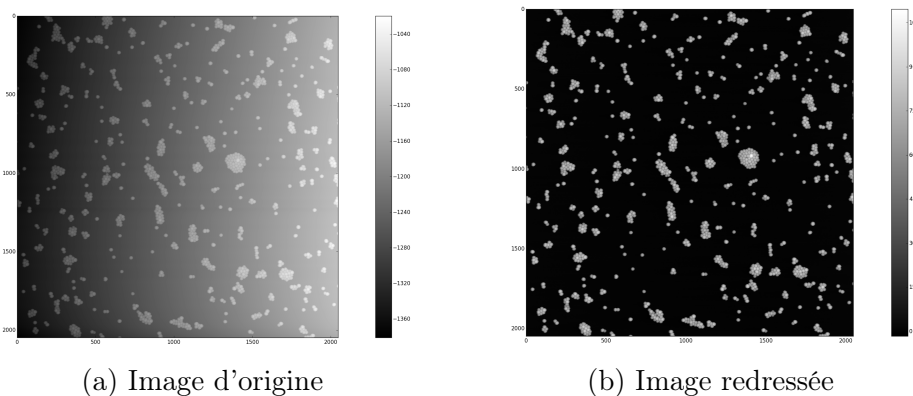


FIGURE 2.6 – Exemple d'image AFM et du résultat de son redressement. Point de départ de l'analyse d'image AFM.

- **Extraction Masque** : application d'une binarisation en extrayant le seuil par la méthode de Otsu. Ainsi on obtient un masque binaire qui sépare les objets du fond. Un exemple de masque binaire est présenté sur la Figure 2.6

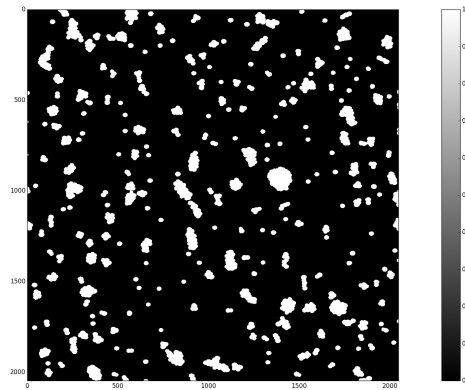


FIGURE 2.7 – Exemple d'un résultat de binarisation pour l'image 2.6

- **Extraction particules** : On applique un étiquetage en composante connexe afin d'extraire individuellement chaque particule ou agrégat. Afin d'obtenir des particules primaires, on peut appliquer le même algorithme que pour les images MEB (schéma de la Figure 2.3)

### Extraction des caractéristiques des particules

Commençons par quelques définitions à propos des matériaux.

**Définition de propriétés du matériau** Les propriétés surfaciques sont extraites de la norme ISO 25178-2 (2012), qui définit les spécifications géométriques des produits.

**Rugosité** : La rugosité  $S_q$  d'un matériau est définie comme l'écart-type de la distribution des hauteurs du substrat.

**Hauteur** La hauteur de la surface  $S_a$  est définie comme la moyenne arithmétique des hauteurs du substrat.

**Analyse globale** : Dans cette analyse, la hauteur et la rugosité de la surface sont calculées à partir du masque estimé dans l'étape précédente. Ce masque sert à récupérer la population du fond de l'image, c'est-à-dire du substrat. Ensuite, nous pouvons calculer la hauteur de chaque particule.

## 2.1. ANALYSE D'IMAGES DE NANOPARTICULES

---

Pour cela, on parcourt les composantes connexes de l'image. On cherche la valeur maximale de chaque composante puis sa hauteur est calculée comme la valeur maximale à laquelle on a retranché la hauteur du substrat  $S_a$ .

**Analyse locale :** Dans cette analyse, nous calculons la hauteur de chaque particule en utilisant l'information locale en chaque pixel de l'image. Pour cela, nous considérons une zone autour de chaque particule, de dimension égale à trois fois la dimension de la boîte englobante de la particule. Sur cette sous image, nous calculons la hauteur ainsi que la rugosité locale du substrat autour de la particule. La hauteur de l'objet est alors estimée en retranchant à la hauteur maximale de l'objet la hauteur locale du substrat.

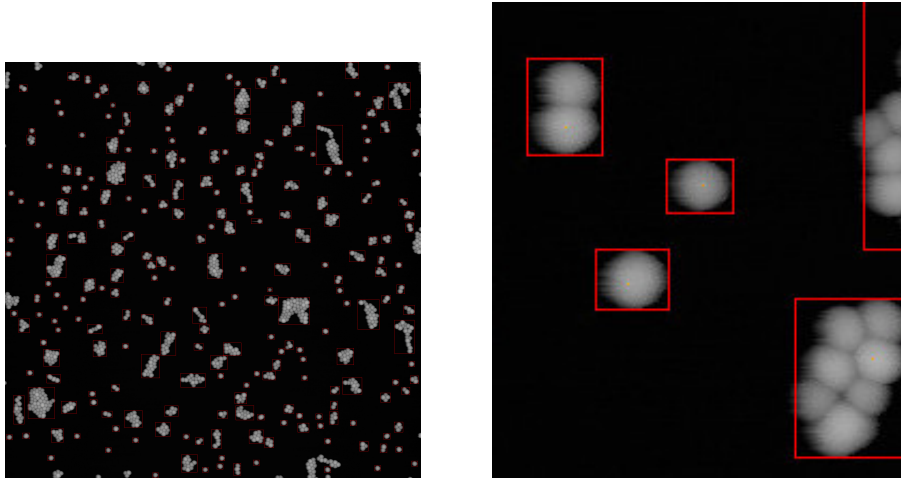


FIGURE 2.8 – Exemple de résultat d'analyse d'image AFM. En rouge les boîtes englobantes des particules et en orange, le point le plus haut de la particule/agrégat. À gauche l'image complète et à droite une sous partie de l'image afin de mieux visualiser les tracés graphiques.

L'incertitude sur la hauteur de la particule est par définition la rugosité du substrat. En effet, la rugosité représente les irrégularités du substrat, la particule étant posée sur ce substrat, nous ne connaissons donc pas la véritable hauteur de substrat sur laquelle repose la particule. Nous en avons juste une estimation dont l'analyse locale de la hauteur des particules et du substrat permet de fournir une plus grande précision pour la hauteur de chacune des particules même si le matériau n'est pas régulier.

## 2.2 Recalage et correspondance des images

Dans cette section, nous présentons une technique de mise en correspondance des particules AFM avec les particules MEB. Dans ce cadre, nous avons des images de la même zone provenant des deux microscopes. Durant cette section, nous travaillerons sur un patchwork d'images MEB et une image AFM couvrant la même zone, respectivement présentées sur les Figures 2.1 et 2.2.

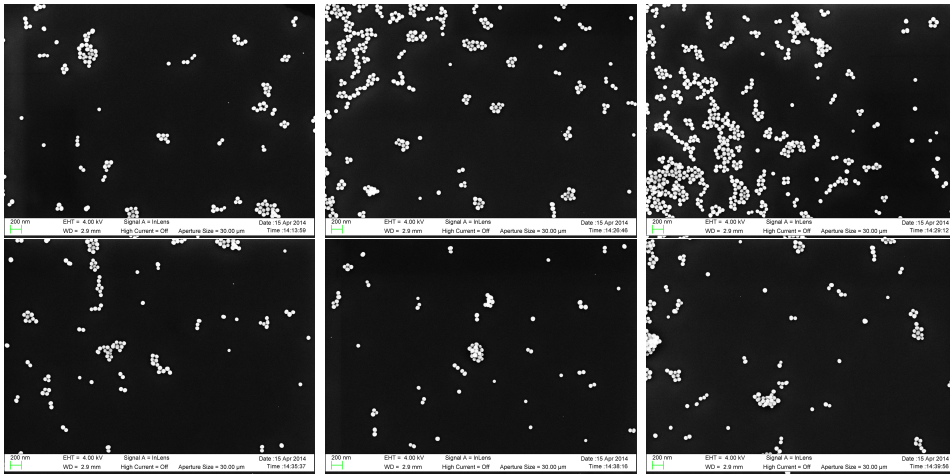


FIGURE 2.1 – Exemple patchwork image MEB pour la mise en correspondance avec l'image AFM présentée sur la Figure 2.2.

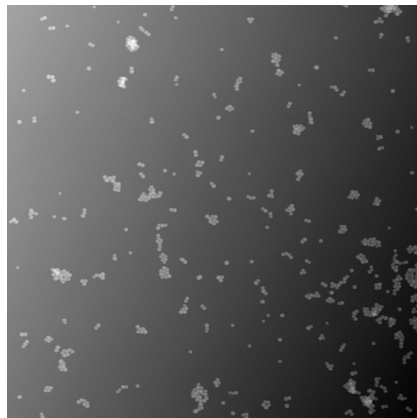


FIGURE 2.2 – Exemple d'une image AFM brute pour la mise en correspondance avec le patchwork d'image MEB présenté sur la Figure 2.1.

L'image AFM ayant une résolution planaire inférieure à celle de l'image MEB, nous avons plusieurs images MEB pour la même superficie que l'image

## 2.2. RECALAGE ET CORRESPONDANCE DES IMAGES

AFM. L'objectif sera de trouver, pour chacune des images MEB, la transformation qui permet de passer du système de coordonnées de l'image MEB à celui de l'image AFM. Une fois cette transformation déterminée, vient un appariement pour les particules. La Figure 2.3 présente le schéma du fonctionnement de l'algorithme de recherche de correspondances.

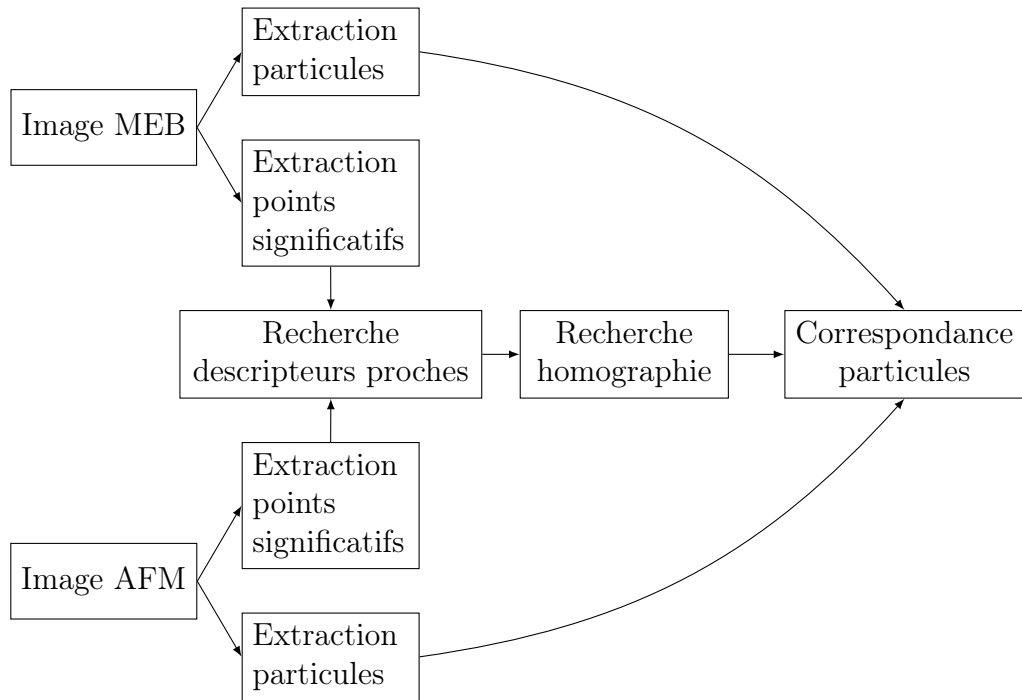


FIGURE 2.3 – Schéma du fonctionnement de l'algorithme de mise en correspondance des particules.

La transformation que nous cherchons est une homographie. Il s'agit d'une transformation linéaire entre deux plans. Les homographies sont des transformations classiques pour la création de mosaïques d'images ou bien de panorama, comme l'illustre la publication [Shum and Szeliski \(1997\)](#). Nous avons choisi une homographie car le positionnement des particules peut subir une déformation non linéaire entre l'image AFM et l'image MEB. Afin de trouver l'homographie et de mettre en correspondance les particules, nous appliquons les étapes présentées dans le schéma de la Figure 2.3, que nous détaillerons dans cette section.

**Étape préliminaire : Préparation des images** Avant de rechercher les transformations pour la mise en correspondance des images, nous préparons

les données. Pour cela, nous appliquons un redressement pour l'image AFM afin d'avoir une surface plane avec les objets posés dessus (décrit dans le Chapitre 1).

Nous effectuons ensuite un redimensionnement de l'image MEB pour avoir la même taille de pixel, ce qui permet d'être indépendant de l'échelle. En effet, l'échelle MEB est beaucoup plus petite en surface que l'image AFM mais est plus grande en nombre de pixels. Une fois le redimensionnement appliqué, les particules en pixel sont sensiblement de même taille. Sensiblement car les dimensions latérales des particules en AFM intègrent la largeur de la pointe AFM (effet décrit dans l'Annexe 1.A du Chapitre 1). Les motifs formés par les particules auront alors la même forme et sensiblement la même dimension. On peut alors chercher la transformation qui permet de passer de l'image MEB à l'image AFM.

**Recherche de la transformation** Nous extrayons des points significatifs dans les images en utilisant des descripteurs de KAZE, décrit par [Alcantarilla et al. \(2012\)](#), dans l'image MEB et dans l'image AFM, présentées sur la Figure 2.4. Les points significatifs sont composés de plusieurs valeurs autres que la position comme les changements de couleur à différentes échelles, les directions de gradients et d'autres informations, qui représentent des propriétés locales de l'image. Les descripteurs contiennent donc une description de la région autour du point où ils sont calculés.

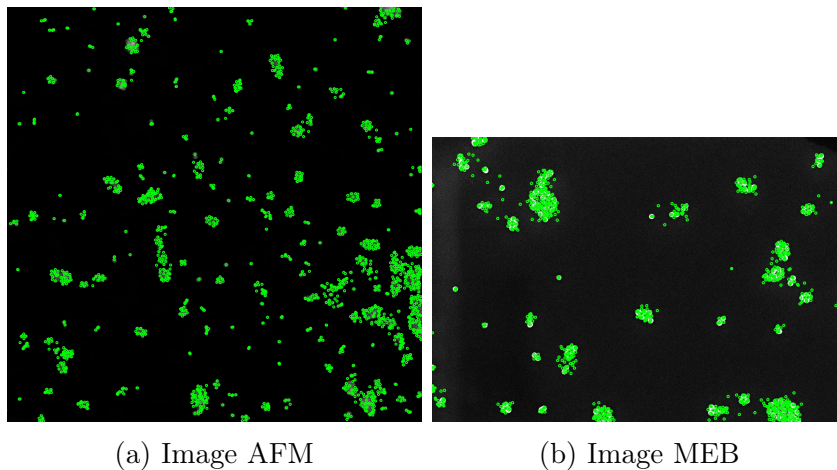


FIGURE 2.4 – Exemple de points significatifs de KAZE extraits sur l'image MEB et sur l'image AFM, dont la position est marquée par des cercles verts sur les images.

À partir des points représentatifs, il faut sélectionner les points ayant



## 2.2. RECALAGE ET CORRESPONDANCE DES IMAGES

---

le plus de correspondance avant de chercher la transformation. Pour cela, nous appliquons la méthode décrite dans l'article de [Muja and Lowe \(2009\)](#). Pour déterminer si deux points significatifs sont proches, nous appliquons une norme  $L^2$  entre les deux ensembles de points significatifs et créons des couples en prenant pour chaque point significatif le plus proche voisin dans l'ensemble des points caractéristiques de l'autre image. Nous prenons alors l'ensemble des couples formés et ne gardons que ceux avec une distance assez faible. Un exemple de résultat d'appariement de points significatif est présenté sur la Figure 2.5.

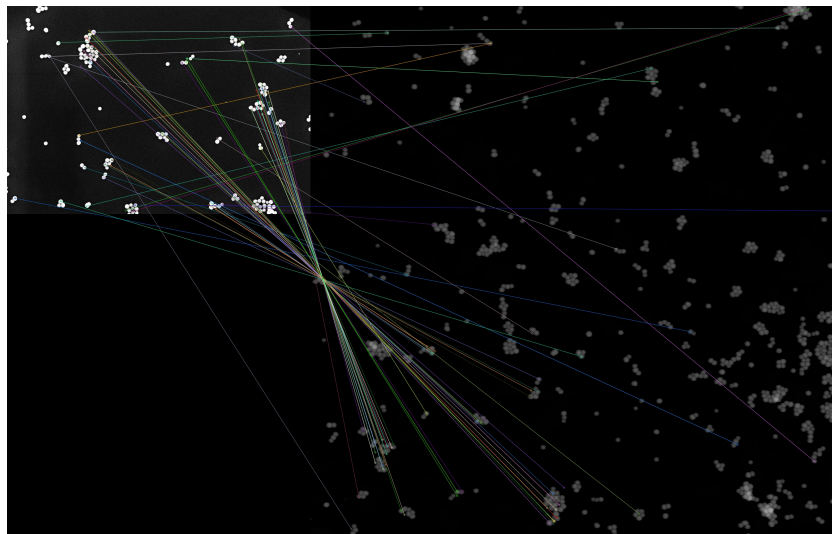


FIGURE 2.5 – Exemple de résultat de correspondance des points significatifs, par plus proche voisin. Dans le haut à gauche de la figure, l'image MEB est tracée et dans la partie droite de la figure, l'image AFM. Les traits représentent les points considérés comme voisins en utilisant la méthode décrite dans l'article de [Muja and Lowe \(2009\)](#).

En utilisant la liste réduite des couples, nous allons chercher l'homographie qui permet de transformer les positions des points de l'image MEB en ceux de l'image AFM. Une représentation des correspondances est présentée sur la Figure 2.6. Nous utilisons alors un algorithme de RANSAC, décrit dans le livre de [Forsyth and Ponce \(2011\)](#), pour chercher l'homographie qui explique la plus grande partie des transformations des couples de points significatifs. Le RANSAC fonctionne de la manière suivante :

1. Tirage aléatoire de  $k$  éléments dans l'ensemble de données.
2. Calcul du modèle correspondant aux observations.
3. Calcul de l'erreur du modèle par rapport aux observations.

4. Refaire les opérations 1 à 3  $N$  fois et retenir le modèle générant l'erreur minimum.

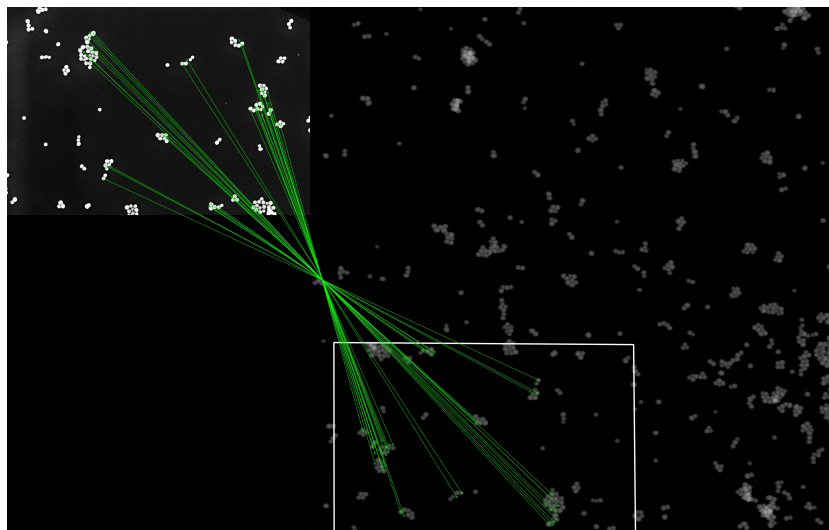


FIGURE 2.6 – Exemple de résultat de correspondance des points significatifs entre une image MEB (en haut à gauche) et une image AFM (à droite). Les traits représentent les liens entre points significatifs correspondant entre les deux images.

Partant de cette homographie, nous sommes en mesure de connaître l'ensemble des pixels de l'image AFM correspondants à ceux de l'image MEB, dont les limites sont présentées sur l'image de la Figure 2.6.

**Appariement des particules** Pour cela, nous extrayons les particules dans les images MEB et l'image AFM via les techniques décrites dans la section 2.1. Deux options sont disponibles, soit on prend en compte uniquement les particules isolées, soit on prend en compte toutes les particules, ce qui implique de segmenter les agrégats en AFM, qui peut être fait de la même manière que pour les particules MEB ( voir la section 2.1.1). Nous transformons ensuite les coordonnées des particules MEB, en utilisant la matrice de transformation calculée selon la technique décrite précédemment, puis nous recherchons la particule AFM la plus proche, qui est alors la particule correspondante.

## 2.3 Agrégation d'estimateurs

Nous avons extrait des sections précédentes des populations de diamètres de particules, extraites par MEB et AFM avec ou sans correspondance. Ces différentes populations sont des estimations de la même grandeur. Dans cette section, nous expliquerons la méthode d'agrégation d'estimateurs proposée récemment par [Lavancier and Rochet \(2016\)](#). Ceci va nous permettre d'estimer de manière plus précise les différentes caractéristiques de nos nanoparticules en exploitant, en outre, simultanément les diamètres extraits par MEB et AFM dans les différentes configurations possibles.

### 2.3.1 Présentation de la méthode

Supposons que nous disposons d'un certain nombre de méthodes différentes capables d'estimer un paramètre réel ou vectoriel d'un modèle statistique. L'objectif de l'agrégation d'estimateurs est de fournir la meilleure combinaison linéaire de ces estimateurs. On se restreindra au choix de combinaison linéaire de ceux-ci. La recherche d'une telle combinaison doit être faite relativement à un critère. Nous nous intéresserons ici à la minimisation de l'erreur quadratique moyenne.

Le problème remonte au 19<sup>ème</sup> siècle, par exemple [Laplace \(1812\)](#) cherchait à trouver la meilleure combinaison de la moyenne et de la médiane, son objectif étant d'estimer le paramètre de position d'une distribution symétrique.

#### Cas d'un unique paramètre réel

Soit  $\theta \in \mathbb{R}$  une quantité inconnue d'un modèle statistique et  $\mathbf{T} = (T_1, \dots, T_n)^\top$ , avec  $n \geq 2$ , une collection d'estimateurs de  $\theta$ , pas nécessairement calculés sur le même jeu de données. L'objectif est de trouver la meilleure combinaison linéaire des estimateurs  $T_i$  afin d'avoir un estimateur plus efficace, tout en gardant les avantages de chaque estimateur. L'estimateur que l'on recherche va être de la forme :

$$\hat{\theta}_\lambda = \lambda^\top \mathbf{T} = \sum_{i=1}^n \lambda_i T_i,$$

où  $\lambda = (\lambda_1, \dots, \lambda_n)^\top$  est un vecteur de poids. Pour la recherche des poids, nous nous restreindrons au sous-ensemble  $\Lambda = \{\lambda \in \mathbb{R}^n : \sum_{i=1}^n \lambda_i = 1\}$ . Le choix de restreindre l'ensemble des solutions est un choix de contraintes largement utilisé dans la littérature, proposé dans l'article de [Bates and Granger \(1969\)](#). Une explication de la raison de cette contrainte minimum est proposée dans l'article de [Lavancier and Rochet \(2016\)](#).

