

**Modélisation Statistique de Transformées Multirésolutions et
Application à la Détection de Défauts de Textures**

par

Marouene Mejri

Mémoire présenté au Département d'Informatique et d'Ingénierie en vue de
l'obtention du grade de Maître ès sciences (M.Sc.)

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS
Département d'Informatique et d'Ingénierie

Gatineau, Québec, Canada, mai 2013

Remerciements

Au Prof. *Mohand Saïd Allili* pour sa grande disponibilité, sa patience et son aide dans le cadre de ce travail de recherche et de rédaction.

À la Prof. *Nadia Baaziz* pour sa disponibilité, ses efforts et son encadrement précieux dans le cadre de ce travail de recherche et de rédaction.

Aux Profs. *Ana-Maria Cretu* et *Larbi Talbi* pour avoir accepté de lire et d'analyser ce mémoire.

À mes collègues *Ouiza*, *Nadjate* et *Idrissa* qui m'ont souvent aidé à travailler dans une bonne ambiance.

À mes parents, mes sœurs et mon frère qui m'ont soutenu et encouragé tout au long de ma carrière académique.

À mes amis *Mehdi* et *Ahmed* en souvenir de nos éclats de rire et des bons moments.

Résumé

La détection automatisée de défauts de textiles est un problème important en industrie. Dans le passé, plusieurs travaux ont été proposés dans ce but, à l'aide d'algorithmes basés sur le traitement d'images. En effet, au cours de la production de tissus, un dispositif spécialisé de capture d'images enregistre des images du tissu pour fin de traitement par ordinateur. Le but est d'extraire les caractéristiques de la texture du tissu pour détecter des défauts contenus à l'aide de techniques de traitement d'images. Jusqu'à ce jour, il n'existe pas de méthode standard qui assure la détection de défauts de texture dans les tissus avec haute précision.

Dans ce mémoire, nous présentons une nouvelle méthode de détection de défauts de texture dans les tissus de textile. Cette méthode se base sur la modélisation statistique utilisant les mélanges finis de Gaussiennes généralisées (MoGG) des sous-bandes de transformées multirésolutions (Ondelettes, Contourlettes) d'une image. Cette modélisation permet d'obtenir des signatures compactes et précises des structures de textures. La comparaison de textures est ainsi effectuée par une mesure de similarité entre MoGGs grâce à une approximation de la divergence de Kullback-Leibler (KL). Notre méthode est capable d'atteindre une détection et localisation très précises de défauts de texture dans les images de la base de données TILDA, tout en assurant un temps de traitement raisonnable.

Table des matières

Remerciements	ii
Résumé	iii
Table des matières	iv
Liste des tableaux	viii
Liste des figures	ix
Liste des acronymes	xii
Chapitre 1 — Introduction générale	1
Chapitre 2 — État de l’art sur la détection de défauts dans les textiles	6
2.1 Introduction	6
2.2 Industrie du textile et défis	7
2.3 Inspection traditionnelle de tissu	10

2.4	Inspection automatique de tissu	11
2.4.1	Les approches statistiques	13
2.4.2	Les approches spectrales	15
2.4.3	Les approches basées sur la modélisation	16
2.5	Critères d'évaluation des algorithmes de détection de défauts de texture .	17
2.6	Sommaire du chapitre	19
Chapitre 3 — Étude des transformées multirésolutions		20
3.1	Généralités	20
3.2	La transformée en ondelettes	22
3.3	La transformée en contourlettes	24
3.3.1	La pyramide Laplacienne	25
3.3.2	Le filtrage bidimensionnel directionnel	27
3.4	La transformée en contourlettes redondantes	31
3.5	Sommaire du chapitre	34
Chapitre 4 — Analyse en Composantes Indépendantes (ACI) appliquée à la détection de défauts de texture		35
4.1	Introduction	35
4.2	Analyse en Composantes Indépendantes	36
4.3	Détection des défauts de texture de tissus basée sur l'ACI	38
4.4	Application sur un ensemble d'images	40