Si les  $T_i$  sont de carré intégrable alors le vecteur de poids  $\lambda^*$  qui minimise le risque quadratique peut s'exprimer en fonction de la matrice de covariance des estimateurs  $\Sigma$  où  $\Sigma_{i,j} = \mathbb{E}[(T_i - \theta)(T_j - \theta)]$ ,  $i, j = 1, \dots, n$ . Si l'on tient compte de la condition  $\sum_{i=1}^n \lambda_i = 1$  l'erreur quadratique de  $\hat{\theta}_\lambda$  se simplifie en :

$$\mathbb{E}(\hat{\theta}_\lambda - \theta)^2 = \mathbb{E}\left(\sum_{i=1}^n \lambda_i (T_i - \theta)\right)^2 = \lambda^\top \Sigma \lambda$$

La meilleure combinaison linéaire  $\lambda^*$  qui minimise l'erreur quadratique moyenne est la solution du problème d'optimisation sous contrainte :

$$\lambda^* = \operatorname{argmin}_{\lambda \in \Lambda} \lambda^\top \Sigma \lambda = \frac{\Sigma^{-1} \mathbf{1}}{\mathbf{1}^\top \Sigma^{-1} \mathbf{1}}$$

avec  $\mathbf{1} = (1, \dots, 1)^\top \in \mathbb{R}^n$ . On obtient alors l'estimateur dit oracle suivant :

$$\hat{\theta}^* := \sum_{i=1}^n \lambda_i^* T_i$$

En pratique, l'oracle n'est pas connu mais peut être approché en calculant une estimation  $\hat{\Sigma}$  de la matrice de covariance. L'estimateur agrégé  $\hat{\theta}$  est alors construit en remplaçant  $\Sigma$  par son estimation  $\hat{\Sigma}$  dans l'expression de  $\lambda^*$ . On a alors :

$$\hat{\theta} = \hat{\theta}_{\hat{\lambda}} \quad \text{avec} \quad \hat{\lambda} = \frac{\hat{\Sigma}^{-1} \mathbf{1}}{\mathbf{1}^\top \hat{\Sigma}^{-1} \mathbf{1}}.$$

### Cas où l'on estime plusieurs paramètres réels

L'objectif est d'étendre la méthode afin d'estimer plusieurs paramètres réels simultanément. Supposons  $\theta = (\theta_1, \dots, \theta_d)^\top \in \mathbb{R}^d$  avec pour chaque élément  $\theta_j$  une collection d'estimateurs  $\mathbf{T}_j$  de taille  $k_j \geq 1$ . Cela revient à supposer que  $\mathbf{T}_1 \in \mathbb{R}^{k_1}, \dots, \mathbf{T}_d \in \mathbb{R}^{k_d}$  et  $\mathbf{T} = (\mathbf{T}_1^\top, \dots, \mathbf{T}_d^\top)^\top$  avec  $k = \sum_{j=1}^d k_j \geq d$

L'estimateur agrégé du vecteur  $\theta$  s'exprime alors de la façon suivante :

$$\hat{\theta}_\lambda = \lambda^\top \mathbf{T} \in \mathbb{R}^d,$$

où  $\lambda$  est une matrice de taille  $k \times d$ . Afin d'avoir un estimateur qui est sans biais, on va supposer que notre estimateur  $T$  est défini à partir d'un vecteur de poids  $\lambda$  qui appartient à l'ensemble de contraintes suivantes :

$$\Lambda_{max} = \{\lambda \in \mathbb{R}^{k \times d} : \lambda^\top J = I\}$$

où

$$J = \begin{pmatrix} \mathbf{1}_{k_1} & 0 & \cdots & 0 \\ 0 & \mathbf{1}_{k_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{1}_{k_d} \end{pmatrix}$$

et  $I$  désigne la matrice identité.

Comme précédemment, on cherche à minimiser l'erreur quadratique :

$$\begin{aligned} \mathbb{E}\|\hat{\theta}_\lambda - \theta\|^2 &= \mathbb{E}\|\lambda^\top \mathbf{T} - \theta\|^2 \\ &= \mathbb{E}\left[\text{tr}\left[(\mathbf{T} - J\theta)^\top \lambda \lambda^\top (\mathbf{T} - J\theta)\right]\right] \\ &= \text{tr}(\lambda^\top \Sigma \lambda). \end{aligned}$$

L'estimateur agrégé  $\hat{\theta} = \hat{\lambda}^\top \mathbf{T}$  est obtenu en résolvant le problème inverse :

$$\hat{\lambda} = \underset{\lambda \in \Lambda}{\text{argmin}} \text{tr}(\lambda^\top \hat{\Sigma} \lambda).$$

Le papier de [Lavancier and Rochet \(2016\)](#) donne davantage de détails et notamment explicite des propriétés non asymptotiques et asymptotiques de ces procédures.

### 2.3.2 Pratique/Implémentation

Afin d'appliquer la procédure d'agrégation que nous avons décrite, nous avons besoin de calculer la matrice de covariance des estimateurs. Pour estimer cette matrice de covariance, nous allons utiliser la technique de rééchantillonnage par bootstrap, présentée par exemple dans le livre de [Davison and Hinkley \(1997\)](#), qui va nous permettre, à partir d'une population dont on ne connaît pas la loi, d'estimer la variance d'un estimateur. Dans le cas où la loi des estimateurs serait connue ou la matrice de covariance aurait une forme paramétrique, d'autres techniques pourraient être mises en place.

#### Principe du bootstrap

Pour la description du principe de bootstrap, nous nous sommes inspirés des notes de cours de [Coeurjolly and Drouilhet \(2007\)](#).

Soit  $\mathbf{Y} = (Y_1, \dots, Y_n)$  un vecteur de données, où les  $Y_i$  suivent tous la même loi  $\mathcal{L}_Y$  qu'une variable aléatoire  $Y$  donnée. On s'intéresse à la loi de probabilité inconnue de  $T = \mathcal{T}(\mathbf{Y})$ . Nous disposons d'une réalisation  $\mathbf{y} = (y_1, \dots, y_n)$  du vecteur aléatoire  $\mathbf{Y}$ , et connaissons  $t = \mathcal{T}(\mathbf{y})$ . L'objectif

va être de capter la variabilité de  $T$  ou, autrement dit, de déterminer une estimation  $\widehat{\mathcal{L}}_{T\mathbf{y}}$  de la loi  $\mathcal{L}_T$  de  $T$ .

Une estimation de la loi de  $\mathcal{L}_Y$  est la loi empirique  $\widehat{\mathcal{L}}_Y(y)$  des  $y_1, \dots, y_n$ , à savoir la loi d'équiprobabilité sur  $(y_1, \dots, y_n)$ . Le principe du bootstrap va consister à remplacer  $Y$  par  $Y^* = y_U$  avec  $U \rightsquigarrow \mathcal{U}(\{1, \dots, n\})$ . La loi  $\mathcal{L}_Y$  est ainsi remplacée par  $\mathcal{L}_Y^* = \mathcal{L}_{Y^*}$  correspondant à la loi empirique de tous les jeux de données différents  $y_1^*, \dots, y_n^*$ . L'estimation  $T = \mathcal{T}(Y)$  est alors approchée par  $T^* = \mathcal{T}(Y^*)$ .

La loi  $\mathcal{L}_T$  est remplacée par la loi  $\mathcal{L}_T^* = \mathcal{L}_{T^*}$  correspondant à la loi empirique de toutes les réalisations  $t_1^*, \dots, t_n^*$  de la statistique  $T^*$  calculées en appliquant à tous les jeux de données  $y_1^*, \dots, y_n^*$  (i.e. tous les éléments de l'ensemble  $(y)^n$ ) la transformation  $\mathcal{T}$  (i.e.  $t_j^* = \mathcal{T}(y_j^*)$ , pour  $j = 1, \dots, n$ ).

Lorsqu'il n'est pas possible d'obtenir tous les jeux de données, l'approche expérimentale des probabilités, appelée couramment approche Monte Carlo, nous propose de répéter un grand nombre  $m$  de fois l'expérience aléatoire consistant à choisir au hasard  $n$  éléments avec remise parmi les  $y_1, \dots, y_n$  et ainsi d'obtenir  $m$  réalisations  $y_{[1]}^*, \dots, y_{[m]}^*$  de  $Y^*$ . La loi  $\mathcal{L}_Y$  est remplacée par  $\mathcal{L}_Y^{*,m} = \mathcal{L}_{Y^*}^m$  correspondant à la loi empirique des  $m$  jeux de données  $y_{[1]}^*, \dots, y_{[m]}^*$ . Finalement, la loi  $\mathcal{L}_T$  est remplacée par  $\mathcal{L}_T^{*,m} = \mathcal{L}_{T^*}^m$  correspondant à la loi empirique des  $t_{[1]}^*, \dots, t_{[m]}^*$  obtenue à partir des  $y_{[1]}^*, \dots, y_{[m]}^*$ .

En résumé, dans notre cas, nous avons un échantillon  $X = (x_1, \dots, x_n)$  de variables supposées indépendantes et identiquement distribuées selon une loi. Nous allons simuler  $B$  échantillons de taille  $n$  en tirant aléatoirement les valeurs du vecteur  $X$ . En effet, on va générer  $X_b^* = (x_{1b}^*, \dots, x_{nb}^*)_{b=1\dots B}$ ,  $B$  échantillons sélectionnés de manière indépendante. Nous appliquerons donc l'approche Monte Carlo, afin d'estimer la matrice de covariance des estimateurs sur la population  $X$ .

Dans notre cas pratique, nous allons avoir un vecteur aléatoire avec plusieurs estimateurs ou bien plusieurs vecteurs aléatoires avec un ou plusieurs estimateurs. Dans le cas de plusieurs vecteurs aléatoires, nous allons considérer plusieurs cas :

- Soit les vecteurs aléatoires sont des sous-échantillons indépendants suivant la même loi. Alors nous appliquons le bootstrap indépendamment aux vecteurs aléatoires, afin de calculer la matrice de covariance des estimateurs.
- Soit les vecteurs aléatoires consistent en plusieurs observations des mêmes objets. Dans ce cas, la procédure de bootstrap doit être la même pour les différents vecteurs aléatoires. Pour cela, nous simulons  $(I_{1b}^*, \dots, I_{nb}^*)_{b=1\dots B}$ , une carte de tirages à appliquer, ce qui nous

## 2.3. AGRÉGATION D'ESTIMATEURS

---

permettra de générer  $x_{ib}^* = x_{I_{ib}}$ . Ainsi, le calcul de la matrice de covariance est correct, car les correspondances entre éléments sont prises en compte.

### Calcul de l'intervalle de confiance

Afin d'avoir une validation de nos résultats, nous avons choisi de calculer l'intervalle de confiance de différents paramètres. Pour cela, nous avons utilisé les techniques de calcul d'incertitude par bootstrap, présentées dans le chapitre 5 du livre de [Davison and Hinkley \(1997\)](#). Nous avons choisi trois manières différentes pour constituer un intervalle de confiance par bootstrap :

- basic bootstrap où  $IC = (2\theta - \theta_{(1-\alpha/2)}^*; 2\theta - \theta_{(\alpha/2)}^*)$  où  $\theta_{(1-\alpha/2)}^*$  représente le quantile d'ordre  $1 - \alpha/2$  des coefficients bootstrappés  $\theta^*$ . Cette technique nécessite un unique bootstrap pour calculer les  $\theta^*$ .
- percentile où  $IC = (\theta_{(\alpha/2)}^*; \theta_{(1-\alpha/2)}^*)$  où  $\theta_{(1-\alpha/2)}^*$  représente le quantile d'ordre  $1 - \alpha/2$  des coefficients bootstrappés  $\theta^*$ . Cette technique nécessite un unique bootstrap pour calculer les  $\theta^*$ .
- studentized où  $IC = (\theta - t_{(1-\alpha/2)}^* \cdot \hat{s}e_\theta; \theta - t_{(\alpha/2)}^* \cdot \hat{s}e_\theta)$  où  $t_{(1-\alpha/2)}^*$  représente le quantile d'ordre  $1 - \alpha/2$  du t-test de student bootstrappé  $t^* = (\hat{\theta}^* - \hat{\theta}) / \hat{s}e_{\hat{\theta}^*}$ , avec  $\hat{s}e_\theta$  l'écart-type des coefficients de la population d'origine. Cette technique nécessite un double bootstrap afin de calculer la statistique studentisée.

### 2.3.3 Résultats agrégation

#### Présentation des estimateurs étudiés

Comme expliqué dans la section 2.1, dans notre application nous mesurons les diamètres de particules sur des images MEB et AFM. Nous supposons que les réalisations de diamètres de particules sont issues d'une loi symétrique. Notre objectif sera d'estimer les paramètres de position et d'échelle. Pour cela, nous avons les informations suivantes :

$$\begin{aligned} \mathbf{Y}^{MEB} &= (Y_1^{MEB}, \dots, Y_{n_{MEB}}^{MEB}) \\ \mathbf{Y}^{AFM} &= (Y_1^{AFM}, \dots, Y_{n_{AFM}}^{AFM}) \\ \mathbf{Y}^{COL,MEB} &= (Y_1^{COL,MEB}, \dots, Y_{n_{COL}}^{COL,MEB}) \\ \mathbf{Y}^{COL,AFM} &= (Y_1^{COL,AFM}, \dots, Y_{n_{COL}}^{COL,AFM}) \end{aligned}$$

$\mathbf{Y}^{COL,MEB}$  et  $\mathbf{Y}^{COL,AFM}$  modélisent les diamètres des  $n_{COL}$  particules co-localisées dans les images MEB et AFM. Le vecteur  $\mathbf{Y}^{MEB}$  (respectivement  $Y^{AFM}$ ) modélise les diamètres des particules issues des images MEB

(respectivement AFM) et non présentes dans les images AFM (respectivement MEB). Il est clair que  $\mathbf{Y}^{COL,MEB}$  et  $\mathbf{Y}^{COL,AFM}$  ne sont pas indépendants. Comme l'illustre l'étude en simulation 2.B placée en annexe de ce chapitre, il est primordial de tenir compte de cette possible dépendance des échantillons dans l'estimation de la matrice de covariance par exemple de  $(\bar{\mathbf{Y}}^{COL,MEB}, \bar{\mathbf{Y}}^{COL,AFM})$

Pour l'estimation du paramètre de position, nous étudierons les estimateurs moyenne, médiane et moyenne tronquée à 10%.

Pour l'estimation du paramètre d'échelle, nous étudierons les estimateurs écart-type,  $S_n$  et  $Q_n$ . Les estimateurs  $S_n$  et  $Q_n$  sont des estimateurs robustes pour l'estimation du paramètre d'échelle, présentés par [Croux and Rousseeuw \(1992\)](#). Pour un échantillon  $\{x_1, x_2, \dots, x_n\}$ , l'estimateur  $S_n$  est défini par :

$$S_n = c \operatorname{med}_i \{ \operatorname{med}_j |x_i - x_j| \}$$

où  $\operatorname{med}$  est la médiane. Pour chaque  $i$ , on calcule la médiane de  $\{|x_i - x_j|; j = 1, \dots, n\}$ , ce qui donne  $n - 1$  valeurs de médiane.  $S_n$  est alors estimée par la médiane de ces médianes. Le facteur  $c$  est un facteur de réajustement, obtenu de manière asymptotique, il est égal à 1.1926.

L'estimateur  $Q_n$  est défini par :

$$Q_n = d \{ |x_i - x_j| \mid i < j \}_{(k)}$$

où  $k = \binom{n}{2} / 4$  soit le premier quartile des  $\binom{n}{2}$  distances, et  $d = 2.2219$ , qui est un facteur de réajustement.

Au final nous avons six estimateurs sur chaque population. Afin de couvrir les techniques d'agrégation présentées précédemment, nous allons combiner les estimateurs sur les populations de diamètres de différentes manières. Nous avons à notre disposition deux jeux de données :

1. un jeu où les données sont non appariées, c'est-à-dire des diamètres sortant d'une analyse AFM et d'une analyse MEB, de la même famille de particules mais pas au même endroit. Nous avons 63 particules AFM et 113 particules MEB, modélisées par les vecteurs  $\mathbf{Y}^{MEB}$  et  $\mathbf{Y}^{AFM}$ .
2. un jeu où les données sont appariées, c'est-à-dire des diamètres sortant d'une analyse AFM et d'une analyse MEB, de la même famille de particules mesurées au même endroit et mises en correspondance en utilisant la technique présentée dans la section 2.2. Ces données sont modélisées par les vecteurs  $\mathbf{Y}^{COL,MEB}$  et  $\mathbf{Y}^{COL,AFM}$



### 2.3. AGRÉGATION D'ESTIMATEURS

---

Pour le jeu de données 1, c'est-à-dire les données non appariées, nous allons utiliser la méthode d'agrégation d'estimateurs en univariée, c'est-à-dire en combinant de manière séparée les estimateurs de position et d'échelle. Pour le jeu de données 2, c'est-à-dire les données appariées, nous allons utiliser la méthode multivariée et estimer le paramètre agrégé de position et d'échelle en une seule fois.

Dans les deux cas, nous avons extrait les valeurs des diamètres en appliquant les techniques présentées en section 2.1.1 pour le MEB et la section 2.1.2 pour l'AFM.

Dans la suite, nous allons présenter des résultats d'estimations avec intervalle de confiance. Pour l'obtention des estimateurs et des intervalles de confiance, nous avons appliqué différentes techniques bootstrap (basic, percentile, student) avec une génération de 10000 échantillons.

#### Estimateurs sur les populations non appariées

Nous allons ici étudier les populations  $\mathbf{Y}^{AFM}$  et  $\mathbf{Y}^{MEB}$  en présentant les résultats des estimateurs présentés précédemment sur chaque population.

**Estimation pour la population AFM** Les tables 2.1 et 2.2 présentent respectivement les résultats du calcul des estimateurs du paramètre de position et d'échelle sur la population de diamètres extraits par AFM  $\mathbf{Y}^{AFM}$  qui comporte 63 diamètres de particules.

**Estimation pour la population MEB** Les tables 2.3 et 2.4 présentent respectivement les résultats du calcul des différents estimateurs du paramètre de position et d'échelle sur la population de diamètres MEB  $\mathbf{Y}^{MEB}$  qui comporte 113 diamètres de particules.

#### Observations sur les estimations sur les populations individuelles

D'après les tables 2.1 et 2.3, les valeurs des estimateurs du paramètre de position pour le diamètre des nanoparticules isolées, c'est-à-dire moyenne, médiane et moyenne tronquée à 10%, entre le MEB et AFM sont significativement différentes. Cette différence s'explique par la technique de mesure sur le profil de la particule en MEB. En effet, comme présenté lors de la description du MEB dans l'annexe 2.A, la poire d'émission générée par l'équipement peut être assez large proportionnellement à l'objet et donc la position de mesure exacte n'est pas évidente à évaluer (car dépendante de nombreux paramètres physiques). La technique AFM, est elle plus robuste pour l'extraction du diamètre comme hauteur de la particule. Nous choisissons cette

Estimateur	Valeur estimée	Intervalle de confiance à 95%			
		Méthode de calcul	Borne inférieure	Borne supérieure	Largeur
Médiane	73.60	Basic	70.91	76.51	5.59
		Percentile	70.70	76.30	5.59
		Student	70.69	76.13	5.44
Moyenne	73.86	Basic	72.05	75.66	3.62
		Percentile	72.05	75.67	3.62
		Student	72.15	75.01	2.86
Moyenne tronquée à 10%	73.74	Basic	71.70	75.74	4.03
		Percentile	71.74	75.78	4.03
		Student	72.87	76.59	3.72

TABLE 2.1 – Résultat du calcul d'estimateurs du paramètre de position sur une population de 63 diamètres de particules isolées mesurées par AFM. Chaque résultat présente la valeur de l'estimateur ainsi que les intervalles de confiance à 95% estimés par bootstrap avec une génération de 10000 échantillons.

Estimateur	Valeur estimée	Intervalle de confiance à 95%			
		Méthode de calcul	Borne inférieure	Borne supérieure	Largeur
Écart-type	8.70	Basic	7.81	9.79	1.98
		Percentile	7.60	9.58	1.98
		Student	7.79	9.81	2.02
$S_n$	9.92	Basic	8.66	11.91	3.25
		Percentile	7.93	11.18	3.25
		Student	8.29	12.49	4.20
$Q_n$	9.55	Basic	8.60	11.46	2.86
		Percentile	7.65	10.51	2.86
		Student	7.50	11.54	4.04

TABLE 2.2 – Résultat du calcul d'estimateurs du paramètre d'échelle sur une population de 63 diamètres de particules isolées mesurées par AFM. Chaque résultat présente la valeur de l'estimateur ainsi que les intervalles de confiance à 95% estimés par bootstrap avec une génération de 10000 échantillons.

### 2.3. AGRÉGATION D'ESTIMATEURS

Estimateur	Valeur estimée	Intervalle de confiance à 95%			
		Méthode de calcul	Borne inférieure	Borne supérieure	Largeur
Médiane	76.68	Basic	75.35	77.68	2.33
		Percentile	75.68	78.01	2.33
		Student	75.09	77.693	2.61
Moyenne	76.84	Basic	75.44	78.20	2.77
		Percentile	75.48	78.25	2.77
		Student	75.10	77.63	2.53
Moyenne tronquée à 10%	76.70	Basic	75.28	78.12	2.84
		Percentile	75.29	78.13	2.84
		Student	75.64	78.50	2.86

TABLE 2.3 – Résultat du calcul d'estimateurs du paramètre de position d'une population de 113 diamètres de particules isolées mesurées par MEB. Chaque résultat présente la valeur de l'estimateur ainsi que les intervalles de confiance à 95% estimés par bootstrap avec une génération de 10000 échantillons.

Estimateur	Valeur estimée	Intervalle de confiance à 95%			
		Méthode de calcul	Borne inférieure	Borne supérieure	Largeur
Écart-type	8.98	Basic	7.84	10.16	2.33
		Percentile	7.80	10.12	2.33
		Student	7.61	10.75	3.13
$S_n$	8.74	Basic	7.72	9.98	2.26
		Percentile	7.50	9.76	2.26
		Student	7.61	9.97	2.36
$Q_n$	8.89	Basic	8.02	10.13	2.11
		Percentile	7.65	9.76	2.11
		Student	7.86	10.31	2.45

TABLE 2.4 – Résultat du calcul d'estimateurs du paramètre d'échelle pour une population de 113 diamètres de particules isolées mesurées par MEB. Chaque résultat présente la valeur de l'estimateur ainsi que les intervalles de confiance à 95% estimés par bootstrap avec une génération de 10000 échantillons.

méthode comme mesure de référence par recommandation du LNE, qui a justifié ce choix par le fait qu'il n'y a pas de biais sur la hauteur de la particule. Dans le cadre de l'AFM, si l'appareil est bien calibré, la plus grande source d'incertitude vient de la rugosité du substrat, dont la valeur est négligeable par rapport au diamètre de la particule.

### Estimation du biais

Pour l'estimation du biais, nous appliquons une agrégation sur les estimateurs de position des populations en combinant la moyenne, la médiane et la moyenne tronquée à 10% pour l'AFM et le MEB. Nous choisissons d'estimer le biais comme la différence des deux estimateurs du paramètre agrégé de position. Nous avons alors deux valeurs d'estimation du paramètre de position de la loi des diamètres en agrégrant les estimateurs des tables 2.1 et 2.3.

Estimateur	Valeur estimée	Intervalle de confiance à 95%			
		Méthode de calcul	Borne inférieure	Borne supérieure	Largeur
AFM Agrégé	74.25	Basic	72.94	75.76	2.82
		Percentile	72.74	75.56	2.82
		Student	72.98	75.79	2.81
MEB Agrégé	76.80	Basic	75.40	77.90	2.50
		Percentile	75.71	78.21	2.50
		Student	75.45	78.01	2.57

TABLE 2.5 – Estimateurs agrégés des estimateurs du paramètre de position pour les populations de diamètres de particules extraits par MEB et par AFM. Les estimateurs qui sont agrégés pour le paramètre de position sont présentés sur les tables 2.1 et 2.3.

On observe que l'intervalle de confiance sur l'estimation du paramètre de position est plus petit que ceux des estimateurs non agrégés. La valeur du paramètre de symétrie est de 74.25 nm pour l'AFM et de 76.8 nm pour le MEB. Dans ce cas précis, nous avons alors un biais estimé de 3.55 nm.

### Agrégation des estimateurs sur populations non appariées

Afin d'estimer le paramètre de position pour la population de particules isolées, nous corrigeons d'abord le biais pour la population MEB comme expliqué précédemment.

### 2.3. AGRÉGATION D'ESTIMATEURS

Nous agrégeons les 6 estimateurs de position pour l'obtention d'une unique valeur. Nous agrégeons les 6 estimateurs du paramètre d'échelle pour l'obtention du paramètre d'échelle de la loi que suit la population des diamètres.

La table 2.6 présente les résultats de l'agrégation. Nous observons que les intervalles de confiance sont plus petits que ceux des estimateurs sur population individuelle présentés sur les tables 2.1, 2.2, 2.3 et 2.4. Nous observons un premier gain sur l'application de l'agrégation d'estimateurs.

Estimateur	Valeur estimée	Intervalle de confiance à 95%			
		Méthode de calcul	Borne inférieure	Borne supérieure	Largeur
Position	73.71	Basic	72.74	74.66	1.92
		Percentile	72.75	74.67	1.92
		Student	72.77	74.68	1.92
Échelle	8.44	Basic	7.76	9.11	1.35
		Percentile	7.77	9.12	1.35
		Student	6.97	8.31	1.34

TABLE 2.6 – Résultat d'agrégation d'estimateurs pour l'estimation du paramètre de position et d'échelle pour une population de diamètre de particules isolées. Les estimateurs qui sont agrégés pour l'estimation du paramètre de position sont présentés sur les tables 2.1 et 2.3 (avant correction du biais). Les estimateurs agrégés pour le paramètre de position sont présentés sur les tables 2.2 et 2.4.

Les particules analysées dans cet exemple ont un paramètre de position  $73.71nm$  et un paramètre d'échelle  $8.44nm$ .

#### Estimation multivariée avec appariement

Dans cet exemple, nous avons un jeu de 6 images MEB et une image AFM prises au même endroit. Nous appliquons l'algorithme de mise en correspondance présenté dans la section 2.2. Nous apparions les particules et avons au final une population de 60 particules avec un diamètre par AFM et un diamètre par MEB, respectivement représentés par les vecteurs  $\mathbf{Y}^{COL,AFM}$  et  $\mathbf{Y}^{COL,MEB}$ . Nous appliquons une correction de biais via la technique définie précédemment puis dans un second temps appliquons la méthode multivariée de l'agrégation d'estimateurs afin d'estimer le paramètre de position et d'échelle de la population de diamètres dans le cas de particules appariées. Les estimateurs que nous agrégeons pour l'estimation paramètre de position et d'échelle sont les mêmes que précédemment.

La table 2.7 présente le résultat de l'agrégation qui est faite en utilisant la méthode d'agrégation multivariée.

Estimateur	Valeur estimée	Intervalle de confiance à 95%			
		Méthode de calcul	Borne inférieure	Borne supérieure	Largeur
Position	73.25	Basic	71.77	74.67	2.91
		Percentile	71.82	74.72	2.91
		Student	71.77	74.69	2.92
Échelle	8.14	Basic	7.27	8.97	1.71
		Percentile	7.30	9.01	1.71
		Student	7.42	9.13	1.71

TABLE 2.7 – Résultat d'agrégation d'estimateur avec correspondance des particules, des estimateurs de la position et d'échelle.

On observe sur la table 2.7 que le diamètre des particules a un paramètre de position  $73.25nm$  et un paramètre d'échelle  $8.14nm$ .

Les valeurs de la largeur d'intervalle de confiance à 95% sont plus grande que dans la table 2.6 car les échantillons sont plus petits dans le cas de données non appariées. En effet, la largeur de l'intervalle de confiance est proportionnelle à la taille des populations. Suite à ce constat, nous avons mis dans l'Annexe 2.B, de ce chapitre, une étude qui met en évidence le fait de tenir compte de l'appariement lors de l'agrégation d'estimateurs.

### Agrégation d'estimateurs avec et sans appariement

Dans cet exemple, nous prenons en compte toutes les particules des tables précédentes et appliquons la méthode d'agrégation d'estimateurs multivariée en tenant compte de l'appariement pour deux des quatre populations dont nous disposons. La table 2.8 contient les résultats de cette agrégation. Nous observons un réel gain au niveau de la taille de l'intervalle de confiance par rapport aux tables 2.7 et 2.6.

### 2.3. AGRÉGATION D'ESTIMATEURS

Estimateur	Valeur estimée	Intervalle de confiance à 95%			
		Méthode de calcul	Borne inférieure	Borne supérieure	Largeur
Position	73.55	Basic	72.76	74.35	1.59
		Percentile	72.80	74.39	1.59
		Student	72.83	74.42	1.59
Échelle	8.33	Basic	7.81	8.85	1.04
		Percentile	7.81	8.85	1.04
		Student	7.84	8.89	1.04

TABLE 2.8 – Résultat d'agrégation des estimateurs présents sur les tables 2.6 et 2.7 pour l'estimation du paramètre de position et du paramètre d'échelle du diamètre des particules.

#### Conclusion

Durant cette section, nous avons montré des exemples d'applications avec le gain obtenu grâce à l'application de la méthode d'agrégation d'estimateurs pour l'estimation du diamètre de particules sphériques.

L'agrégation d'estimateur du paramètre de position d'une part et d'échelle d'autre part, combiné à la fusion d'information provenant des images AFM et MEB a permis d'augmenter la précision en diminuant de façon significative la largeur des intervalles de confiance associés.

# Annexes

## 2.A Principe de fonctionnement d'un MEB

Pour la présentation du principe du fonctionnement du MEB, nous nous sommes inspiré du livre de [Brisset et al. \(2012\)](#). Inventé en 1929, le Microscope Électronique à Balayage (MEB ou SEM pour Scanning Electron Microscope en anglais) est la technique la plus utilisée pour observer des objets sub-micron. Cette technique à l'avantage de ressortir une image ce qui permet d'observer sans forcément analyser. Cette technique est basée sur l'interaction électrons-matière. La Figure 2.A.1 montre le fonctionnement d'un microscope SEM. Le principe est simple, le SEM est un canon à électron qui envoie des électrons sur la surface et, lorsque la surface réagit, renvoie des électrons secondaires que les détecteurs du SEM collectent. Il existe différents types de microscopes électroniques en fonction du type d'électron collecté par les capteurs.

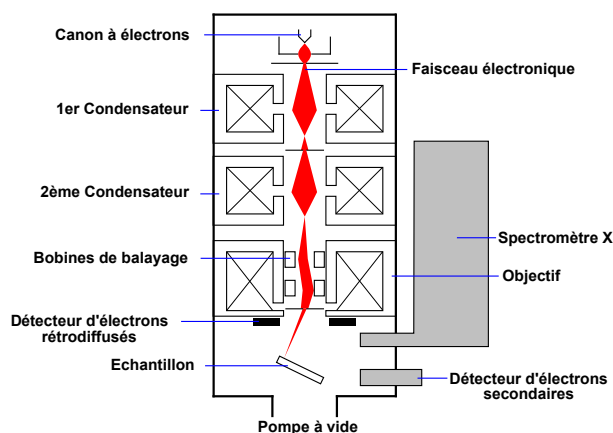


FIGURE 2.A.1 – Schéma d'un MEB

La colonne est composée d'un canon à électrons pour créer le faisceau, de bobines et de condensateurs pour paramétrer le faisceau et d'un objectif



## 2.A. PRINCIPE DE FONCTIONNEMENT D'UN MEB

pour le diriger et créer le balayage. La colonne à électrons est maintenue sous vide par une pompe primaire à un niveau minimum de  $10^{-3} Pa$ . Le faisceau entre dans une chambre où se situe l'échantillon. Il impacte la surface, ce qui entraîne une interaction électron matière, présente sur la Figure 2.A.2. On y observe ce qui est appelé une poire d'interaction. L'échantillon se charge en électron puis envoie des électrons dans différentes directions, voir le schéma sur la Figure 2.A.2. En fonction de l'énergie de l'angle entre la surface et le faisceau d'électron, la quantité d'électrons émis par la surface change. En effet, plus l'angle entre le faisceau et la surface est grand, plus la surface va émettre des électrons secondaires (où backscattered electron), qui sont les électrons collectés par le capteur du MEB. Cette technique permet donc d'avoir une image des changements d'angles et de niveaux. C'est pour cela que la mesure sur image MEB est complexe car la position exacte de mesure n'est pas connue exactement lorsque les objets ont une taille proche de la taille du faisceau.

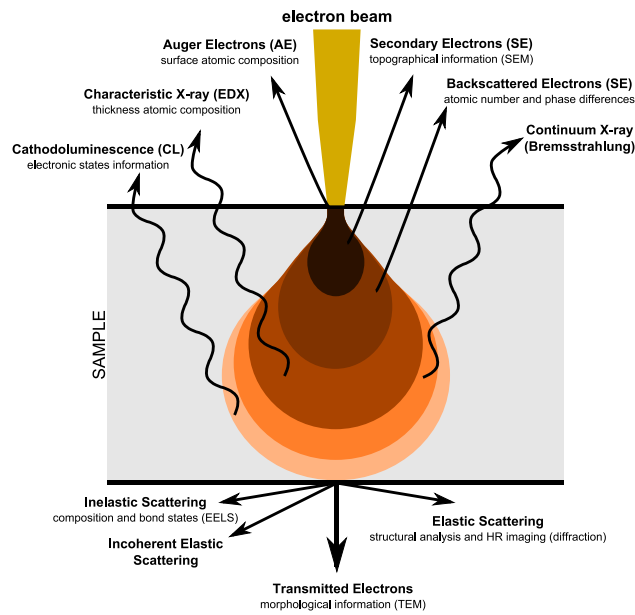


FIGURE 2.A.2 – Schéma de l'interaction électron matière.

Enfin les électrons secondaires créés sont recueillis par le détecteur pour le retransmettre en une image contrastée.

## 2.B Importance de la dépendance entre échantillons

Nous souhaitons ici proposer une étude qui mettra en évidence l'importance d'estimer la matrice de covariance  $\Sigma$  correctement lorsque les échantillons sont dépendants.

Nous allons étudier la variance d'estimateurs de différentes méthodes de calcul d'estimateur de la moyenne de deux vecteurs de taille  $n$  en fonction de leur corrélation. Pour cela, nous simulons un vecteur gaussien bivarié  $X = (X_1, X_2)^\top$  de  $n$  réalisations indépendantes, chaque réalisation suivant une loi normale de moyenne nulle et de matrice de covariance  $\Sigma = \begin{pmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{pmatrix}$ .

Dans notre cas nous allons poser  $\sigma = 1$  et  $n = 100$ . Nous ferons varier  $\rho$  de 0 à 1, afin de voir l'impact de la corrélation sur les variances.

### 2.B.1 Présentation des estimateurs étudiés

Nous allons calculer les moyennes empiriques de  $X_1$  et  $X_2$ , que nous nommerons  $\hat{\theta}_1$  et  $\hat{\theta}_2$ .

Afin d'avoir une base de comparaison pour l'agrégation d'estimateurs, nous allons créer un estimateur qui est la combinaison linéaire de  $\hat{\theta}_1$  et  $\hat{\theta}_2$ , que nous définissons de la façon suivant :

$$\hat{\theta}_M = \alpha \hat{\theta}_1 + (1 - \alpha) \hat{\theta}_2$$

Par hypothèse, la variance de  $\hat{\theta}_M$  est évaluée à :

$$\text{Var}(\hat{\theta}_M) = \alpha^2 \frac{\sigma^2}{n} + (1 - \alpha)^2 \frac{\sigma^2}{n} + 2\alpha(1 - \alpha) \frac{\rho\sigma^2}{n} \quad (2.1)$$

et on peut vérifier que la valeur de  $\alpha$  minimisant cette variance est  $\alpha = \frac{1}{2}$

L'estimateur  $\hat{\theta}_M = \frac{1}{2} \hat{\theta}_1 + \frac{1}{2} \hat{\theta}_2$  est donc la combinaison linéaire sans biais de  $\hat{\theta}_1$  et  $\hat{\theta}_2$  de variance minimale.

L'objectif de cette étude étant d'étudier les différents estimateurs obtenus par agrégation de  $\hat{\theta}_1$  et  $\hat{\theta}_2$ , nous calculons également les estimateurs agrégés dont les poids sont obtenus via la méthode précédemment décrite. Au final, pour cette étude, nous allons calculer et comparer 5 estimateurs de la moyenne différents :

1.  $\hat{\theta}_1$  estimateur de la moyenne sur la population 1.
2.  $\hat{\theta}_2$  estimateur de la moyenne sur la population 2.
3.  $\hat{\theta}_M = \frac{1}{2} \hat{\theta}_1 + \frac{1}{2} \hat{\theta}_2$ .  $\hat{\theta}_M$ .

## 2.B. IMPORTANCE DE LA DÉPENDANCE ENTRE ÉCHANTILLONS

---

4.  $\hat{\theta}_D = \lambda_1 \hat{\theta}_1 + \lambda_2 \hat{\theta}_2$ .  $\hat{\theta}_D$  est l'estimateur obtenu par agrégation des estimateurs  $\hat{\theta}_1$  et  $\hat{\theta}_2$  dont les poids  $\lambda_1$  et  $\lambda_2$  sont estimés en utilisant la méthode présentée précédemment. L'estimateur de la matrice de covariance est obtenu par technique bootstrap en supposant les échantillons appariés.
5.  $\hat{\theta}_A = \lambda_1 \hat{\theta}_1 + \lambda_2 \hat{\theta}_2$ .  $\hat{\theta}_A$  est l'estimateur obtenu par agrégation des estimateurs  $\hat{\theta}_1$  et  $\hat{\theta}_2$  dont les poids  $\lambda_1$  et  $\lambda_2$  sont estimés en utilisant la méthode présentée précédemment. L'estimateur de la matrice de covariance est obtenu par technique bootstrap en supposant (à tort) les échantillons indépendants.

### 2.B.2 Présentation des variances d'estimateurs étudiées

Nous allons étudier la variance de 5 estimateurs présentés précédemment. Il existe plusieurs façons d'estimer la variance d'un estimateur, dans notre cas, nous calculerons la variance par bootstrap et par Monte Carlo.

Notre protocole de test sera donc le suivant, on effectue une étude Monte Carlo où nous simulons  $m = 1000$  fois des vecteurs gaussiens bivariés, pour un  $\rho$  fixé. À chaque itération, nous allons estimer les estimateurs précédemment définis, ainsi que les variances suivantes :

1.  $\hat{\sigma}_{\hat{\theta}_1}^2$  : estimateur de la variance de l'estimateur  $\hat{\theta}_1$  via la matrice de covariance, obtenue par bootstrap. Cet estimateur est indifférent à la dépendance des échantillons.
2.  $\hat{\sigma}_{\hat{\theta}_2}^2$  : estimateur de la variance de l'estimateur  $\hat{\theta}_2$  via la matrice de covariance, obtenue par bootstrap. Cet estimateur est indifférent à la dépendance des échantillons.
3.  $\hat{\sigma}_{\hat{\theta}_{M,D}}^2$  : estimateur de la variance de l'estimateur  $\hat{\theta}_M$  obtenue par bootstrap sur les indices, c'est-à-dire en supposant les échantillons appariés.
4.  $\hat{\sigma}_{\hat{\theta}_D}^2$  : estimateur de la variance de l'estimateur  $\hat{\theta}_D$ , calculé par bootstrap sur les indices, c'est-à-dire en supposant les échantillons appariés.
5.  $\hat{\sigma}_{\hat{\theta}_{M,A}}^2$  : estimateur de la variance de l'estimateur  $\hat{\theta}_M$  obtenue par bootstrap sur les éléments, c'est-à-dire en supposant (à tort) les échantillons indépendants.
6.  $\hat{\sigma}_{\hat{\theta}_A}^2$  : estimateur de la variance de l'estimateur  $\hat{\theta}_A$ , calculé par bootstrap sur les valeurs, c'est-à-dire en supposant (à tort) les échantillons indépendants.

Nous estimons  $m$  fois les variances précédentes, puis estimons la moyenne empirique des  $m$  valeurs de variance de chacun des estimateurs, pour un  $\rho$

fixé. Nous obtenons la moyenne Monte Carlo de la variance des estimateurs que nous noterons par  $M_{MC}(\hat{\sigma}_{\hat{\theta}}^2)$

D'autre part, pour chacun des estimateurs  $\hat{\theta}$  précédemment définis, nous calculons leurs variances Monte Carlo. C'est-à-dire que nous estimons  $m$  fois chaque estimateur, puis calculons la variance empirique des  $m$  valeurs de chacun des estimateurs, pour un  $\rho$  fixé. Nous noterons cette variance Monte Carlo des estimateurs par  $V_{MC}(\hat{\theta})$ .

En théorie, pour un  $\hat{\theta}$ ,  $M_{MC}(\hat{\sigma}_{\hat{\theta}}^2)$  et  $V_{MC}(\hat{\theta})$  convergent vers la même valeur.

La valeur de variance de l'estimateur  $\hat{\theta}_M$  peut être calculée en partant de l'équation 2.1, avec  $\alpha = \frac{1}{2}$  :

$$\text{Var}(\hat{\theta}_M) = \frac{1}{4} \frac{\sigma^2}{n} + \frac{1}{4} \frac{\sigma^2}{n} + 2 \frac{1}{4} \frac{\rho}{n} = \sigma^2 \frac{(1 + \rho)}{2n}$$

Théoriquement, les valeurs de variance de  $\hat{\theta}_M$ , obtenue par moyenne Monte Carlo et par variance Monte Carlo devraient converger vers  $\text{Var}(\hat{\theta}_M)$ .

Les poids optimum étant  $\frac{1}{2}$ , on suppose que les poids des estimateurs  $\hat{\theta}_D$  et  $\hat{\theta}_A$ , calculés par la méthode d'agrégation, seront proches de  $\frac{1}{2}$  et donc la variance des estimateurs  $\hat{\theta}_D$  et  $\hat{\theta}_A$  sera proche de la valeur de  $\text{Var}(\hat{\theta}_M)$ .

### 2.B.3 Résultats

La Figure 2.B.1 représente le tracé des résultats du calcul des variances d'estimateurs précédemment définies. Les valeurs de variances ont été normalisées en multipliant la valeur par le nombre d'éléments contenus dans un échantillon, soit  $n$ . Nous avons également tracé les variances théoriques des estimateurs  $\text{Var}(\hat{\theta}_i)$ ,  $i = 1, 2$  et  $\text{Var}(\hat{\theta}_M)$ .

#### Analyse de la figure 2.B.1

- $M_{MC}(\hat{\sigma}_{\hat{\theta}_1}^2)$  suit la même tendance que  $V_{MC}(\hat{\theta}_1)$ , qui est égale à la valeur théorique de cet estimateur. Ceci est donc cohérent, nous pouvons appliquer la même conclusion pour les variances de  $\hat{\theta}_2$ .
- $M_{MC}(\hat{\sigma}_{\hat{\theta}_{MD}}^2)$  ainsi que  $V_{MC}(\hat{\theta}_M)$  suivent la valeur théorique. En revanche,  $M_{MC}(\hat{\sigma}_{\hat{\theta}_{MA}}^2)$  ne suit pas la valeur théorique et est proche de  $\frac{1}{2}$ . En effet, le fait de calculer la variance en supposant les échantillons indépendants revient à supposer  $\rho = 0$ , on a donc une mauvaise estimation de la variance, et en particulier une sous estimation.
- De façon consistante,  $M_{MC}(\hat{\sigma}_{\hat{\theta}_D}^2)$  et  $V_{MC}(\hat{\theta}_D)$  suivent la même tendance, comme pour la variance théorique que  $\hat{\theta}_M$ . En effet, comme on

## 2.B. IMPORTANCE DE LA DÉPENDANCE ENTRE ÉCHANTILLONS

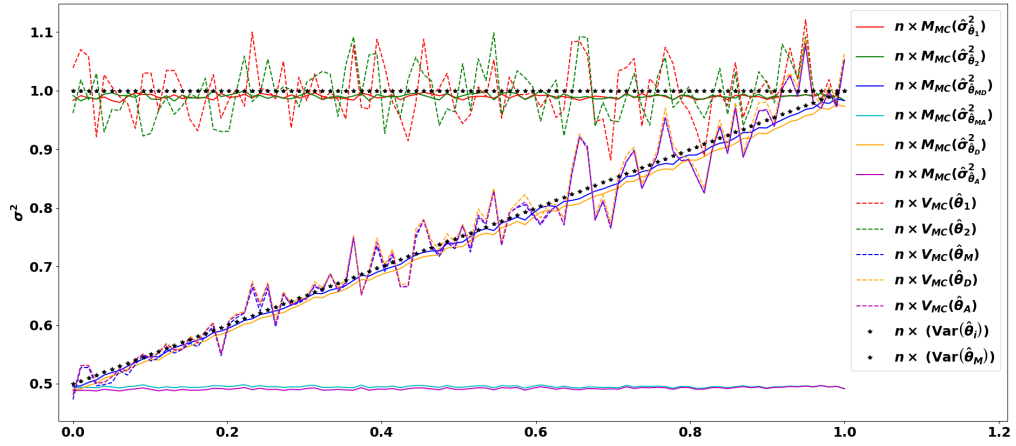


FIGURE 2.B.1 – Tracé des différentes variances estimées par moyenne Monte Carlo ( $M_{MC}(\hat{\sigma}_{\hat{\theta}}^2)$ ) en trait plein, variance Monte Carlo ( $V_{MC}(\hat{\theta})$ ) en pointillés et variance théorique en noir, des estimateurs de moyenne. Les valeurs de variance sont tracées en fonction de la corrélation  $\rho$  pour les différents estimateurs. Les variances sont normalisées en les multipliant par le nombre d'éléments contenus dans les échantillons, soit  $n$ .

pouvait le supposer, la méthode d'agrégation d'estimateurs permet de calculer les poids optimaux.

- $M_{MC}(\hat{\sigma}_{\hat{\theta}_A}^2)$  ne suit pas  $V_{MC}(\hat{\theta}_A)$ . En effet,  $V_{MC}(\hat{\theta}_A)$  suit la variance de  $\hat{\theta}_M$  mais  $M_{MC}(\hat{\sigma}_{\hat{\theta}_A}^2)$  est proche de  $\frac{1}{2}$ . Cette divergence montre une incohérence car les valeurs estimées sont les mêmes. Le fait que  $V_{MC}(\hat{\theta}_A)$  soit proche de la valeur théorique de  $\hat{\theta}_M$  n'est pas surprenant, car la méthode d'obtention des poids de l'estimateur  $\hat{\theta}_A$  doit retourner des poids très proches de ceux de  $\hat{\theta}_M$ . Mais lors du calcul de  $M_{MC}(\hat{\sigma}_{\hat{\theta}_A}^2)$ , on suppose à tort les échantillons indépendants, ce qui revient à supposer que  $\rho = 0$ , ce qui explique pourquoi il est proche de  $\frac{1}{2}$ . On en déduit que le fait de supposer à tort les échantillons indépendants implique une erreur sur le résultat de l'estimateur.

### 2.B.4 Conclusion

Les estimateurs agrégés sans prendre en compte la correspondance entre les particules, disposent d'un intervalle de confiance plus petit, à cause de la plus faible variance, mais cet intervalle de confiance est sous-estimé. En effet, on observe que dans le cas où la correspondance n'est pas prise en compte lors du calcul de la matrice de covariance, les variances d'estimateurs

ne sont pas cohérentes avec la valeur théorique. Cette étude a mis en avant cette incohérence et permet à un utilisateur non averti de ne pas se laisser prendre au piège de ne pas tenir compte de cette correspondance pour avoir un intervalle de confiance plus petit.