4.5	Critiques	42
4.6	Sommaire du chapitre	44
Chapitre 5 — Modélisation avec les mélanges finis de Gaussiennes généralisées		46
5.1	Introduction	46
5.2	Le modèle de mélanges finis de Gaussiennes généralisées	48
5.2.1	Avantages	48
5.2.2	Estimation de paramètres	49
5.2.3	Calcul de similarité entre MoGGs	55
5.3	Application à la détection de défauts de texture dans les textiles	56
5.3.1	Phase d'apprentissage	56
5.3.2	Phase de détection	57
5.4	Sommaire du chapitre	67
Chapitre 6 — Expérimentations		68
6.1	Introduction	68
6.2	Outils de travail	69
6.3	Méthode d'évaluation	70
6.4	Intégration des différentes transformées multirésolutions et classification des divergences de KL	73
6.4.1	Seuillage des divergences de KL avec l'algorithme des K-moyennes	75

6.4.2	Seuillage des divergences de KL avec l'algorithme d'Otsu	76
6.4.3	Seuillage des divergences de KL avec l'algorithme d'Otsu médian .	77
6.4.4	Classification des divergences de KL avec le classificateur de Bayes	78
6.5	Influence de la taille des blocs	85
6.6	Traitement par chevauchement de blocs	89
6.7	Comparaison avec la méthode de détection basée sur l'ACI	90
6.8	Analyse de la complexité algorithmique	95
6.9	Sommaire du chapitre	96
Chapitre 7 — Conclusion générale		97
Bibliographie		100

Liste des tableaux

6.1	Table de vérité de l'opérateur logique <i>XOR</i>	72
6.2	Classification des divergences de KL avec l'algorithme des K-moyennes.	75
6.3	Classification des divergences de KL avec l'algorithme d'Otsu.	76
6.4	Classification des divergences de KL avec l'algorithme d'Otsu médian.	77
6.5	Statistiques de taux d'erreurs lors de la phase d'apprentissage des classificateurs de Bayes avec différents modèles statistiques.	81
6.6	Statistiques de taux de détection basée sur l'ACI.	91
6.7	Statistiques de taux de détection basée sur les contourlettes redondantes et le modèle MoGG.	92

Liste des figures

2.1	Système de tricotage industriel.	7
2.2	Architecture d'un système typique d'inspection automatique de tissu [42].	12
2.3	Classification des matériaux texturés et des approches de détection de défauts de texture [42].	14
3.1	Schéma de décomposition récursive en ondelettes et l'obtention des quatre sous-bandes.	23
3.2	Exemple d'application de la transformée en ondelettes à 3 niveaux sur une image naturelle.	24
3.3	Schéma de décomposition en pyramide Laplacienne.	27
3.4	Représentation du fonctionnement des ondelettes et des contourlettes pour les images [51].	28
3.5	Représentation fréquentielle de la décomposition en contourlettes ayant 4 niveaux de résolution ($L = 4$).	29
3.6	Exemple d'application de la transformée en contourlettes à 2 niveaux.	30
3.7	Schéma de décomposition en contourlettes redondantes.	32
3.8	Exemple d'application de la transformée en contourlettes redondantes.	33

4.1	Exemple d'une texture d'un tissu de textile.	36
4.2	Diagramme du système de détection de défauts basé sur l'ACI [57].	40
4.3	Comparaison des résultats obtenus.	41
4.4	Proposé du nouveau système de détection des défauts de textiles basé sur l'ACI.	44
5.1	Exemple de distribution platykurtique et leptokurtique.	49
5.2	Image de texture extraite de Vistex [70].	53
5.3	Histogrammes des sous-bandes contourlettes de l'image Fabric et leurs modélisations MoGG.	54
5.4	Histogramme de l'image d'approximation de l'image Fabric et sa modélisation MoGG.	55
5.5	La modélisation par les mélanges finis des Gaussiennes généralisées des sous-bandes de la transformée en ondelettes d'une image exempte de défauts.	59
5.6	La modélisation par les mélanges finis des Gaussiennes généralisées des sous-bandes de la transformée en ondelettes d'une image contenant un défaut pointu.	60
5.7	L'algorithme de détection de défauts basé sur la modélisation MoGG des sous-bandes des transformées multirésolutions.	61
5.8	Exemple de détection de défauts.	62
5.9	La modélisation MoGG des sous-bandes ondelettes du bloc(1,1) de l'image présentée dans la Figure 5.8.	63
5.10	La modélisation MoGG des sous-bandes contourlettes du bloc(1,1) de l'image présentée dans la Figure 5.8.	64