2.B. IMPORTANCE DE LA DÉPENDANCE ENTRE ÉCHANTILLONS

# Bibliographie

- Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). KAZE features. In *Eur. Conf. on Computer Vision (ECCV)*.
- Bates, J. M. and Granger, C. W. (1969). The combination of forecasts. *Journal of the Operational Research Society*, 20(4) :451–468.
- Beucher, S. and Meyer, F. (1992). The morphological approach to segmentation : the watershed transformation. *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-*, 34 :433–433.
- Borgefors, G. (1986). Distance transformations in digital images. *Comput. Vision Graph. Image Process.*, 34(3) :344–371.
- Brisset, F. et al. (2012). *Microscopie électronique à balayage et microanalyses*. EDP sciences.
- Coeurjolly, D., Montanvert, A., and Chassery, J.-M. (2007). Géométrie discrète et images numériques. Ouvrage collectif, *Traité IC2*, Hermès, pp-416.
- Coeurjolly, J.-f. and Drouillhet, R. (2007). Notes de cours : Bootstrap et méthodes de rééchantillonnage.
- Croux, C. and Rousseeuw, P. J. (1992). Time-efficient algorithms for two highly robust estimators of scale. In *Computational statistics*, pages 411–428. Springer.
- Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and Their Applications*. Cambridge University Press, Cambridge. ISBN 0-521-57391-2.
- Delvallée, A., Feltin, N., Ducourtieux, S., Trabelsi, M., and Hochepped, J. F. (2015). Direct comparison of afm and sem measurements on the same set of nanoparticles. *Measurement Science and Technology*, 26(8) :085601.



## BIBLIOGRAPHIE

---

- Forsyth, D. and Ponce, J. (2011). *Computer vision : a modern approach*. Upper Saddle River, NJ; London : Prentice Hall.
- Haralick, R. M. and Shapiro, L. G. (1993). *Computer and Robot Vision*, volume 2. Addison Wesley.
- ISO 25178–2 (2012). Spécification géométrique des produits (gps) – État de surface : Surfaique – partie 2 : Termes, définitions et paramètres d'états de surface. Standard, International Organization for Standardization, Geneva, CH.
- Laplace, P. d. (1812). *Théorie analytique des probabilités*. Ve. Courcier.
- Lavancier, F. and Rochet, P. (2016). A general procedure to combine estimators. *Computational Statistics & Data Analysis*, 94(C) :175–192.
- Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340.
- Najman, L. and Talbot, H. (2008). *Morphologie Mathématique 1 : approches déterministes*. Hermès / Lavoisier. (260 pp.).
- Otsu, N. (1979). A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1) :62–66.
- Shum, H.-Y. and Szeliski, R. (1997). Panoramic image mosaics. Technical report, Technical Report MSR-TR-97-23, Microsoft Research.
- Soille, P. (2003). *Morphological Image Analysis : Principles and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 edition.

# Chapitre 3

## Optimisation de paramètres de procédé de fabrication

### Introduction

Les copolymères à blocs autrement appelés "block copolymers" (cf Article [Park et al. \(2003\)](#)) sont des substances chimiques qui sont, entre autre, utilisées dans le milieu du semi-conducteur dans le cadre de la fabrication de microprocesseurs. Dans ce chapitre nous présenterons un type de structure générée par des copolymères à blocs : le lamellaire. Le lamellaire, ce sont des lignes qui se sont auto-assemblées en surface libre. Afin d'analyser les structures produites, nous avons à notre disposition des images prises par microscopie électronique à balayage (MEB). La première partie de ce chapitre est dédiée à l'analyse d'images de lamellaire. Nous présenterons tout d'abord le lamellaire ainsi que les objectifs de l'analyse d'images de lamellaire puis l'algorithme développé pour répondre à ces objectifs.

Le développement d'une nouvelle formulation pour faire des structures avec une certaine dimension est faite de la façon suivante : l'expert va réaliser un plan d'expérience. Il va faire varier les paramètres à optimiser, comme par exemple la température, l'épaisseur de la sous-couche, la concentration du polymère et d'autres. Le plan d'expérience réalisé, il doit ensuite analyser les résultats et à partir de cela, redéfinir une zone restreinte dans l'espace des paramètres où le procédé respecte les propriétés désirées. Il refait ensuite des plans d'expériences de manière récursive jusqu'à avoir les valeurs souhaitées ou une fenêtre suffisamment petite.

---

Nous nous sommes intéressés à cette problématique en cherchant comment simplifier l'optimisation de paramètres de procédé de fabrication. Notre approche est la suivante, nous devons être en mesure d'estimer le résultat pour un point qui n'est pas dans le plan d'expérience, en appliquant une méthode de régression. Puis, partant de cette estimation, rechercher dans un second temps la zone la plus petite possible dans l'espace des paramètres qui permet de répondre aux propriétés souhaitées. La seconde partie de ce chapitre est dédiée à la régression pour l'estimation de résultat avec des valeurs de paramètres qui ne sont pas dans le plan d'expérience initiale.

## Sommaire

---

<b>Introduction</b> . . . . .	<b>121</b>
<b>3.1 Introduction</b> . . . . .	<b>123</b>
3.1.1 Présentation du lamellaire . . . . .	123
3.1.2 Objectifs de l'analyse d'image de lamellaire . . . . .	123
3.1.3 État de l'art . . . . .	127
<b>3.2 Analyse d'image lamellaire</b> . . . . .	<b>127</b>
3.2.1 Calcul de la période . . . . .	128
3.2.2 Calcul de l'orientation . . . . .	129
3.2.3 Pré-traitement . . . . .	130
3.2.4 Binarisation . . . . .	131
3.2.5 Calcul du squelette . . . . .	132
3.2.6 Extraction des défauts . . . . .	132
3.2.7 Calcul des dimensions des lignes . . . . .	133
3.2.8 Conclusion . . . . .	134
<b>3.3 Méthodes de régression</b> . . . . .	<b>135</b>
3.3.1 Description des données . . . . .	135
3.3.2 Régressions linéaires pénalisées . . . . .	136
3.3.3 Interpolation par pondération inverse à la distance . . . . .	137
3.3.4 Krigeage . . . . .	137
3.3.5 Réseau de neurones . . . . .	139
3.3.6 Forêt aléatoire : Random Forest . . . . .	140
<b>3.4 Résultats</b> . . . . .	<b>142</b>
3.4.1 Critères de performance . . . . .	142
3.4.2 Résultats régression . . . . .	143
<b>3.5 Conclusion et discussion</b> . . . . .	<b>150</b>

---

## 3.1 Introduction

### 3.1.1 Présentation du lamellaire

Les copolymères ont la propriété de s'auto-assembler, en formant des structures organisées lorsqu'ils sont chauffés. Nous nous intéressons ici à la structure "lamellaire". Sous certaines conditions chimiques et thermodynamiques, les copolymères s'organisent en une succession de couches, comme illustré dans la Figure 3.1.

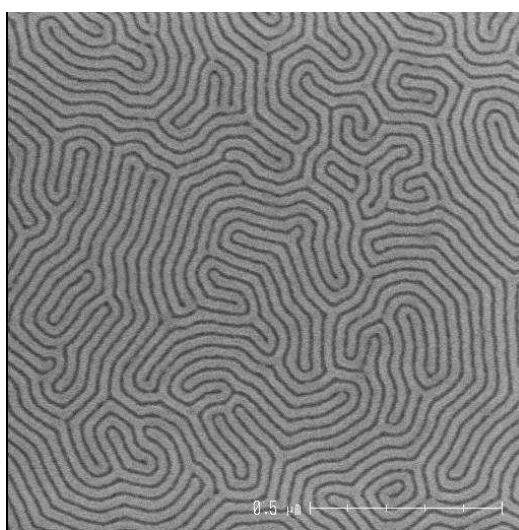


FIGURE 3.1 – Exemple d'image de lamellaire

Afin d'explorer le domaine du lamellaire, nous invitons le lecteur à lire les articles de [Zhang et al. \(2010\)](#), [Williamson et al. \(2015\)](#), [Harrison et al. \(2002\)](#), [Ryu et al. \(2013\)](#) et [Harrison et al. \(2000\)](#).

### 3.1.2 Objectifs de l'analyse d'image de lamellaire

L'objectif est d'extraire d'une image comme celle présentée dans la Figure 3.1 un certain nombre d'informations. Nous devons séparer les lignes claires des lignes sombres puis en extraire les défauts et des propriétés dimensionnelles des lignes. Nous allons dans un premier temps présenter les défauts et leur nomenclature puis dans un second temps les propriétés dimensionnelles à extraire.

#### Nomenclature des défauts

Nous devons extraire les défauts des images, ce sont les points caractéristiques des lignes. La nomenclature des défauts est décrite dans la publication de [Zhang et al. \(2010\)](#).

Nous avons trois types de défauts, représentés sur la Figure 3.2 :

1. le **junction**, que l'on peut observer sur l'image 3.2a, autrement nommé le défaut en **Y**, lorsqu'une ligne se divise et forme alors un **Y** localement.
2. le **end of line**, que l'on peut observer sur l'image 3.2b, autrement nommé le défaut de fin de ligne.
3. le **dot**, que l'on peut observer sur l'image 3.2c, autrement nommé le défaut cylindre, est une ligne tellement courte qu'elle forme un cercle.

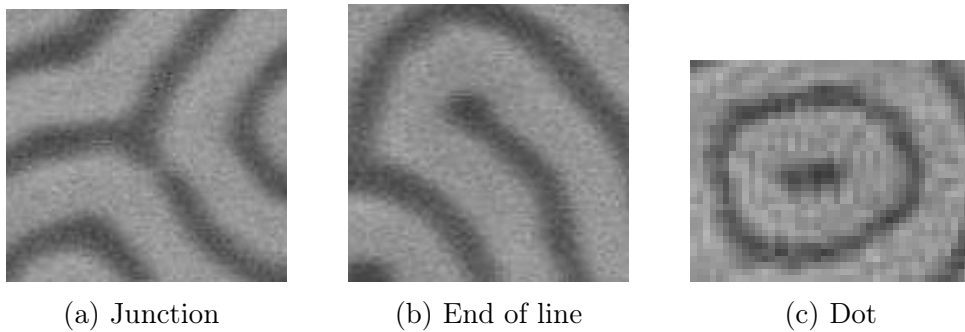


FIGURE 3.2 – Exemples des différents types de défauts

Le défaut de type cylindre, n'est en réalité que deux défauts de fin de ligne très proches. Nous ne traiterons donc que les deux premiers types de défauts.

#### Dimensionnel des lignes

**CD** : Le CD ou critical dimension est la largeur de la ligne, c'est le  $w_i$  présenté sur la Figure 3.3.

**Rugosité** : Deux différentes sortes de rugosités sont observées : la rugosité de ligne et la rugosité de bord de ligne.

La rugosité de bord de ligne, nommé LER pour Line Edge Roughness est une grandeur physique qui représente les irrégularités d'un front de ligne comme on peut l'observer sur la Figure 3.4. Ces variations physiques vont

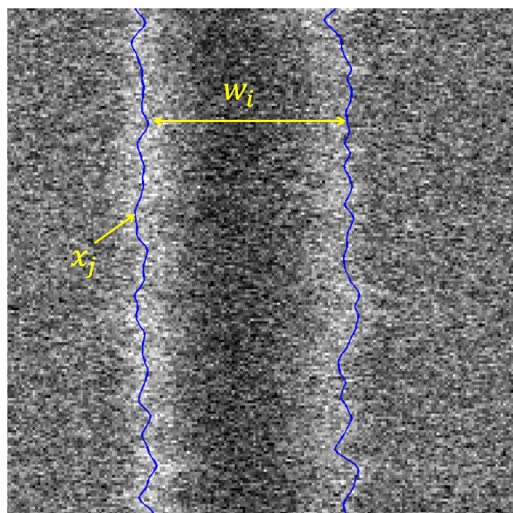


FIGURE 3.3 – Schéma montrant les composantes pour calculer la rugosité sur une ligne



FIGURE 3.4 – Vue en coupe d'une ligne de résine

### 3.1. INTRODUCTION

---

engendrer des perturbations électriques au niveau du dispositif électronique final.

Le LER est défini mathématiquement par trois fois l'écart type de la position d'un bord de ligne :

$$LER = 3\sigma = 3\sqrt{\frac{1}{N-1} \sum_{j=0}^{N-1} (x_j - \bar{x})^2}$$

où  $\bar{x}$  représente la moyenne des positions  $x_j$ ,  $j = 0, \dots, N - 1$ , présenté sur la Figure 3.3

Le LER à gauche et à droite de la ligne est défini de la même manière, seule la position des points change.

Le LWR ou Line Width Roughness représente la rugosité de la ligne. Elle représente l'irrégularité du CD le long de la ligne. Elle est décrite mathématiquement comme trois fois l'écart type du CD le long de la ligne :

$$LWR = 3\sigma = 3\sqrt{\frac{1}{N-1} \sum_{j=0}^{N-1} (w_j - \bar{w})^2}$$

où  $\bar{w}$  représente la moyenne des dimensions critiques  $w_j$ ,  $j = 0, \dots, N - 1$ , présenté sur la Figure 3.3

La distribution du CD le long d'une ligne est supposée être approchée par une loi normale, ce qui explique le choix du  $3\sigma$  afin de pouvoir considérer 99.7% des valeurs, et donc d'estimer correctement la rugosité.

#### Liste des paramètres à extraire des images

La table 3.1 présente les différents paramètres à extraire des images de microscopie électronique à balayage de lamellaire.

Nom	Explications
Période	Période moyenne du signal formé par les lignes
CD	Largeur de la ligne. Deux CD devront être calculés, un pour les lignes claires et un pour les lignes foncées.
LWR	Rugosité de la largeur de ligne. Deux LWR devront être calculés, un pour les lignes claires et un pour les lignes foncées.
LER	Rugosité de bord de ligne. Un seul LER sera retourné car il est calculé avec la frontière entre le clair et le sombre.
Défaut junction	Défaut de type Y (image 3.2a), à fournir pour les lignes claires et foncées. Le nombre de défauts ainsi que la densité.
Défaut end of line	Défaut de type fin de ligne (image 3.2b), à fournir pour les lignes claires et foncées. Le nombre de défauts ainsi que la densité

TABLE 3.1 – Tableau récapitulatif des paramètres

### 3.1.3 État de l’art

Actuellement, il n’existe pas d’offre commerciale permettant l’analyse d’image MEB de lamellaire afin d’en extraire les propriétés dimensionnelles et les défauts. Il existe des logiciels qui proposent de l’analyse de lignes mais pour des lignes droite, qui sont les motifs classiques de la fabrication de microprocesseurs en utilisant les étapes en place actuellement. Cependant, les laboratoires et entreprises travaillant dans le domaine ont implémenté des solutions individuelles (au titre de l’article de [Murphy et al. \(2015\)](#)), souvent non-automatiques, non optimisées, peu robustes et demandant plusieurs paramètres à l’utilisateur final.

## 3.2 Analyse d’image lamellaire

Le schéma de la Figure 3.1 décrit le pipeline de l’algorithme d’analyse d’images MEB de lamellaire. Dans cette section, nous détaillerons les différentes étapes présentes sur ce schéma en expliquant pourquoi nous avons choisi ces approches ainsi que leurs avantages et limitations.

Le schéma 3.1 présente les étapes de l’algorithme dédié à l’analyse d’image de lamellaire. L’algorithme doit retourner les informations dimensionnelles et de défektivité pour les deux types de lignes, claires ou foncées. C’est pour-



### 3.2. ANALYSE D'IMAGE LAMELLAIRE

---

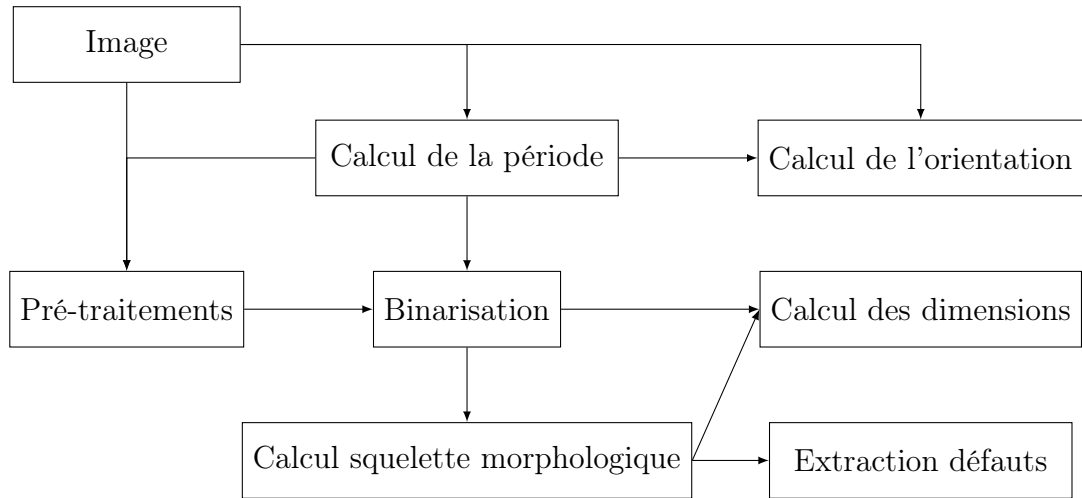


FIGURE 3.1 – Schéma du fonctionnement de l'algorithme d'analyse d'image de lamellaire.

quoi nous avons besoin de les séparer en classes distinctes afin de les analyser séparément. Les images de lamellaire présentent des lignes périodiques dont l'orientation est aléatoire. Nous allons utiliser cette propriété des images comme un avantage pour la calibration des différentes étapes. Nous montrerons dans un premier temps comment extraire la période de ces images. Nous expliquerons ensuite comment nous améliorons le signal via une diminution du bruit et une correction d'illumination.

Nous présenterons alors comment nous séparons les lignes en deux classes distinctes pour analyse dimensionnelle et de défektivité. Pour l'extraction de défaut ainsi que pour la dimension des lignes, nous allons utiliser le squelette morphologique. Nous expliquerons alors comment nous obtenons le squelette et comment nous extrayons les défauts de ce dernier. Nous finirons par l'analyse dimensionnelle, qui utilise les résultats des étapes précédentes pour une caractérisation complète des lignes.

#### 3.2.1 Calcul de la période

Le calcul de la période consiste en l'extraction de la fréquence la plus courante sur l'image. Cette grandeur est représentative de l'épaisseur d'une ligne foncée plus celle d'une ligne claire. Afin d'extraire cette grandeur, on applique une transformée de Fourier 2D à l'image d'origine en niveaux de gris.

On peut observer le résultat d'une transformée de Fourier sur une image dans la Figure 3.2b

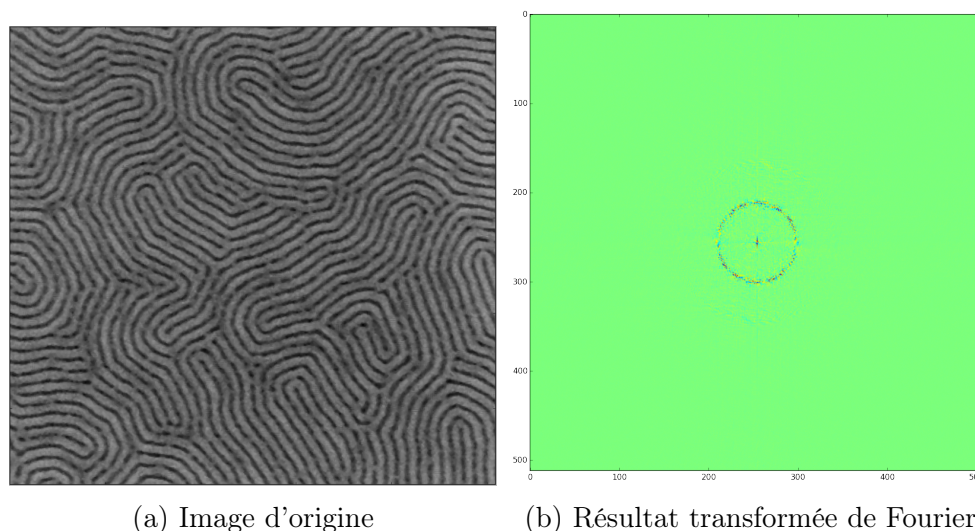


FIGURE 3.2 – Exemple de résultat de l'application d'une transformée de Fourier à une image de lamellaire

Afin de s'affranchir des basses fréquences, le centre de la matrice résultant de la transformée de Fourier est ignoré.

À partir de cette image, nous cherchons la position du point maximal qui est la valeur pour laquelle la répétition est maximale. Nous obtenons ainsi la fréquence la plus présente. Afin d'obtenir la période, il faut appliquer l'inverse de la position sur la largeur de l'image.

Nous avons choisi d'utiliser une transformée de Fourier car c'est une méthode simple et efficace pour extraire une période sur un signal. Nous aurions pu aller plus loin en utilisant une fonction d'autocorrélation ou bien en cherchant à interpoler un cercle, ou plus exactement une ellipse, sur l'image résultant de la transformée de Fourier. Mais dans notre application, nous souhaitons extraire la période moyenne de l'image, ce qui correspond à la fréquence dominante dans l'image.

### 3.2.2 Calcul de l'orientation

Le calcul de l'orientation locale, est fait en prenant pour un point donné, une région de deux fois la période autour de ce point. Nous appliquons ensuite une transformée de Fourier en 2D afin de récupérer l'orientation de la

fréquence locale dominante. Nous obtenons ainsi l'orientation pour un point donné. Nous ne calculons pas l'orientation pour tous les points dans l'image, nous discrétisons le calcul de l'orientation en fonction de la période puis appliquons une interpolation au plus proche voisin afin d'avoir une estimation de l'orientation sur toute l'image.

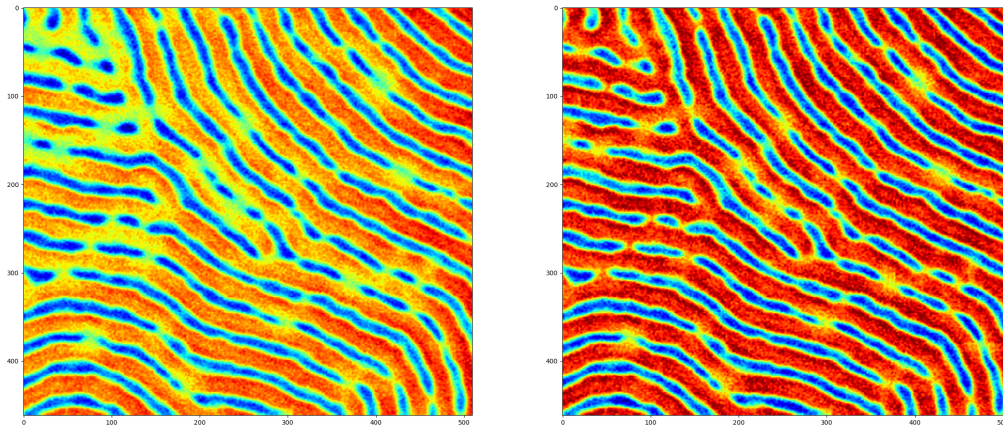
### 3.2.3 Pré-traitement

L'image obtenue par microscopie électronique à balayage contient du bruit et une illumination non homogène. Comme toutes les techniques de prise d'image, les capteurs sont sujets au bruit, surtout lorsque les objets à analyser sont de petites dimensions, à la limite de résolution des appareils. Une autre source de défauts est l'illumination non uniforme sur les images. Nous utilisons le terme illumination même s'il ne s'agit pas de lumière mais d'électrons cependant les défauts engendrés sont semblables. En effet, le fond de l'image n'est pas uniforme et dans certaines zones, les lignes de même catégorie (claire ou foncée) n'ont pas le même niveau de gris, comme l'illustre l'image [3.3a](#) de la Figure [3.3](#). Ce changement d'intensité pose problème lorsque nous devons segmenter l'image. C'est pourquoi nous devons dans un premier temps améliorer le signal via un pré-traitement.