5.11	La modélisation MoGG des sous-bandes contourlettes redondantes du bloc(1,1) de l'image présentée dans la Figure 5.8.	65
5.12	La modélisation MoGG des sous-bandes ondelettes du bloc(3,3) de l'image présentée dans la Figure 5.8.	66
6.1	Arborescence de la base de données TILDA.	70
6.2	Exemple de construction de la vérité terrain d'une image contenant un défaut.	71
6.3	Nombre d'erreurs des différents classificateurs de Bayes versus le Nombre d'itérations lors de la phase d'apprentissage.	82
6.4	Modélisations des divergences de KL correspondantes au Tableau 6.5.	84
6.5	Exemples de détection de défauts avec la méthode de classification par seuillage de K-moyennes.	86
6.6	Exemples de détection de défauts avec la méthode de classification par seuillage d'Otsu.	87
6.7	Exemples de détection de défauts avec la méthode de classification par seuillage d'Otsu médian.	88
6.8	Schéma de traitement d'image par chevauchement de blocs.	89
6.9	Exemple de traitement d'image avec et sans chevauchement de blocs de taille 64×64 pixels.	90
6.10	Les 12 images défectueuses et mal détectées par l'algorithme basé sur la transformée en contourlettes redondantes et la modélisation par les mélanges finis de Gaussiennes généralisées.	93
6.11	Comparaison de la détection de défauts de notre méthode avec la détection de défauts de la méthode de référence.	94

Liste des acronymes

ACI : Analyse en Composantes Indépendantes.

CT : Contourlet Transform. Transformée en Contourlettes.

DWT : Discrete Wavelet Transform. Transformée en Ondelettes discrètes.

KL : Kullback-Leibler

MoGG : Mixture of Generalized Gaussian. Mélanges finis de Gaussiennes Généralisées.

PL : Pyramide Laplacienne.

PLR : Pyramide Laplacienne Redondante.

RCT : Redundant Contourlet Transform. Transformée en Contourlettes Redondantes.

Chapitre 1

Introduction générale

L'image est la représentation visuelle du monde réel perçu soit par l'œil soit par un matériel d'acquisition d'image. L'image est un thème important dans la vie humaine surtout avec le progrès technologique que nous vivons aujourd'hui. En effet, l'amélioration du matériel d'acquisition d'image vient dans le cadre de l'amélioration de la qualité d'image que nous souhaitons avoir et justifie en outre l'importance de l'image dans notre quotidien. De plus, l'importance de l'image se manifeste aussi dans l'intégration des capteurs d'images dans plusieurs périphériques utilisés de façon quotidienne tels que les webcams intégrés dans les ordinateurs portables, les caméras dans les cellulaires, etc.

L'image est utilisée dans différents domaines tels que la médecine, la robotique, la télévision, la vidéoconférence, la vidéosurveillance, le Web et d'autres. Ce besoin important de l'image dans le quotidien a introduit de nouvelles problématiques parmi lesquelles nous pouvons citer la compression d'images, la recherche d'images, la segmentation, l'extraction d'objets, l'organisation et l'indexation de l'information sur le Web, etc.

Toutes ces problématiques se basent essentiellement sur le traitement d'images par ordinateur ou plus généralement la notion de l'apprentissage par machine. En effet, lorsqu'il

s'agit d'un grand volume de données, l'opération de recherche d'images ou de comparaison d'images est difficile, voire impossible, pour l'être humain, d'où le besoin de confier cette opération à une machine capable de stocker ce volume de données et le traiter plus rapidement. Par ailleurs, dans certains domaines d'applications, la machine est celle qui prend la décision. A titre d'exemple, dans la robotique, un robot doit apprendre comment réagir suite à la détection d'un objet dans son environnement.