Pour la correction du bruit, nous appliquons un filtre gaussien avec un masque de taille  $3 \times 3$  et  $\sigma = 1$ , qui permet de supprimer en partie le bruit.

La correction d'illumination est un peu plus complexe et il existe plusieurs solutions à ce problème. Afin de le résoudre, nous utilisons une approche morphologique. Nous appliquons un top-hat qui est la différence entre l'image brute et l'image après application d'une ouverture morphologique. Une ouverture est une érosion suivie d'une dilatation. Rappelons qu'une dilatation est l'application d'un filtre où l'élément central est remplacé par l'élément maximum de l'image à l'emplacement du masque tandis que l'érosion est le remplacement de l'élément central par le minimum. Le masque nécessaire pour l'amélioration du signal est de dimension proportionnel à la période. Un exemple de résultat est présent sur la Figure [3.3](#). On observe sur cette image que les deux types de ligne ont des niveaux d'intensité homogènes sur toute l'image.

Une autre voie aurait été de s'orienter vers des techniques plus complexes mais plus génériques à l'exemple de la technique présentée dans le papier de [Geffray et al. \(2016\)](#). En effet, le top-hat ne permet pas de corriger l'illumination si les objets sont tous de taille similaire, ce qui est le cas pour notre application.



(a) Image d'origine représentée avec une colormap jet

(b) Image après application d'un top-hat représentée avec une colormap jet

FIGURE 3.3 – Exemple de résultat de correction d'illumination en utilisant une approche morphologique à une image de lamellaire

### 3.2.4 Binarisation

Dans cette étape nous voulons « couper » l'image en deux parties. Une pour les lignes claires et une pour les lignes foncées. L'objectif est donc de faire de la segmentation des niveaux de gris, pour séparer les deux classes de lignes, représentées en deux populations de niveaux de gris. Pour cela, il existe beaucoup d'approches possibles. Nous pourrions appliquer un seuillage par la méthode d'Otsu, une segmentation en niveaux de gris en appliquant un algorithme de K-moyennes avec  $k = 2$  sur la population des niveaux de gris des pixels de l'image mais nous allons utiliser la périodicité de nos images comme un avantage et appliquer un seuillage adaptatif.

Le seuillage adaptatif est un seuillage par masque. La méthode, ainsi que celles citées précédemment sont présentées dans le livre de [Gonzalez et al. \(2002\)](#). Nous allons appliquer un seuillage adaptatif avec une taille de masque proportionnelle à la taille de la période. Le seuil choisi pour chaque position de masque est la moyenne sur ce masque. Ainsi le seuillage adaptatif permet de binariser l'image en séparant les lignes claires des lignes foncées. L'image 3.4b présente un résultat de binarisation sur une image de lamellaire en utilisant la méthode de seuillage adaptatif.

### 3.2.5 Calcul du squelette

Nous avons besoin d'extraire le squelette des lignes pour l'analyse des défauts ainsi que pour une partie de l'analyse dimensionnelle des lignes. La squelettisation peut être faite de différentes façons. Dans notre cas, nous nous sommes intéressés au squelette morphologique. Le squelette morphologique est obtenu en appliquant un algorithme de Zhang-Suen, présenté dans l'article de [Zhang and Suen \(1984\)](#). Un exemple de résultat est présenté sur l'image [3.4c](#) de la Figure [3.4](#). L'algorithme de squelettisation de Zhang-Suen est un algorithme itératif qui analyse chaque pixel de l'image et le conserve uniquement s'il respecte certaines conditions sur le voisinage. L'algorithme est appliqué de manière récursive jusqu'à qu'il ne reste qu'un squelette d'un pixel de large.

### 3.2.6 Extraction des défauts

Pour l'extraction des défauts, nous nous sommes inspirés de l'analyse d'empreintes digitales, et plus précisément, de l'analyse numérique des minuties. Dans les articles de [Zaeri \(2011\)](#), de [Mahmood \(2011\)](#) et de [Farina et al. \(1999\)](#), on nous explique comment extraire les différentes minuties d'un squelette ou bien d'une image binaire.

Nous avons tiré parti de cette phase d'apprentissage et avons développé l'Algorithme [1](#). Cet algorithme étudie le voisinage direct de chaque point du squelette afin de savoir s'il s'agit d'un point cohérent ou d'un défaut, et dans ce dernier cas la nature de ce défaut.

---

**Algorithm 1:** Extraction des défauts sur le squelette

---

**Input:**  $I$  a binary image.  
**Output:**  $EOF$  and  $Y$  a list of point respectively contains the end of line point and the junction point.

```

 $EOF \leftarrow []$ 
 $Y \leftarrow []$ 
 $[height, width] \leftarrow size(I)$ 
for  $i \leftarrow 1$  to  $height - 1$  do
    for  $j \leftarrow 1$  to  $width - 1$  do
        if  $I(i, j)$  then
             $p1 \leftarrow I(i + 1, j)$ 
             $p2 \leftarrow I(i + 1, j - 1)$ 
             $p3 \leftarrow I(i, j - 1)$ 
             $p4 \leftarrow I(i - 1, j - 1)$ 
             $p5 \leftarrow I(i - 1, j)$ 
             $p6 \leftarrow I(i - 1, j + 1)$ 
             $p7 \leftarrow I(i, j + 1)$ 
             $p8 \leftarrow I(i + 1, j + 1)$ 
             $CN \leftarrow$ 
                 $\frac{|p1-p2|+|p2-p3|+|p3-p4|+|p4-p5|+|p5-p6|+|p6-p7|+|p7-p8|+|p8-p1|}{2}$ 
            if  $CN < 2$  then
                 $EOF \leftarrow [EOF; i, j]$ 
            else if  $CN > 2$  then
                 $Y \leftarrow [Y; i, j]$ 
return  $C$ 

```

---

Le résultat de l'extraction des défauts sur le squelette est présenté dans la Figure 3.4e, avec en vert les points en Y et en rouge les points en fin de ligne.

### 3.2.7 Calcul des dimensions des lignes

L'analyse dimensionnelle des lignes consiste en l'extraction de différentes grandeurs. Nous souhaitons extraire le diamètre des lignes (autrement appelé CD pour Critical Dimension), tout au long de la ligne ainsi que la rugosité des lignes, c'est à dire les variations de ces dernières.

L'extraction du diamètre ou largeur de ligne est basée sur la transformée de distances de l'image binaire, présentée dans l'article de Felzenszwalb and Huttenlocher (2012). La transformée de distances va calculer pour chaque point non nul de l'image binaire, la distance au bord le plus proche. On

### 3.2. ANALYSE D'IMAGE LAMELLAIRE

---

peut observer sur la Figure 3.4d un résultat de transformée de distances. Les valeurs de distances obtenues sur le squelette permettent d'avoir un demi-CD. Les valeurs obtenues sont donc multipliées par 2. À partir de cette distribution, on obtient le CD et le LWR.

On calcule également pour chaque point du bord de ligne, sa distance au squelette morphologique, ce qui nous donne la distribution de valeurs permettant d'extraire le LER.

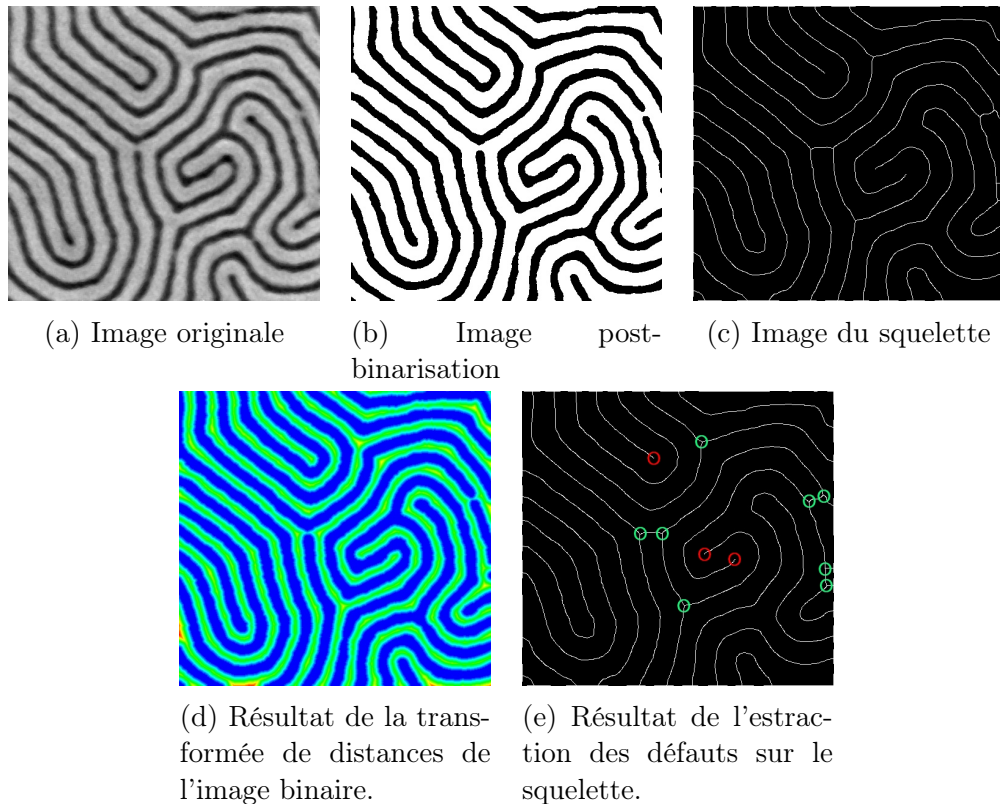


FIGURE 3.4 – Exemple des images intermédiaires lors de l'analyse d'image MEB de lamellaire.

#### 3.2.8 Conclusion

Dans cette section nous avons présenté un algorithme d'analyse d'image de lamellaire dirigé par les données. Les choix techniques de chaque étape ne sont pas nécessairement les techniques optimales à l'état de l'art mais cet assemblage de techniques permet une caractérisation complète et robuste d'images de microscopie électronique à balayage de lamellaire. Cet algorithme est intégré dans le logiciel Platypus<sup>TM</sup> de POLLEN METROLOGY.

### 3.3 Méthodes de régression

La première étape pour faire de l'optimisation de paramètres de procédé est d'être en mesure de prédire le résultat pour des paramètres de procédé qui ne sont pas dans la base d'apprentissage. En effet, dans notre cas, nous avons à notre disposition des résultats d'un premier plan d'expérience, sous la forme d'un jeu de données avec les paramètres et les résultats associés. Nous supposons que ce premier plan d'expérience couvre l'espace des paramètres. Nous voulons être en mesure d'estimer l'espace de taille minimum qui permet d'obtenir les résultats souhaités. Afin d'estimer cette fenêtre, la première étape est l'estimation du résultat sur l'espace des paramètres. Pour effectuer cette étape, nous avons besoin de réaliser une étape de prédiction. Nous présentons dans cette section différentes techniques de régression.

#### 3.3.1 Description des données

Dans notre application nous avons d'un côté les paramètres de procédé de fabrication et les réponses associées, c'est-à-dire les résultats de l'analyse présentée dans la section précédente.

Nous allons modéliser les paramètres de procédé de fabrication, qui sont les variables explicatives, comme un élément d'un sous ensemble de  $\mathbb{R}^p$ . Nous allons les noter :

$$\mathbf{x} = (x_1, \dots, x_p)^\top \in \mathcal{P} \subset \mathbb{R}^p$$

Nous modélisons les réponses associées, qui sont les variables expliquées, comme étant des éléments d'un sous-ensemble de  $\mathbb{R}^q$ . Nous allons les noter :

$$\mathbf{y} = (y_1, \dots, y_q)^\top \in \mathcal{Q} \subset \mathbb{R}^q$$

Les variables de réponses pourront également être notées  $y_k = y_k(\mathbf{x})_{k=1, \dots, q}$ .

Dans notre cas, nous disposons de  $n$  réalisations. Nous avons alors les couples  $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1, \dots, n}$ .

L'objectif est le suivant : prédire la variable réponse  $\hat{\mathbf{y}}_r$  étant donné un nouveau  $\mathbf{x}_r \in \mathbb{R}^p$ . Nous avons choisi d'aborder via différentes méthodes de régression : régressions linéaires pénalisées, interpolation par pondération inverse à la distance, krigeage, réseaux de neurones et forêts aléatoires. Cette liste est loin d'être exhaustive. Néanmoins les méthodes citées précédemment couvrent un large spectre de techniques récentes comme l'illustre le livre de [Friedman et al. \(2001\)](#).



### 3.3.2 Régressions linéaires pénalisées

Dans cette section nous allons présenter les techniques de régressions linéaires pénalisées lasso (pour Least Absolute Shrinkage and Selection Operator) et ridge, présentées dans le livre de [Friedman et al. \(2001\)](#). Les deux méthodes suivent la même logique. L'idée est d'appliquer une combinaison linéaire des variables explicatives en ajoutant une contrainte sur la norme des coefficients. Nous considérons dans ce travail  $q$  modèles linéaires :

$$y_{i,k} = \sum_{j=1}^p \beta_{k,j} x_{i,j} + \beta_{k,0} + \epsilon_k \quad , \quad i = 1, \dots, n \text{ et } k = 1, \dots, q.$$

Les estimateurs ridge et lasso des coefficients sont respectivement définis par :

$$\hat{\beta}_k^{ridge} = \operatorname{argmin}_{\beta_k} \left\{ \sum_{i=1}^n \left( y_{i,k} - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

et

$$\hat{\beta}_k^{lasso} = \operatorname{argmin}_{\beta_k} \left\{ \frac{1}{2} \sum_{i=1}^n \left( y_{i,k} - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}$$

où  $\lambda > 0$ .

Les estimateurs de la variable par les méthodes ridge et lasso sont respectivement :

$$\hat{\mathbf{y}}_{r,k}^{ridge} = \hat{\beta}_{k,0}^{ridge} + \sum_{j=1}^p \hat{\beta}_{k,j}^{ridge} x_{r,j}$$

et

$$\hat{\mathbf{y}}_{r,k}^{lasso} = \hat{\beta}_{k,0}^{lasso} + \sum_{j=1}^p \hat{\beta}_{k,j}^{lasso} x_{r,j}$$

avec  $k \in [1, q]$

La pénalisation  $L^2$  (ridge) a été introduite pour améliorer les estimateurs par moindres carrés en présence de forte colinéarité. La pénalisation  $L^1$  (le lasso) permet quant à elle de mettre à 0 les coefficients de faible amplitude.

La difficulté des méthodes lasso et ridge est de fixer le  $\lambda$ . Afin de fixer ce paramètre de régularisation, nous avons mis en place une validation croisée par k-fold, présentée dans le livre de [James et al. \(2013\)](#). Une perspective de cette section serait de considérer une pénalisation pour un modèle linéaire multivarié.

### 3.3.3 Interpolation par pondération inverse à la distance

Le concept d'interpolation par pondération inverse à la distance est intuitif. Le principe est d'estimer la valeur d'un nouveau point  $\mathbf{x}_r$  comme une combinaison linéaire des  $\mathbf{y}_i$  en ajoutant une pondération proportionnelle à l'inverse de la distance de  $\mathbf{x}_r$  aux  $\mathbf{x}_i$ . Dans notre cas, nous nous sommes intéressés à l'interpolation de Shepard, présentée dans le papier de [Rap et al. \(2009\)](#).

L'expression de l'interpolation par Shepard est la suivante :

$$\hat{\mathbf{y}}_r^{shepard} = \begin{cases} \mathbf{y}_i & \text{si } \mathbf{x}_r = \mathbf{x}_i \\ \frac{\sum_{i=0}^n w_i(\mathbf{x}_r) \mathbf{y}_i}{\sum_{j=0}^n w_j(\mathbf{x}_r)} & \text{si } \mathbf{x}_r \neq \mathbf{x}_i \end{cases}$$

où les  $w_i$  sont des pondérations données par :

$$w_i(\mathbf{x}_r) = \begin{cases} \left[ \frac{R_w - \|\mathbf{x}_r - \mathbf{x}_i\|}{R_w \|\mathbf{x}_r - \mathbf{x}_i\|} \right]^2 & \text{si } \|\mathbf{x}_r - \mathbf{x}_i\| < R_w \\ 0 & \text{si } \|\mathbf{x}_r - \mathbf{x}_i\| \geq R_w. \end{cases}$$

La difficulté de ce type d'interpolation est de fixer le rayon  $R_w$ . On peut fixer le poids en fonction de la distribution spatiale des  $\mathbf{x}_i$  afin d'avoir toujours une portion de la population de départ prise en compte. Nous avons dans notre cas normalisé les points  $\mathbf{x}_i$  entre 0 et 1 afin de simplifier la détermination de  $R_w$ .

### 3.3.4 Krigeage

Le krigeage est une méthode qui tient compte de la dépendance spatiale des données. Initialement utilisée en géoscience par [Krige \(1951\)](#) puis formalisée par le mathématicien français [Matheron \(1963\)](#), elle est maintenant utilisée dans la plupart des domaines. Le papier de [Roustant et al. \(2012\)](#) présente le krigeage comme un outil couramment utilisé comme modèle de substitution dans les expériences numériques, duquel nous nous sommes inspirés pour la présentation de la méthode.

Posons  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , et  $\mathbf{Y}_k = (y_k(\mathbf{x}_1), \dots, y_k(\mathbf{x}_n))^T$ , l'ensemble des observations. On suppose que les  $y_k$  sont les réalisations d'un champ aléatoire de premier et second moments connus. Pour un  $\mathbf{x}_r$ , le principe du krigeage est d'estimer de façon optimale  $y_{r,k}$  comme une combinaison linéaire des  $y_{i,k}$  :

$$\hat{y}_{r,k}^{krigeage} = \lambda(\mathbf{x}_r)^T \mathbf{Y}_k = \sum_{i=1}^n \lambda_i(\mathbf{x}_r) y_k(\mathbf{x}_i)$$

### 3.3. MÉTHODES DE RÉGRESSION

---

Dans le krigeage simple et le krigeage ordinaire, on cherche le processus  $Y$  comme la somme d'une fonction à tendance déterministe connue  $\mu : \mathbf{x} \in \mathbb{R}^p \rightarrow \mu(\mathbf{x}) \in \mathbb{R}$  et d'un processus centré de carré intégrable  $Z$  :

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}) \quad (3.1)$$

Dans la suite, nous noterons  $Y(\mathbf{x})$  la valeur du processus au point  $\mathbf{x}$  et  $Y(\mathbf{X})$  le vecteur des valeurs du processus aux points de  $\mathbf{X}$ .

La covariance de  $Z$  est connue et est notée  $C : (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^p \times \mathbb{R}^p \rightarrow C(\mathbf{u}, \mathbf{v}) \in \mathbb{R}$ . Nous supposons le processus  $Y$  centré. S'il ne l'est pas, on peut aisément le centrer en appliquant  $Y(\mathbf{x}) - \mu(\mathbf{x})$ . Le meilleur estimateur non biaisé est celui qui va minimiser l'erreur quadratique moyenne :

$$\text{MSE}(\mathbf{x}) := E \left[ (Y(\mathbf{x}) - \lambda(\mathbf{x})^\top Y(\mathbf{X}))^2 \right] \quad (3.2)$$

Soit  $\mathbf{C} = (C(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n}$  la matrice de covariance de  $Y(\mathbf{X})$  et  $\mathbf{c}(\mathbf{x}) = (C(\mathbf{x}, \mathbf{x}_i))_{1 \leq i \leq n}$  la covariance de  $Y(\mathbf{X})$  avec  $Y(\mathbf{x})$ . Si  $\mathbf{C}$  est inversible, alors :

$$\lambda^*(\mathbf{x}) = \mathbf{C}^{-1} \mathbf{c}(\mathbf{x})$$

Si on remplace  $Y(\mathbf{X})$  par les réalisations  $\mathbf{Y}_k$  et  $\lambda(\mathbf{x})$  par  $\lambda^*(\mathbf{x})$  dans l'équation 3.2, on obtient :

$$\begin{aligned} m_{sk}(\mathbf{x}) &= \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1} \mathbf{Y}_k \\ s_{sk}^2(\mathbf{x}) &= C(\mathbf{x}, \mathbf{x}) - \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1} \mathbf{c}(\mathbf{x}). \end{aligned}$$

Si on généralise et décentre  $Y$ , on a alors :

$$m_{sk}(\mathbf{x}) = \mu(\mathbf{x}) + \mathbf{c}(\mathbf{x})^\top \mathbf{C}^{-1} (\mathbf{Y}_k - \mu)$$

Avec  $\mu = \mu(\mathbf{X})$  la tendance aux positions des observations.

Lors du simple krigeage,  $Z$  est supposé être un processus gaussien. L'orthogonalité entre  $\mathbf{Y}(\mathbf{x}) - \lambda^*(\mathbf{x})^\top Y(\mathbf{X})$  et  $Y(\mathbf{X})$  assure l'indépendance, ce qui donne  $E[\mathbf{y}_{r,k} | Y(\mathbf{X}) = \mathbf{Y}_k] = m_{sk}(\mathbf{x})$ . De même, nous avons  $\text{Var}[\mathbf{y}_{r,k} | Y(\mathbf{X}) = \mathbf{Y}_k] = s_{sk}^2(\mathbf{x})$ . On peut alors écrire la loi conditionnelle de  $\mathbf{y}_{r,k}$  en fonction des grandeurs du krigeage simple :

$$\mathbf{y}_{r,k} | Y(\mathbf{X}) = \mathbf{Y}_k \sim \mathcal{N}(m_{SK}(\mathbf{x}), s_{SK}^2(\mathbf{x}))$$

Pour éviter les problèmes numériques de l'estimation de  $\mathbf{C}^{-1}$ , le krigeage peut être enrichi par un effet pépète plus couramment appelé effet nugget en anglais, présenté dans l'article de [Roustant et al. \(2012\)](#). L'effet de pépète correspond à la variance en zéro. On peut supposer que dans 3.1,  $Y$  est la

superposition de  $Z$  et d'un bruit blanc gaussien de faible variance. L'effet pépite permet donc de représenter une variation entre des mesures très proches et ainsi une prise en compte de plusieurs phénomènes comme par exemple des mesures faites à des instants différents, une instabilité provenant de la façon dont est extraite la mesure (capteur bruité par exemple) ou encore un réel effet pépite qui implique un brutal changement de valeur.

### 3.3.5 Réseau de neurones

Les réseaux de neurones sont présentés par exemple dans le cours de [Parizeau \(2004\)](#). Le principe est simple, chaque neurone est une combinaison linéaire des entrées suivie d'une fonction d'activation. La Figure 3.1 présente le schéma de fonctionnement d'un neurone. On peut représenter un neurone comme :

$$\alpha = f\left(\sum_{i=1}^R w_{i,j} p_i - b\right)$$

où  $w_{i,j}$  représente le poids de la couche  $j$ ,  $f$  la fonction d'activation,  $b$  est le seuil d'activation du neurone. La fonction d'activation est une fonction qui permet d'appliquer un seuil sur le résultat du neurone. Les fonctions d'activation souvent utilisées pour notre application sont des tangentes hyperboliques, des sigmoïdes ou bien une fonction RELU qui veut dire Unité de Rectification Linéaire. Celle-ci retourne 0 si l'entrée est inférieure ou égale à 0 et identité sinon.

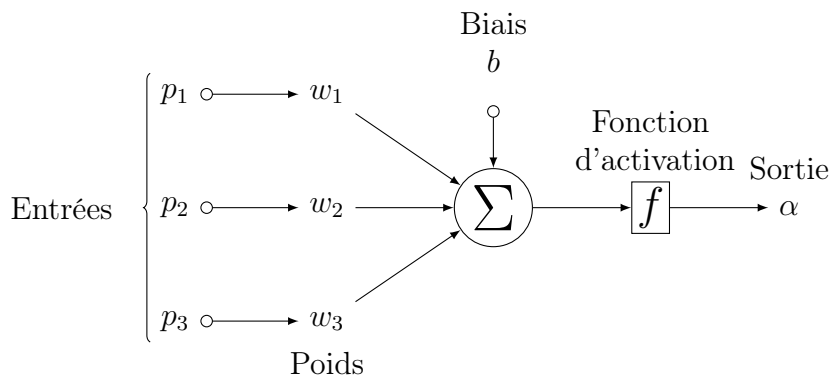


FIGURE 3.1 – Schéma d'un neurone.

Les poids sont optimisés couche par couche de la dernière à la première couche, par rétro-propagation des erreurs dans le réseau. Les erreurs sont calculées par validation croisée.

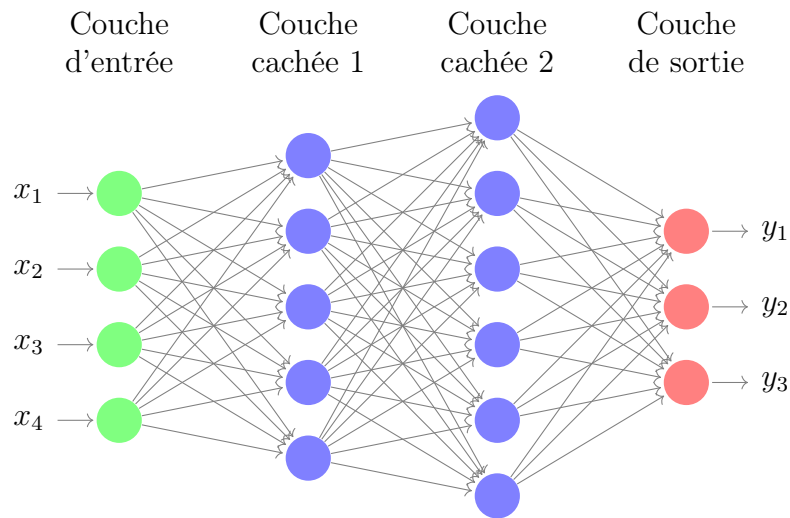


FIGURE 3.2 – Schéma d'un réseau de neurones à couches totalement connectées avec 4 couches.

La structure du réseau de neurones peut-être multiple. Nous allons présenter uniquement le type de structure que nous avons choisi d'utiliser dans nos travaux : les réseaux de neurones à couches totalement connectées. La Figure 3.2 présente un exemple de ce type de réseau.

Le principe de l'architecture de ce type de réseaux est que la sortie de chaque neurone est en entrée de tous les neurones constituant la couche suivante. La seule contrainte de ce type de réseaux est la dimension de la dernière couche du réseau qui doit avoir la même dimension que le résultat qu'on souhaite estimer. Dans notre cas, le nombre de neurones de la dernière couche du réseau est égal à  $q$ . Le nombre de couches et de neurones par couche sont des paramètres qu'il faut fixer.

Dans notre cas, nous avons choisi de faire un réseau composé de 4 couches. Pour les fonctions d'activations, nous avons choisi un mélange de fonctions RELU et de tangente hyperbolique. Pour le nombre de neurones dans les couches cachées le nombre est proportionnel à la dimension des données d'entrées.

### 3.3.6 Forêt aléatoire : Random Forest

La technique de Random Forest ou forêt aléatoire en français a été formalisée par Breiman (2001) et est décrite dans le livre de Friedman et al. (2001). Le principe de construction d'un modèle de Random Forest est le suivant : on

applique un bootstrap sur les données en générant  $B$  échantillons de même taille que celui d'entrée. Pour chaque échantillon bootstrap, on entraîne un arbre de décision à partir de l'échantillon en limitant sa croissance par validation croisée que l'on nommera  $T_b$ .

Un exemple d'arbre de décision est présenté sur la Figure 3.3. Un arbre de décision prend en entrée une valeur, de dimension quelconque, puis à chaque nœud pose une condition. Suivant si la condition est respectée ou non, la donnée va au nœud enfant correspondant. L'opération est répétée jusqu'à arriver à une feuille de l'arbre, qui contient la décision.

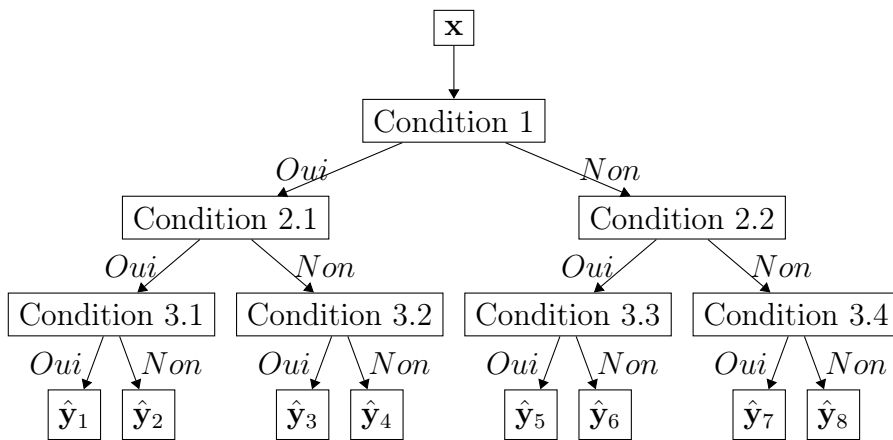


FIGURE 3.3 – Schéma d'un arbre de décision de profondeur 3.

La construction de l'arbre est faite en partant du sommet et en allant jusqu'aux feuilles. L'article de [Rokach and Maimon \(2005\)](#) présente un algorithme de construction d'arbre de décision. Partant du sommet, on calcule le meilleur partage des données. Pour de la classification on cherchera à minimiser la variance de chaque sous-ensemble tandis que pour de la régression le but sera de la maximiser. La variable de séparation de chaque nœud est choisie par validation croisée. Ensuite, peut intervenir une étape d'élagage, qui consiste en la suppression des sous parties de l'arbre qui contiennent trop peu de valeurs ou bien qui dépassent une profondeur souhaitée sont supprimée. L'élagage peut être fait pendant la construction de l'arbre, on parle alors de pré-élagage.

Dans les forêts aléatoires, on a construit  $B$  arbres de décisions. Pour effectuer une régression, l'estimation du nouveau point est faite comme la moyenne des  $B$  estimations. On peut alors écrire l'estimation comme :

$$\hat{\mathbf{y}}_r^{random\_forest} = \frac{1}{B} \sum_{b=1}^B T_b(\mathbf{x}_r).$$

Les difficultés de la méthode sont la fixation du nombre d'échantillons de bootstrap  $B$  et le choix de la profondeur de chaque arbre de décision, généralement fixée par validation croisée.

## 3.4 Résultats

Nous allons appliquer les méthodes de régression à nos données. Nous disposons d'un jeu de données constitué de  $n$  expériences avec 3 paramètres qui varient. Nous avons dans notre cas, 3 paramètres de procédés de fabrication. Pour chaque réalisation, nous avons à notre disposition une dizaine d'images, prises à des endroits différents avec trois niveaux de zoom différents. Nous avons alors  $\mathbf{x} = (x_1, x_2, x_3)^\top \in \mathcal{P} \subset \mathbb{R}^3$  et  $\mathbf{y} = (y_1, \dots, y_{23})^\top \in \mathcal{Q} \subset \mathbb{R}^{23}$ . Dans notre cas, nous normalisons toutes nos données entre 0 et 1 pour chaque dimension de  $\mathbf{x}$  et  $\mathbf{y}$ , ce qui revient à dire que  $\mathcal{P} = [0, 1]^3$  et  $\mathcal{Q} = [0, 1]^{23}$ .

Nous avons séparé notre jeu de données en deux sous-ensembles. Un jeu de donnée pour l'apprentissage, qui contient deux tiers des données et le tiers restant est le jeu de validation. Nous allons calculer des métriques de performance pour estimer la qualité de l'estimation sur le sous-ensemble de validation, ceux-ci sont définis dans la section d'après.

### 3.4.1 Critères de performance

Nous allons utiliser les métriques de performances normalisées suivantes :

1. Erreur quadratique moyenne normalisée ou root normalized mean-squared-error en anglais, définie par :

$$RNMSE_k = \left\{ \frac{\sum_i (y_{i,k} - \hat{y}_{i,k})^2}{\sum_i (y_{i,k} - \bar{y}_k)^2} \right\}^{\frac{1}{2}}$$

avec  $k \in [1, 23]$ .

2. L'erreur absolue relative moyenne ou Relative average absolute error en anglais, définie par :

$$RAAE_k = \left\{ \frac{\sum_i |y_{i,k} - \hat{y}_{i,k}|}{n \times STD(y_k)} \right\}^{\frac{1}{2}}$$

où  $STD(y_k)$  est l'écart-type du vecteur  $y_k$  qui contient la dimension  $k$  des vecteurs  $\mathbf{y}$  et  $k \in [1, 23]$ .

3. l'erreur quadratique moyenne normalisée en échelle logarithmique ou root normalized mean-squared-error on log scale en anglais, définie par :

$$RNMSE_{log_k} = \left\{ \frac{\sum_i (\log(|y_{i,k}| + 1) - \log(|\hat{y}_{i,k}| + 1))^2}{\sum_i (\log(|y_{i,k}| + 1) - \log(|\bar{y}_k| + 1))^2} \right\}^{\frac{1}{2}}$$

avec  $k \in [1, 23]$ .

Chaque métrique est définie pour une dimension de  $\mathbf{y}$ . L'erreur globale est calculée comme la moyenne sur toutes les dimensions de  $\mathbf{y}$ . Le fait d'avoir normalisé chaque dimension de  $\mathbf{y}$  entre  $[0, 1]$  permet d'homogénéiser l'impact des dimensions sur le calcul de l'erreur.

### 3.4.2 Résultats régression

Nous avons appliqué les méthodes de régression présentées précédemment sur les échantillons d'apprentissage et de validation.

#### Premiers résultats

Les premiers résultats des critères de performance sur le jeu de validation sont présentés sur la table 3.1. Les chiffres sont loin de décrire des estimateurs performants. En effet, toutes les valeurs des critères de performance sont supérieures à 0,5.

	$\overline{RNMSE}$	$\overline{RAAE}$	$\overline{RNMSE}_{log}$
Krigeage	0.63	0.56	0.59
Shepard	0.67	0.59	0.60
Neural network	0.79	0.64	0.65
Random forest	0.74	0.63	0.63
Lasso	0.76	0.65	0.64
Ridge	0.75	0.65	0.64

TABLE 3.1 – Résultats des différents critères de performance en fonction des méthodes de régression.

Afin de détecter la source du problème, nous avons étudié les résultats obtenus en traçant l'erreur par dimension et en observant les résultats de la régression pour les différentes dimensions de sortie.

La Figure 3.1 présente un exemple de résultat de régression sur la base de validation pour chaque méthode. Nous observons que les régressions donnent



### 3.4. RÉSULTATS

---

des paliers. En effet, nos données de validation sont rangées par jeu de paramètre d'entrée. Étant donné que pour une valeur de paramètres, nous avons une dizaine d'images prises avec 3 niveaux de zoom différents, les données de validations sont donc rangées par paquets d'une dizaine de valeurs ayant les mêmes paramètres. Il est donc normal d'obtenir des paliers pour la régression. Cependant, on observe que, pour chacun de ces paliers, le signal théorique subit de fortes variations.

La Figure 3.2 présente le résultat de l'interpolation sur une des variables de sortie, ce qui revient à la superposition des courbes de la Figure 3.1. On observe que pour chaque palier, les régressions sont proches mais ne sont pas exactement positionnées à la même valeur. Pour chaque jeu de paramètres, on observe un pic, une zone intermédiaire puis un creux. Les trois zones correspondent aux trois niveaux de zoom de la prise d'image.

#### Intégration du zoom

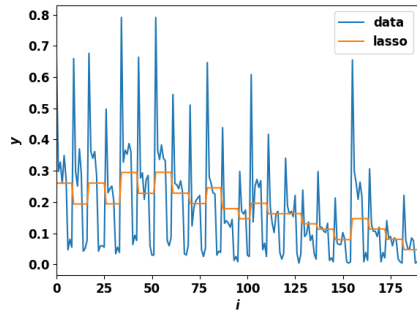
Pour améliorer les résultats de régression, nous avons alors rajouté la valeur du zoom aux paramètres. La Figure 3.3 présente un exemple de résultat de régression sur la base de validation pour chaque méthode en intégrant le zoom dans les paramètres. On observe que le résultat de chaque technique est meilleur que ceux de la Figure 3.1.

L'impact sur les critères de performance est présenté dans la table 3.2. On constate une nette amélioration sur les valeurs de critères en comparaison à la table 3.1.

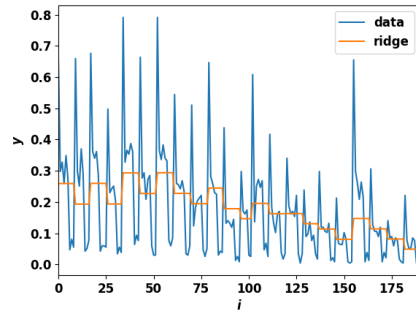
	$\overline{RNMSE}$	$\overline{RAAE}$	$\overline{RNMSE\_log}$
Krigeage	0.39	0.42	0.44
Shepard	0.44	0.45	0.46
Neural network	0.46	0.47	0.48
Random forest	0.51	0.52	0.52
Lasso	0.54	0.54	0.53
Ridge	0.54	0.54	0.53

TABLE 3.2 – Résultats des critères de performance en fonction des méthodes de régression en intégrant le niveau de zoom en paramètre.

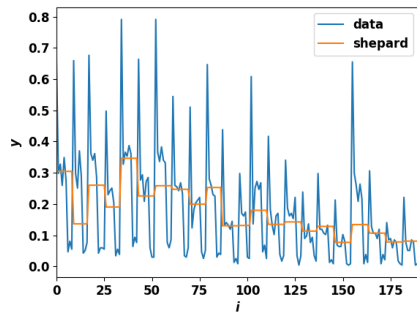
Si on trace le même graphique que celui de la Figure 3.2, nous obtenons le graphique présenté sur la Figure 3.4. On observe que les valeurs sont mieux prédites. Il reste cependant une forte variation de celles-ci pour avoir une prédiction de qualité. D'autre part, les différents résultats de régression donnent



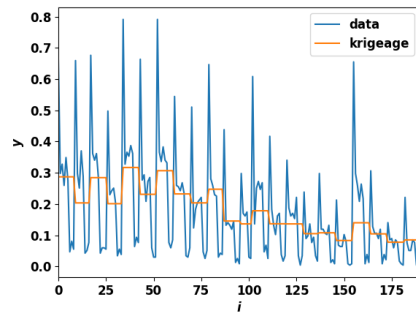
(a) Lasso



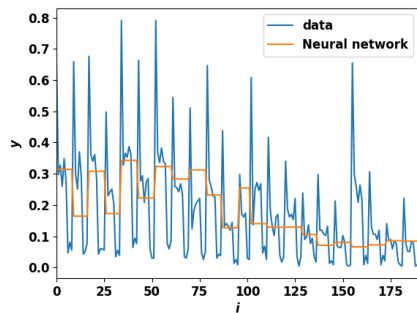
(b) Ridge



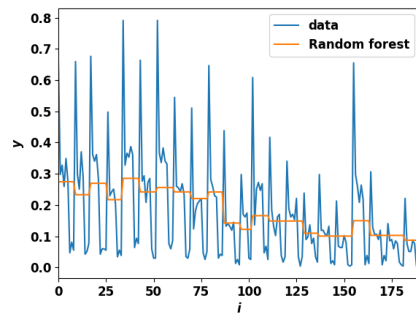
(c) Shepard



(d) Krigage



(e) Réseau de neurones



(f) Forêt aléatoire

FIGURE 3.1 – Exemples de résultats de régression pour les différentes méthodes. En abscisse on a l'indice de l'observation dans la base de validation. La courbe bleue représente les données et la courbe orange la courbe des estimations.

### 3.4. RÉSULTATS

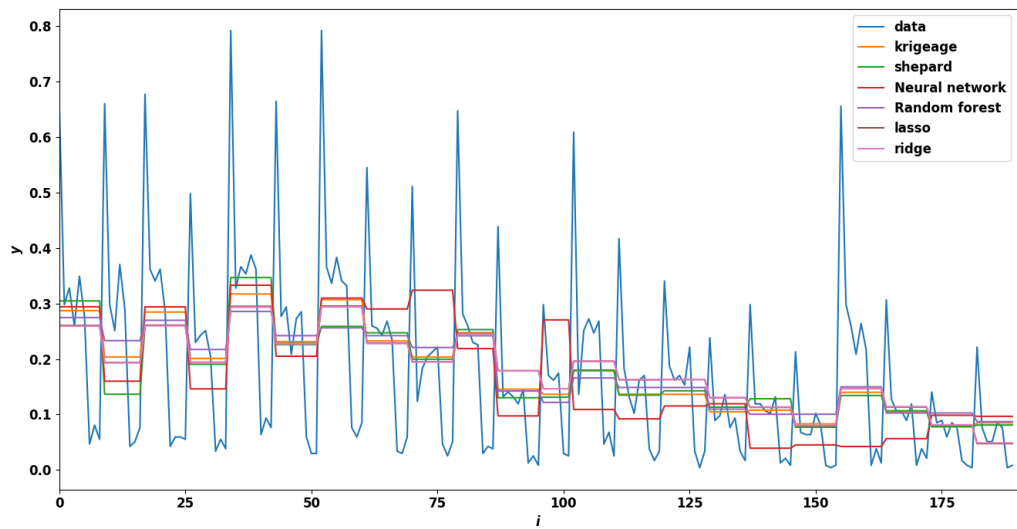


FIGURE 3.2 – Exemple de résultats de régression pour les 6 méthodes testées sur le jeu de validation. En abscisse, l’index dans le jeu de validation, en ordonnée le résultat associé.

les mêmes tendances sans être positionnés aux même valeurs mais semblent plus proche que si le zoom n’est pas pris en compte dans les paramètres.

Les courbes des valeurs des critères de performance sont présentées sur les Figures 3.5, 3.6 et 3.7. On observe sur les trois graphiques que les erreurs varient énormément en fonction de la dimension à estimer. Certaines valeurs des paramètres de sortie semblent en effet plus simple à estimer que d’autre. Les résultats sont cohérents car toutes les erreurs des méthodes de régression suivent la même tendance.

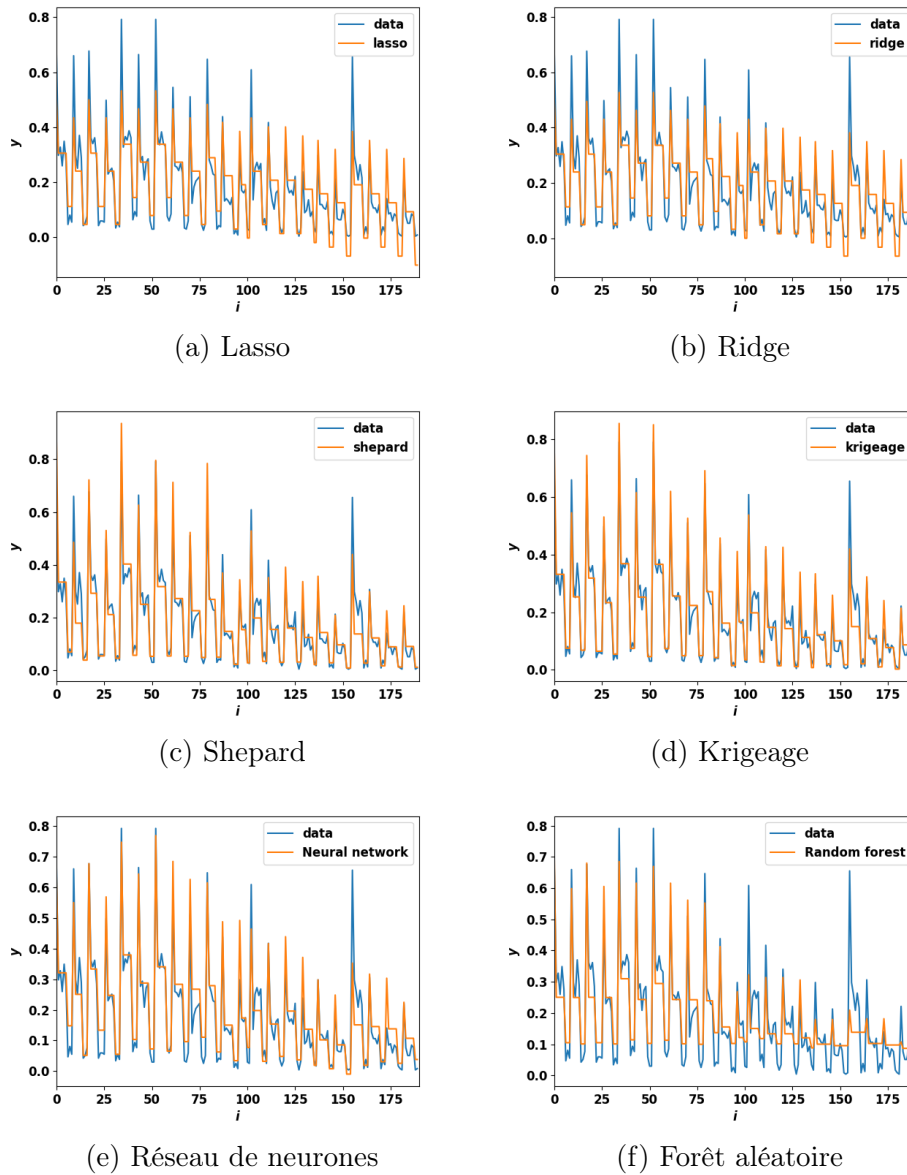


FIGURE 3.3 – Exemples de résultats de régression pour les différentes méthodes. En abscisse on a l'indice de l'observation de la base de validation. La courbe bleue représente les données et la courbe orange la courbe des estimations.

### 3.4. RÉSULTATS

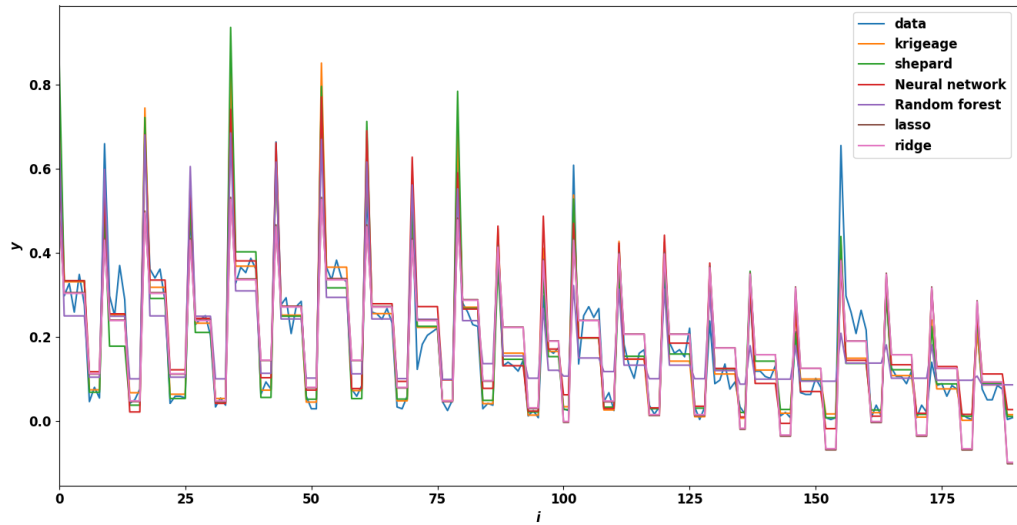


FIGURE 3.4 – Exemple de résultats de régression pour les 6 méthodes testées sur le jeu de validation. En abscisse, l'index dans le jeu de validation, en ordonnée le résultat associé.