Parmi les analyses que nous pouvons effectuer sur l'image, nous citons l'analyse de la texture. La texture est une propriété importante caractérisant les régions d'une image. Malgré qu'il n'y ait pas une définition exacte de la texture, nous pouvons tout de même admettre qu'elle représente la mesure de certaines propriétés d'une région telles que la finesse, la granularité, la régularité et l'homogénéité [19, 34]. Pour analyser les textures d'une image, nous aurons recours dans nos travaux à l'analyse multirésolution. L'analyse multirésolution est l'un des outils les plus utilisés dans l'analyse et le traitement d'images numériques du fait qu'elle permet d'analyser le signal dans différents niveaux de résolutions [31]. Le recours à cette analyse se justifie par la difficulté de l'interprétation de l'information contenue dans l'image directement à partir des niveaux de luminance des pixels [15]. Dans ses travaux [44], Mallat a mis en évidence l'importance des variations d'intensités pour l'analyse de l'image. De plus, les structures que nous désirons analyser dans une image peuvent avoir différentes tailles. Par conséquent, nous avons besoins de plusieurs résolutions afin d'analyser et interpréter ces structures. L'analyse multirésolution est capable de le faire grâce au fait qu'elle décompose l'image dans plusieurs échelles de résolutions et d'orientations différentes. Ces propriétés ont rendu l'analyse multirésolution un outil puissant pour faire la compression, la segmentation, le codage, la transmission de l'image, etc.

Parmi les domaines sur lesquels nous pouvons appliquer ces outils, nous citons la détection des défauts de texture dans les tissus de textile. Le textile représente l'un des domaines

les plus actifs de notre quotidien. L'industrie canadienne du textile compte plus de 400 entreprises, concentrées principalement au Québec et en Ontario [69]. La production textile s'élevait à quelque 6,2 milliards de dollars en 2005. Le secteur emploie plus de 40 000 personnes, et l'industrie versait en 2003 plus de 1,6 milliard de dollars en salaires [69]. Dans certains autres pays puissants en industrie du textile, les chiffres sont plus remarquables. A titre d'exemple, l'industrie chinoise de textile a réalisé des exportations d'une valeur de 46 milliards de dollars au premier semestre de l'année 2011 [68].

Ces statistiques montrent la valeur ajoutée ainsi que l'importance de ce secteur dans l'économie nationale et mondiale. Cependant les gains du secteur du textile dépendent du facteur de qualité du produit. En effet, en 1992, une étude [60] a estimé la réduction du prix du produit textile à 45%-65%. Cette réduction est essentiellement due à la présence de défauts, d'où la nécessité de la mise en place d'un processus de contrôle automatisé de la qualité.

L'automatisation de ce processus est exigée par le coût élevé de l'inspection et le contrôle humain ainsi que la quantité gigantesque du produit à inspecter. Cependant, ce but reste jusqu'à nos jours un défi pour les chercheurs. La complexité de ce problème découle essentiellement de la variation stochastique de l'échelle, de l'élasticité et de l'inclinaison du tissu ainsi que de l'environnement et la nature du processus de tissage [41]. De plus, la comparaison image à image est non efficace pour plusieurs raisons ; si les images n'admettent pas exactement la même taille et la même résolution spatiale (en dpi), la comparaison devient impossible. Dans le cas de translation ou de rotation, la comparaison image à image ne donne pas les résultats souhaités. Une des solutions possibles consiste à extraire une signature compacte des images pour pouvoir les comparer.

Dans notre travail, nous présentons une nouvelle méthode de détection de défauts qui se base sur la modélisation statistique des sous-bandes des transformées multirésolutions d'une image et la comparaison des modèles obtenus. Cette méthode découpe une image

de textile en blocs de même taille. Nous appliquons sur chaque bloc la transformée multirésolution souhaitée pour récupérer les différentes sous-bandes multirésolutions. Nous procédons ensuite à la modélisation statistique des sous-bandes multirésolutions via le modèle de mélanges finis de Gaussiennes généralisées (MoGG). Notre méthode calcule la divergence de Kullback-Leibler (KL) entre les modèles estimés d'un bloc test et les modèles estimés d'un bloc exempt de défaut fixé comme référence. Selon la mesure de la divergence obtenue, la méthode étiquète le bloc test ; "exempt de défauts" ou "contenant des défauts".

Les objectifs de notre travail se résument dans les points suivants :

1. Produire une nouvelle méthode de détection de défauts dans les textures dans les tissus de textile basée sur les transformées multirésolutions et la modélisation par les mélanges finis de Gaussiennes généralisée.
2. Comparer différentes transformées multirésolutions présentées dans ce mémoire et ressortir celle la plus appropriée à la détection de défauts de texture.
3. Implémenter différentes méthodes de classification et les appliquer dans la classification des distances de Kullback-Leibler (KL), comparer le rendement de ces méthodes et ressortir celle la plus appropriée à la détection de défauts de texture.
4. Comparer la nouvelle méthode de détection de défauts de textures dans les tissus de textile avec les méthodes de l'état de l'art.