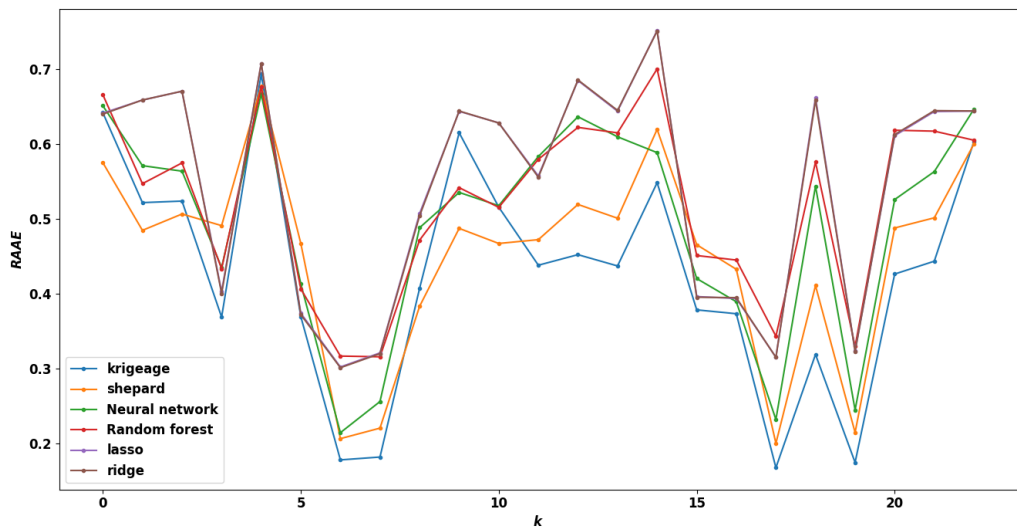


FIGURE 3.5 – Tracé des valeurs du critère de performance  $RAAE$  pour les différentes méthodes de régression en fonction de la dimension  $k \in [1, 23]$ .

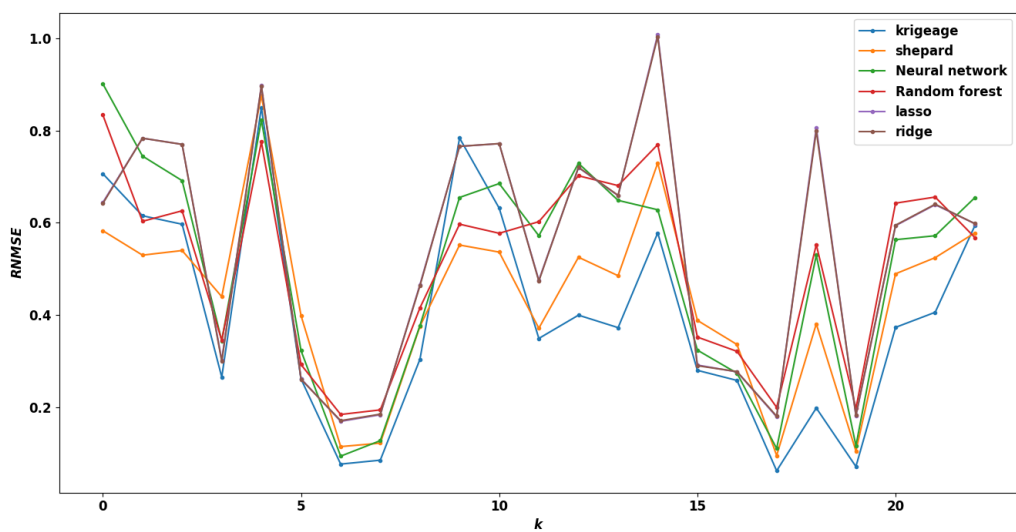


FIGURE 3.6 – Tracé des valeurs du critère de performance  $RNMSE$  pour les différentes méthodes de régression en fonction de la dimension  $\mathbf{k} \in [1, 23]$ .

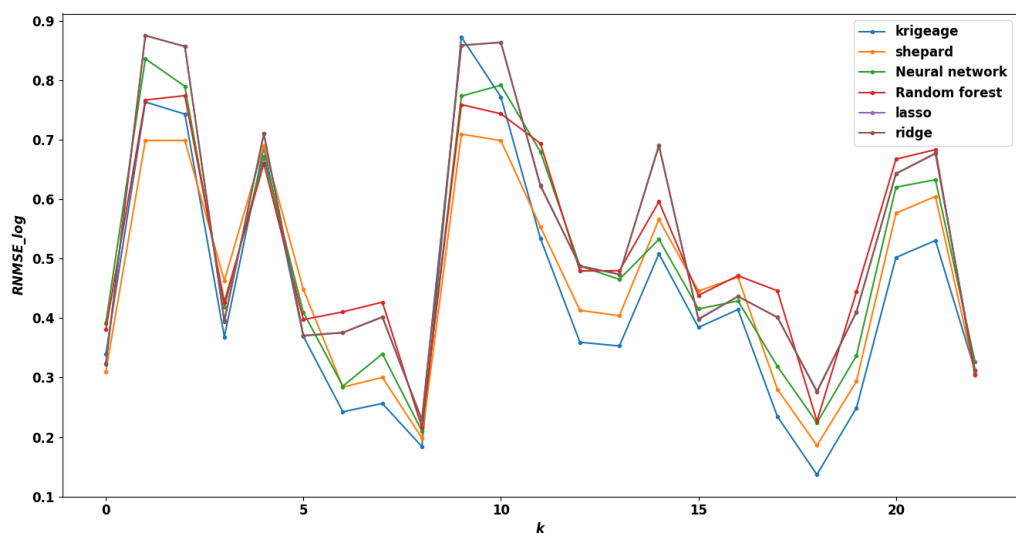


FIGURE 3.7 – Tracé des valeurs du critère de performance  $RNMSE_{log}$  pour les différentes méthodes de régression en fonction de la dimension  $\mathbf{k} \in [1, 23]$ .

## 3.5 Conclusion et discussion

Les résultats présentés précédemment montrent que certaines dimensions sont difficilement estimables. L'intégration du zoom montre une amélioration des résultats de régression mais il reste beaucoup de variations des valeurs pour une valeur de zoom fixée. Pour compenser cette problématique, une solution serait d'augmenter le nombre de prises de mesure. En augmentant le nombre de prises de mesure, soit le nombre d'images, et donc le nombre de données à notre disposition, nous pourrions par exemple estimer pour chaque valeur de paramètre une unique réalisation et ne pas tenir compte du niveau de zoom.

En conclusion, nous avons prouvé que les méthodes de régression permettent d'avoir une estimation plus ou moins performante de la valeur du résultat pour des nouveaux paramètres. Cependant, pour aller plus loin, nous aurions pu chercher à améliorer les techniques comme les réseaux de neurones en jouant sur l'architecture et les fonctions d'activation. D'autre part, avec un jeu de données plus conséquent, nous aurions pu nous orienter vers des méthodes de régression quantile.

Une autre piste d'amélioration intéressante serait de faire de l'agrégation d'estimateurs sur les méthodes de régression. Nous pourrions en effet appliquer un bootstrap sur le jeu de données, puis calculer la matrice de covariance des méthodes de régression qui sont nos estimateurs. Ceci nous permettrait alors calculer les poids associés à chacune de nos méthodes pour en extraire la combinaison linéaire à appliquer.

# Bibliographie

- Breiman, L. (2001). Random forests. *Machine learning*, 45(1) :5–32.
- Farina, A., Kovács-Vajna, Z. M., and Leone, A. (1999). Fingerprint minutiae extraction from skeletonized binary images. *Pattern Recognition*, 32(5) :877 – 889.
- Felzenszwalb, P. F. and Huttenlocher, D. P. (2012). Distance transforms of sampled functions. *Theory of Computing*, 8(1) :415–428.
- Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA :.
- Geffray, S., Klutchnikoff, N., and Vimond, M. (2016). Illumination problems in digital images. a statistical point of view. *Journal of Multivariate Analysis*, 150 :191–213.
- Gonzalez, R. C., Woods, R. E., et al. (2002). Digital image processing.
- Harrison, C., Adamson, D. H., Cheng, Z., Sebastian, J. M., Sethuraman, S., Huse, D. A., Register, R. A., and Chaikin, P. M. (2000). Mechanisms of Ordering in Striped Patterns. *Science*, 290(5496) :1558–1560.
- Harrison, C., Cheng, Z., Sethuraman, S., Huse, D. A., Chaikin, P. M., Vega, D. A., Sebastian, J. M., Register, R. A., and Adamson, D. H. (2002). Dynamics of pattern coarsening in a two-dimensional smectic system. *Phys. Rev. E*, 66 :011706.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.
- Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6) :119–139.



## BIBLIOGRAPHIE

---

- Mahmood, S. A. (2011). Fingerprint identification based on skeleton minutiae extraction. In *ICGST Conference*, pages 12–14.
- Matheron, G. (1963). Principles of geostatistics. *Economic geology*, 58(8) :1246–1266.
- Murphy, J. N., Harris, K. D., and Buriak, J. M. (2015). Automated defect and correlation length analysis of block copolymer thin film nanopatterns. *PLoS ONE*, 10(7) :e0133088.
- Parizeau, M. (2004). Réseaux de neurones. *GIF-21140 et GIF-64326*, 124.
- Park, C., Yoon, J., and Thomas, E. L. (2003). Enabling nanotechnology with self assembled block copolymer patterns. *Polymer*, 44(22) :6725 – 6760.
- Rap, A., Ghosh, S., and Smith, M. H. (2009). Shepard and hardy multiquadric interpolation methods for multicomponent aerosol–cloud parameterization. *Journal of the Atmospheric Sciences*, 66(1) :105–115.
- Rokach, L. and Maimon, O. (2005). Top-down induction of decision trees classifiers—a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4) :476–487.
- Roustant, O., Ginsbourger, D., and Deville, Y. (2012). Dicekriging, diceoptim : Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization.
- Ryu, H. J., Fortner, D. B., Lee, S., Ferebee, R., De Graef, M., Misichronis, K., Avgeropoulos, A., and Bockstaller, M. R. (2013). Role of grain boundary defects during grain coarsening of lamellar block copolymers. *Macromolecules*, 46(1) :204–215.
- Williamson, L., Kim, J., Cao, Y., Lin, G., Gronheid, R., and Nealey, P. F. (2015). Impact of bcp asymmetry on dsa patterning performance.
- Zaeri, N. (2011). Minutiae-based fingerprint extraction and recognition. In *Biometrics*. InTech.
- Zhang, T. Y. and Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3) :236–239.
- Zhang, X., Harris, K. D., Wu, N. L., Murphy, J. N., and Buriak, J. M. (2010). Fast assembly of ordered block copolymer nanostructures through microwave annealing. *ACS nano*, 4(11) :7021–7029.

# Chapitre 4

## Intégration dans les produits Pollen

### Introduction

Dans les chapitres précédents, nous avons présenté différentes méthodes et outils qui répondent à des problématiques industrielles. Ces derniers doivent être intégrés dans un logiciel afin d'être diffusés. Pour cela, il faut passer de la phase de prototypage à logiciel. Ce chapitre présente cette transformation. Dans un premier temps, nous présenterons les généralités du développement logiciel dans un contexte industriel pour l'intégration de nouveaux modules dans un logiciel, partant d'un prototype. Nous passerons ensuite à la description des travaux des chapitres précédents qui ont d'ores et déjà été intégrés dans le logiciel Platypus<sup>TM</sup> de POLLEN METROLOGY. Nous présenterons l'intégration du redressement AFM puis de l'analyse d'images MEB et AFM de nanoparticules ainsi que l'agrégation d'estimateurs puis finirons par l'intégration de l'analyse d'images MEB de lamellaire. Pour chacune des parties précédentes, nous présenterons les principales caractéristiques de l'intégration ainsi que des performances obtenues. Dans la mesure où la prédiction de procédés de fabrication est une technique toujours en cours de développement à l'heure actuelle, nous n'en parlerons pas dans ce chapitre. Le développement de cette technique fait l'objet d'un contrat post-doctoral et suivra, dès lors qu'un prototype satisfaisant sera achevé, les étapes présentées dans ce chapitre pour son intégration dans le logiciel.

---

## Sommaire

---

<b>Introduction</b> . . . . .	<b>153</b>
<b>4.1 Généralités sur le développement logiciel</b> . . . . .	<b>155</b>
4.1.1 La modélisation UML . . . . .	155
4.1.2 Chaîne d'intégration . . . . .	155
<b>4.2 Redressement d'images AFM</b> . . . . .	<b>157</b>
<b>4.3 Algorithme d'analyses de nanoparticules</b> . . . . .	<b>159</b>
4.3.1 Analyse d'images MEB et AFM . . . . .	159
4.3.2 Agrégation d'estimateurs . . . . .	161
<b>4.4 Analyse d'images lamellaire</b> . . . . .	<b>165</b>

---

## 4.1 Généralités sur le développement logiciel

Dans ce chapitre, nous présentons le travail permettant de passer d'un travail de recherche à une intégration dans un produit industriel. Pour cela, nous devons suivre un certain nombre d'étapes. Nous présentons dans cette section les principes généraux, afin de ne pas les répéter dans les sections suivantes .

### 4.1.1 La modélisation UML

La première étape de l'intégration est la modélisation en UML (Unified Modeling Language pour Langage de Modélisation Unifié) de la fonctionnalité. Le langage UML permet de définir des concepts d'un système informatique, de façon graphique et normalisée. Il est couramment utilisé en développement logiciel et en conception orientée objet. Dans notre cas, nous nous intéressons à la conception orientée objet. Le langage UML permet de définir les classes, les attributs, les méthodes et les arguments du code de façon graphique.

De façon simpliste, un diagramme UML est composé de boîtes et de traits. Chaque boîte permet de représenter une classe, une structure ou un objet. Chaque ligne, permet de représenter les liens existant entre les classes, les objets et les structures. Pour le lecteur souhaitant approfondir le sujet, nous conseillons le livre de [Roques \(2011\)](#).

Une fois cette étape préliminaire faite, on peut passer au développement du code, en suivant ce diagramme. Ceci permet de bien modéliser le code afin d'éviter des erreurs de conception qui sont trop souvent découvertes en cours d'implémentation et qui obligent à modifier la structure du code. Une pratique assez courante dans le monde industriel est de générer le code à partir du diagramme UML. Le diagramme permet de définir la structure du code, indépendamment du langage de programmation. Ainsi on peut changer de langage de programmation simplement et il ne reste que le contenu des méthodes à compléter.

Dans ce chapitre, nous ne présenterons pas tous les diagrammes UML par soucis de simplicité mais nous détaillerons uniquement celui de l'agrégation d'estimateur, que nous détaillerons.

### 4.1.2 Chaîne d'intégration

Nous présentons ici brièvement comment l'intégration d'une nouvelle fonctionnalité est faite. Partant du diagramme UML, on fait la structure du code. Puis, pour chaque fonction, on implémente des tests dit tests unitaires. Ces

tests sont automatiques, sont exécutés lors de la compilation pour vérifier que le code fait bien ce pourquoi il est conçu. Dans un test unitaire, on va tester une fonction en lui donnant un argument et on vérifie que la sortie est bien celle attendue. Si c'est le cas, le test passe, si ça ne l'est pas, le test ne passe pas et alors le développeur doit corriger son code. Ainsi, lors de l'intégration d'une nouvelle fonctionnalité ou une refactorisation du code, les tests unitaires permettent par exemple de détecter des bugs de régression ou des effets de bord plus rapidement.

Une fois les tests mis en place (ce qui peut être laborieux) on passe à l'implémentation du code à proprement parler. Une fois le code implémenté, on vérifie que tous les tests unitaires passent. Cette méthode s'appelle du développement dirigé par les tests. Un ouvrage intéressant sur le sujet est le livre de [Langr \(2013\)](#).

Une fois qu'une nouvelle fonctionnalité est développée et que les tests passent, vient la demande de tirage ou pull request en anglais, qui est la proposition d'intégration du code dans la base de code globale. Pour que le code soit accepté, il faut qu'il soit validé par la chaîne d'intégration et par les relecteurs de la demande de tirage.

Dans notre cas, nous avons un serveur d'intégration, qui va compiler le code pour différentes plateformes, exécuter les tests sous les différentes plateformes et analyser le code. L'analyse de code consiste en plusieurs points :

- la vérification de la couverture des tests, afin de vérifier que toutes les fonctions sont bien testées.
- vérification des fuites mémoires.
- détection de variables ou de fonctions non utilisées.
- vérification de la sécurité des exceptions, par exemple : utilisation de l'allocation de la mémoire et des vérifications d'utilisation des destructeurs
- vérification de règles de codage.

Il existe beaucoup de types de vérifications du code disponibles via une collection d'outils. Toutes ces vérifications sont exécutées automatiquement lorsqu'on souhaite intégrer une nouvelle fonctionnalité. Le serveur d'intégration fournit un rapport des différentes métriques et si le code ne respecte pas les contraintes alors il doit être amélioré. Ainsi on s'assure de la qualité du code qui est intégré.

La dernière étape est le test fonctionnel qui lui est exécuté à la main dans l'application afin de vérifier que (même si le code est de qualité) la fonctionnalité est bien celle attendue.

## 4.2 Redressement d'images AFM

Nous faisons ici un bref récapitulatif du fonctionnement de l'algorithme de redressement d'images AFM présenté dans le chapitre 1 en détaillant l'implémentation de chaque étape.

L'algorithme consiste dans un premier temps en l'interpolation par une droite de chacune des lignes de l'image par la méthodes des M-estimateurs asymétriques. On applique deux fois l'interpolation afin d'estimer les coefficients de chaque ligne avec chaque type d'asymétrie possible. Une fois les coefficients extraits, nous devons choisir le bon modèle, en gardant les coefficients ayant la plus petite variance. Nous lissons ensuite les coefficients, afin d'améliorer le redressement. Pour cela, nous avons besoin de détecter si le signal des coefficients contient des sauts, afin de prendre ces derniers en compte lors du lissage. La dernière étape est ensuite de redresser l'image AFM en retirant à chaque ligne la droite définie par les coefficients lissés.

**Interpolation de chaque ligne** D'un point de vue implémentation, une ligne est un ensemble de  $n$  valeurs, qui est représenté dans le code par un vecteur de valeurs. L'interpolation d'une ligne par les M-estimateurs asymétriques est fait en utilisant la méthode de moindres carrés pondérés itératifs. Cette méthode a pour avantage de pouvoir être écrite facilement sous forme matricielle. Par exemple, si on pose  $\mathbf{y}$  le vecteur de valeurs à interpoler,  $\mathbf{B}_p$  la matrice de base polynomiale et  $\mathbf{W}$  la matrice diagonale des poids, on peut écrire le problème des moindres carrés pondérés comme :

$$\mathbf{B}_p^\top \mathbf{W} \mathbf{B}_p \theta = \mathbf{B}_p^\top \mathbf{W} \mathbf{y}$$

avec  $\theta$  les coefficients polynomiaux associés à la base polynomiale  $\mathbf{B}_p$ . Ensuite, on calcule les résidus, recalcule la valeur des poids à partir des résidus puis on répète l'opération jusqu'à convergence. La convergence peut être en fonction de la valeur des résidus, la stabilité des coefficients calculés ou bien un nombre d'itérations.

Dans la structure du code, nous avons besoin des éléments suivants : une base polynomiale et une classe de M-estimateurs asymétriques qui fait l'interpolation par la méthode de moindres carrés pondérés itératifs.

Pour la création de la base polynomiale, dans notre cas précis, nous avons besoin d'une base avec un degré 1. Nous avons fait le choix de créer une classe qui peut générer n'importe quel degré de base polynomiale car cette classe pourrait servir à d'autres endroits dans le code de l'application. Pour cela, on lui donne le vecteur des  $x$  ainsi que le degré du polynôme choisi. Chaque colonne de la matrice est alors un degré de la base polynomiale.

Pour l'interpolation par M-estimateur asymétrique, nous avons besoin des fonctions de poids. Cette classe dispose d'une matrice de base, dans notre cas nous lui fournissons la matrice de base polynomiale, d'un vecteur de valeurs à interpoler, des valeurs pour la fonction de poids et des critères de convergence. On a ensuite une fonction qui applique l'estimation des coefficients et qui les retourne.

Toute cette partie est générique, c'est-à-dire qu'elle n'est pas limitée au redressement AFM mais peut être utilisée n'importe où ailleurs. Nous avons ici un module d'interpolation par M-estimateurs asymétriques.

**Estimation des coefficients pour l'image** Nous arrivons à la partie spécifique. L'objectif est l'extraction des coefficients dans l'image. Pour cela, on parcourt les lignes de l'image et applique pour chaque ligne l'interpolation par M-estimateur asymétrique construite précédemment. Chaque ligne est interpolée deux fois, avec les deux types d'asymétrie et on conserve les coefficient de chaque ligne. Cette étape est coûteuse en temps de calcul car on applique beaucoup d'opérations. Si l'image est de taille  $2048 \times 2048$  alors on applique 4096 interpolations d'un vecteur de taille 2048. Afin d'optimiser le code, on a appliqué une parallélisation sur l'extraction des coefficients en coupant l'image en blocs et en extrayant les coefficients de chaque bloc dans un processus différent. On peut ainsi gagner un facteur 4 à 8 sur un ordinateur classique.

Une fois les deux listes de coefficients obtenues, on peut alors conserver uniquement celle qui a la variance minimum.

**Application du redressement** La première étape est le lissage des coefficients. Pour cela, nous avons une fonction qui calcule les sauts puis une fonction qui applique le lissage en tenant compte des sauts. Enfin, en utilisant la matrice de base polynomiale et la liste des coefficients, on crée une image avec chaque ligne la droite estimée. On soustrait ensuite à l'image AFM d'origine cette image.

**Intégration et performance** La version de l'algorithme décrite précédemment, fut une première fois développée en `Python`, dans le cadre du développement et des tests de performance. La version `Python` redresse une image de taille  $2048 \times 2048$  en environ 18 secondes, sans optimisation, sur un ordinateur équipé d'un Intel®Core™i7-6700HQ et 8Go de ram.

L'algorithme fut ensuite développé en `C++` pour intégration dans le produit. Nous avons utilisé la librairie `Eigen`<sup>1</sup> pour les opérations d'algèbre

---

1. <http://eigen.tuxfamily.org>

linéaire. La version en C++ redresse une image AFM de taille  $2048 \times 2048$  en un temps proche de la seconde avec la même configuration de matériel que celle décrite précédemment. L'algorithme fut ensuite intégré dans le logiciel Platypus<sup>TM</sup> de POLLEN METROLOGY. Une capture d'écran est présentée dans la Figure 4.1. L'avantage pour l'utilisateur final, est que l'algorithme prend uniquement en entrée une image AFM. Il détecte ensuite automatiquement le bon redressement à appliquer et le réalise.

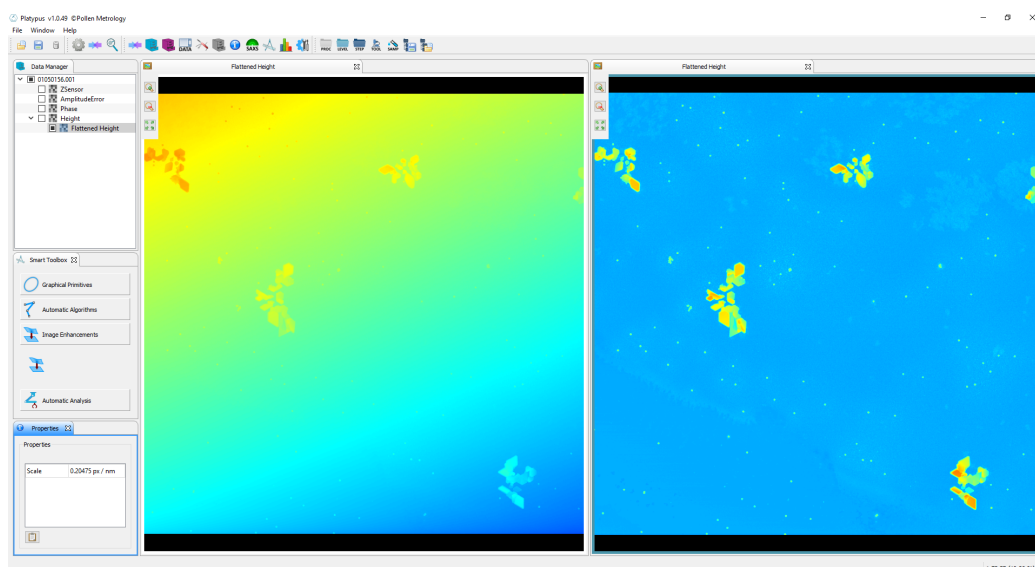


FIGURE 4.1 – Exemple d’une capture d’écran du logiciel Platypus<sup>TM</sup> de POLLEN METROLOGY présentant un résultat d’un redressement AFM avec l’image originale à gauche et le résultat du redressement à droite

## 4.3 Algorithme d’analyses de nanoparticules

### 4.3.1 Analyse d’images MEB et AFM

Les algorithmes d’analyse d’images MEB et AFM de nanoparticules fonctionnent tous les deux sous le même format. On prend l’image en paramètre, on sépare les objets du fond de l’image puis on analyse les particules qui sont présentes dans l’image.

Nous avons alors une classe qui permet de prendre une image et de construire une liste particules.

La première étape est de passer d’une image en niveaux de gris à une liste de composantes connexes. On a alors développé une classe d’extrac-



### 4.3. ALGORITHME D'ANALYSES DE NANOPARTICULES

tion de composantes de connexes de l'image en suivant une interface. Une composante connexe est représentée par une liste de pixels.

Pour chaque composante connexe, on souhaite en extraire de l'information. Pour cela, on a créé une collection d'extracteurs qui suivent tous la même interface. Ainsi un extracteur prend en paramètre une composante connexe et retourne une grandeur. Par exemple le diamètre, l'aire, la circularité, l'ellipticité et bien d'autres. En utilisant les extracteurs, on passe d'une liste de composantes connexes à une liste de particules avec leurs propriétés.

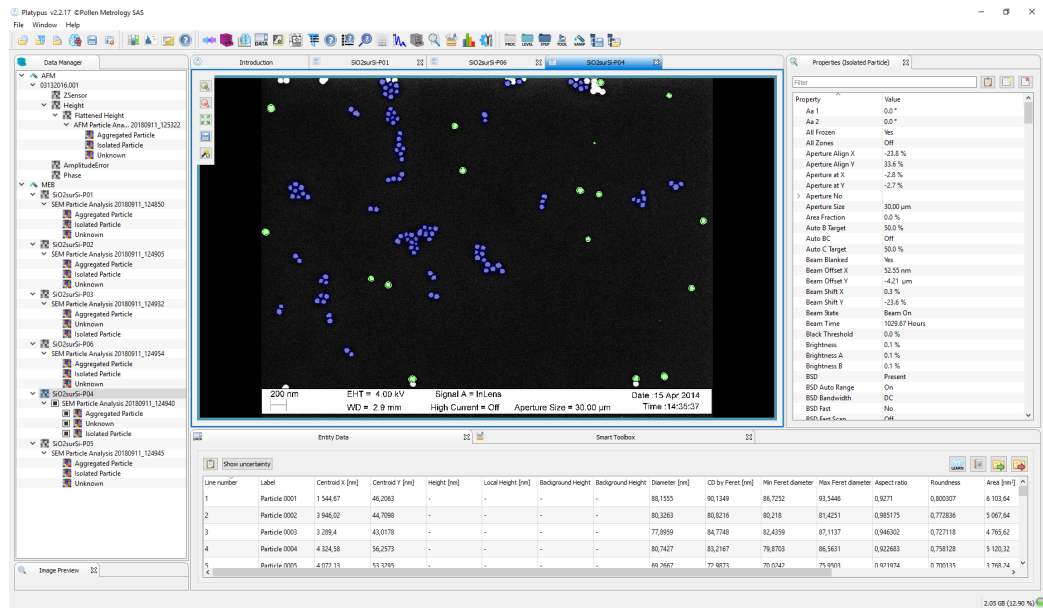


FIGURE 4.1 – Capture d'écran du logiciel Platypus<sup>TM</sup> de POLLEN METROLOGY présentant un résultat d'analyse d'une image de nanoparticules dans une image MEB. La partie centrale présente l'image avec le résultat de l'extraction. La table en bas du logiciel affiche les valeurs des dimensions extraites ainsi que des classifications. La partie de gauche contient le gestionnaire de données ouvertes dans le logiciel. La partie de droite contient les informations sur le nœud sélectionné dans le gestionnaire de données.

La Figure 4.1 présente une capture d'écran d'un résultat d'analyse d'une image MEB de nanoparticules. Dans les images, nous avons besoin de ressortir une information à l'utilisateur sur ce qui a été détecté et mesuré. Afin que l'utilisateur puisse observer les particules détectées et mesurées, on a ajouté des enrichissements graphiques sur l'image avec une couleur par classification des particules, en fonction de si la particule est isolée ou au sein d'un agrégat. Le temps d'analyse d'une image MEB de taille  $2000 \times 1400$  est d'environ 1,5

secondes.

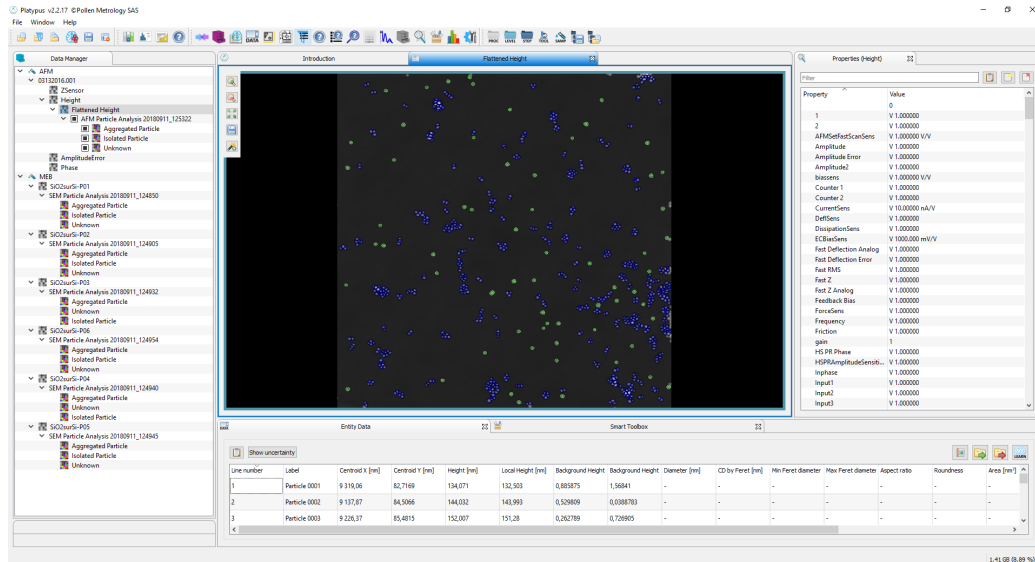


FIGURE 4.2 – Capture d’écran du logiciel Platypus<sup>TM</sup> de POLLEN METROLOGY présentant un résultat d’analyse d’une image de nanoparticules dans une image AFM. La partie centrale présente l’image avec le résultat de l’extraction. La table en bas du logiciel affiche les valeurs des dimensions extraites ainsi que des classifications. La partie de gauche contient le gestionnaire de données ouvertes dans le logiciel. La partie de droite contient les informations sur le nœud sélectionné dans le gestionnaire de données.

La Figure 4.2 présente un résultat d’analyse d’image AFM. Les informations retournées à l’utilisateur sont les mêmes que dans le cas de l’analyse MEB. La seule différence réside dans les grandeurs présentes dans le tableau de résultats, qui contient les mesures spécifiques à l’analyse de nanoparticules par AFM.

Les performances d’analyse d’images AFM est de l’ordre de 2 à 3 secondes pour une image de taille  $2048 \times 2048$  en fonction du nombre de particules contenues dans l’image. Rappelons que pour chaque particule, une analyse locale est faite, c’est pourquoi le temps d’analyse dépend du nombre de particules.

### 4.3.2 Agrégation d’estimateurs

L’agrégation d’estimateurs est effectuée en appliquant une combinaison linéaire des estimateurs à agréger. Les poids de la combinaison linéaire sont

obtenus en lien avec la matrice de covariance des estimateurs. La matrice de covariance est calculée en utilisant la méthode de bootstrap. On applique ensuite la combinaison linéaire et en utilisant le bootstrap généré précédemment, on peut calculer l'intervalle de confiance de l'estimateur agrégé par basic ou percentile bootstrap.

La modélisation de l'agrégation d'estimateurs fut assez complexe. Le diagramme UML de la Figure 4.3 présente le résultat de cette modélisation. Nous allons ici le détailler.

Nous avons un objet `Population`, qui représente une population de valeurs. Sur cet objet, on peut concaténer des populations et on peut calculer un `operator` qui correspond aux les estimateurs à appliquer à la population, comme la moyenne, la médiane, l'écart-type et la variance. La fonction `GetStatisticalValue` est la fonction qui applique un des `operator` et calcule son intervalle de confiance par bootstrap. L'intervalle de confiance est donné avec le pourcentage donné en paramètre tout comme le nombre d'échantillons pour le bootstrap.

Le bootstrap est fait de la façon suivante, la classe `Population` utilise la classe `Bootstrapper` pour calculer une carte de bootstrap en utilisant la fonction `GenerateMap`. En effet, on a choisi de générer des cartes de bootstrap, ce qui revient à des matrices d'index plutôt que de retourner directement les échantillons bootstrappés dans le but de pouvoir appliquer la même carte à des populations qui seraient appariées. Avec la carte générée, on peut appliquer la carte en utilisant la fonction `ApplyMap`

La classe `Population` utilise donc le bootstrap via la classe `Bootstrapper` pour calculer les différents `operator`. Arrivé à ce stade, nous disposons d'un outil qui permet d'avoir sur une population les principaux descripteurs statistiques avec leurs intervalles de confiance. La classe `ExtendedPopulation` est une classe qui est une surcouche pour cacher les éléments de bootstrap et d'intervalle de confiance. Cette classe permet d'avoir une population de calculer un opérateur dessus et d'avoir son intervalle de confiance. Dans cette classe, on fixe le pourcentage pour l'intervalle de confiance et le nombre d'échantillons pour le bootstrap.

L'agrégation utilise également le bootstrap, pour calculer la matrice de covariance des estimateurs. Pour cela, nous avons conçu les classes `Estimator` et `ExtendedEstimator` afin de manipuler des estimateurs et non des populations pour l'agrégation d'estimateurs. `ExtendedEstimator` est à `Estimator` ce que `ExtendedPopulation` est à `Population`.

Ainsi, on fournit les estimateurs que l'on souhaite agréger à la classe `AggregationEstimator`. La fonction `ComputeLamba` calcule les coefficients de la combinaison linéaire à appliquer aux estimateurs. On peut ensuite appliquer cette combinaison linéaire pour obtenir la valeur de l'estimateur agrégé

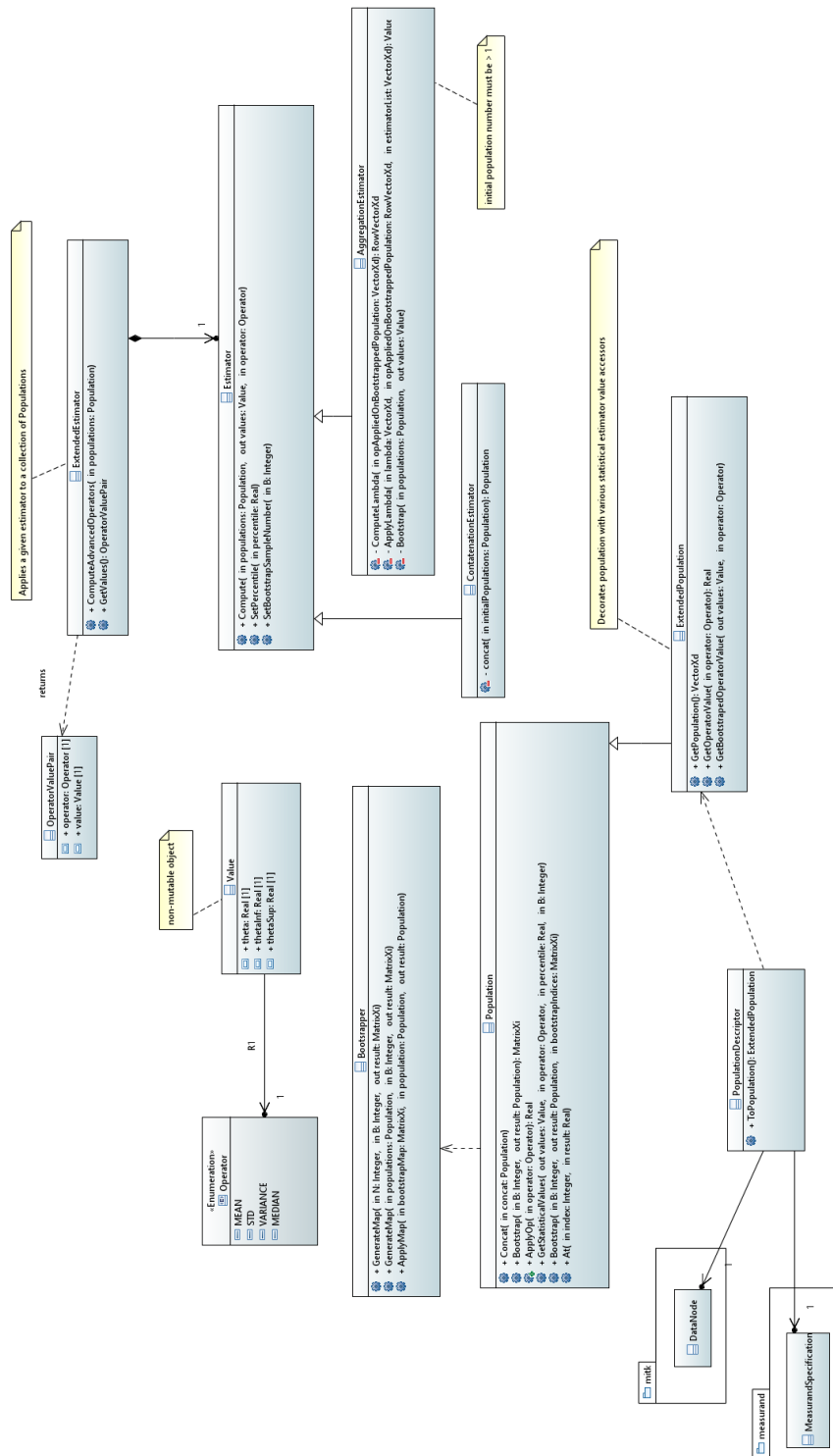


FIGURE 4.3 – Diagramme UML de la structure du code pour l'agrégation d'estimateurs.

### 4.3. ALGORITHME D'ANALYSES DE NANOPARTICULES

avec l'intervalle de confiance associé via la fonction `ApplyLambda`. L'intervalle de confiance est calculé selon la méthode `basic bootstrap`.

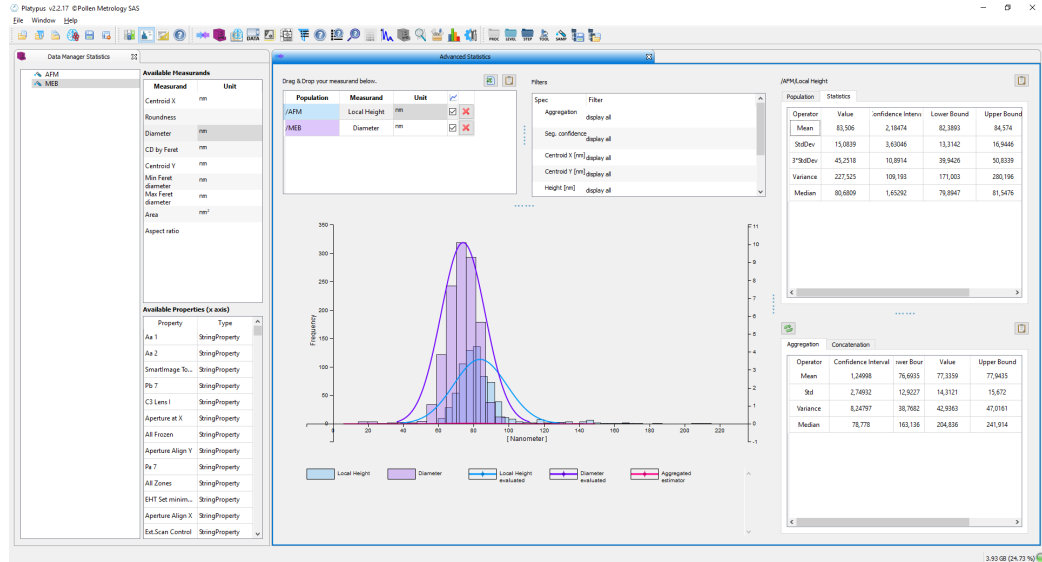


FIGURE 4.4 – Capture d'écran du Platypus<sup>TM</sup> de POLLEN METROLOGY présentant un résultat d'agrégation d'estimateurs sur des populations de diamètres de nanoparticules AFM et MEB. La table en bas à droite contient les résultats de l'agrégation d'estimateurs. La table en haut à gauche les valeurs statistiques classiques sur la population sélectionnée dans la table centrale. La table centrale contient une partie avec les populations et les options graphiques et à droite les différentes caractéristiques permettant de filtrer les populations. À gauche, le gestionnaire de données permettant de naviguer dans les résultats pour sélectionner les populations.

La Figure 4.4 représente la perspective statistique du logiciel Platypus<sup>TM</sup>. Cette perspective, permet de sélectionner les populations et de filtrer les valeurs en fonction des caractéristiques de la population. Elle permet également d'afficher les histogrammes des populations ainsi que d'avoir les valeurs statistiques classiques sur les populations tel que la moyenne, l'écart-type et la médiane. Elle permet également de faire de l'agrégation d'estimateurs lorsque l'utilisateur le souhaite. Le logiciel applique alors une agrégation d'estimateurs sur les populations sélectionnées. Les performances sont liées au nombre et à la taille des populations sélectionnées. Pour agréger deux populations de moins de 500 éléments, l'agrégation prend moins d'une seconde pour calculer l'agrégation d'estimateur pour 3 estimateurs classiques qui sont médiane, moyenne et variance. L'intervalle de confiance est estimé par percentile boots-

trap. En comparaison lors du prototypage en Python de l'agrégation d'estimateurs, le temps de calcul pour l'obtention des valeurs de la table 2.8 prenait environ 5 secondes pour estimer les intervalles de confiance par basic et percentile bootstrap et a pris près de 8h en parallèle sur 8 cœurs pour le calcul de l'intervalle de confiance par bootstrap studentisé. Rappelons que les populations faisaient toutes moins de 150 éléments et qu'il y avait 6 différents estimateurs tandis que le nombre d'échantillons bootstrap était de 10000.

## 4.4 Analyse d'images lamellaire

L'intégration du lamellaire est faite en suivant la même logique que précédemment. L'algorithme d'analyse d'images de lamellaire est fait selon les étapes suivante :

1. Calcul de la période
2. Calcul de l'orientation
3. Pré-traitement
4. Binarisation
5. Calcul du squelette
6. Calcul des défauts
7. Calcul des dimensions des lignes

Pour répondre à ce pipeline, nous avons créé une classe qui calcule la période sur une image, afin de pouvoir la réutiliser dans d'autres algorithmes. Nous avons fait de même pour le calcul de l'orientation. Les pré-traitements et la binarisation sont dans une fonction car ils utilisent des outils fournis par une librairie de traitement d'images. Pour le squelette, nous avons créé une classe `Skeleton`. Cette classe prend une image binaire, applique un algorithme de squelettisation et dispose d'une méthode qui analyse le squelette pour l'extraction des défauts.

Pour l'extraction des grandeurs, on a une classe qui représente les lignes, qui prend l'image binaire, le squelette et les défauts et qui contient les fonctions d'extraction des grandeurs utiles.

Au final, nous avons un algorithme automatique, qui ne prend aucun argument pour fonctionner. Un exemple d'analyse d'image de lamellaire dans le logiciel Platypus<sup>TM</sup> de POLLEN METROLOGY est présenté dans la Figure 4.1. Dans l'application, l'analyse d'une image MEB de lamellaire de taille 512 × 512 a un temps d'exécution en moyenne proche de la seconde.

#### 4.4. ANALYSE D'IMAGES LAMELLAIRE

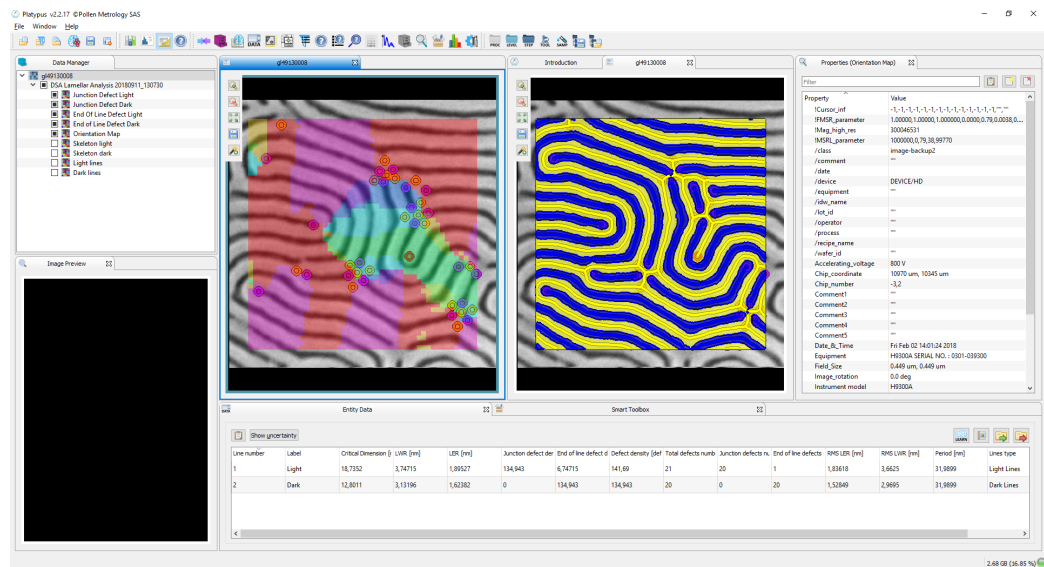


FIGURE 4.1 – Exemple d’une capture d’écran du logiciel Platypus™ de POLLEN METROLOGY présentant un résultat d’une analyse d’image MEB de lamellaire. À gauche du logiciel, le gestionnaire de données, au centre les résultats visuels, avec la carte d’orientation sur l’image de gauche et la segmentation sur l’image de droite. Le bas du logiciel présente une table de résultat et à droite des images, les propriétés du nœud sélectionné dans le gestionnaire de données.

# Bibliographie

Langr, J. (2013). *Modern C++ Programming with Test-Driven Development : Code Better, Sleep Better*. Pragmatic Bookshelf.

Roques, P. (2011). *UML 2 par la pratique*. Eyrolles.