Dans ce mémoire, nous présentons tout d'abord un état de l'art sur la détection de défauts dans les textures de tissus. Ensuite, nous présentons les transformées multirésolutions les plus populaires. Le quatrième chapitre dans ce mémoire présente une méthode de détection de défauts basée sur l'analyse en composantes indépendantes (ACI). Notre nouvelle méthode de détection de défauts, basée sur la modélisation par les mélanges finis de Gaussiennes généralisées (MoGG) appliquée aux sous-bandes des transformées multirésolutions, est détaillée dans le cinquième chapitre. Nous présentons les expérimentations

et l'évaluation de notre méthode dans le sixième chapitre. Nous finissons ce mémoire par une conclusion générale et les travaux futurs.

Chapitre 2

État de l'art sur la détection de défauts dans les textiles

2.1 Introduction

Le domaine du textile est très actif et met sur le marché une grande variété de produits qui servent comme matière première dans différentes industries. Le facteur essentiel qui contrôle ce domaine est la qualité du produit final qui est le tissu, ou plus précisément *la trame de tissu*. Généralement, dans un milieu industriel, une trame de tissu est évaluée selon la présence de défauts et leurs localisations [42]. Ce critère résume en effet la définition du facteur de qualité dans ce domaine. Dans ce chapitre, nous présentons en première section le milieu industriel de production de tissu ainsi que les défis rencontrés dans son inspection. Nous exposons dans la seconde section l'inspection traditionnelle. Dans une troisième section, nous détaillons l'inspection automatisée. Nous discutons dans une quatrième section les critères d'évaluation des algorithmes de détection de défauts de texture selon l'état de l'art disponible.

2.2 Industrie du textile et défis

Dans un système industriel, le tricotage de trame de tissu est totalement automatisé. Le fil enroulé est déroulé par un moteur pour éviter sa coupure. Un chariot est responsable de pousser une ou plusieurs aiguilles dans un sens déterminé pour former un nœud. La succession de cette opération produit une ligne de nœud dite aussi une ligne de trame de tissu. Selon la nature de trame à produire, le nombre de fils et le nombre des aiguilles peut varier. Dans certains autres systèmes de tricotage, il y a un système de filature intégré qui est responsable de la production du fil avant de le passer au système de tricotage [50].

Figure 2.1 présente un schéma simplifié d'un système de tricotage industriel. Cette figure présente uniquement le processus de tricotage et suppose que le fil est déjà produit.

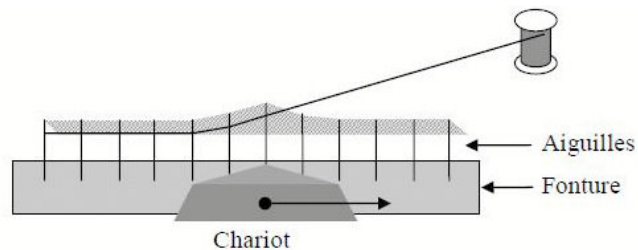


Figure 2.1 – Système de tricotage industriel.

Durant le processus de tricotage d'une trame de tissu, plusieurs types de défauts peuvent se produire. Le fil utilisé peut être coupé résultant alors en un vide dans la trame, une tâche d'huile peut être causée par le matériel de tricotage ou une différence de couleur peut être marquée durant la teinture du tissu. Ci-après, nous présentons les défauts les plus fréquents dans la fabrication de tissu [39].

1. **Extrémités cassées** : Si un fil de chaîne est absent dans le tissu pour une distance courte ou longue.

2. **Médiateur brisé** : Si un fil de trame est absent dans le tissu pour une largeur courte ou longue.

Il est à noter que le fil de chaîne est celui qui est tendu sur le métier à tisser. Le fil de trame étant celui passé alternativement au-dessus et au-dessous des fils de chaîne.
3. **Flotteurs** : Il s'agit d'un type de défaut où un fil de chaîne ou de trame flotte au-dessus de la surface du tissu. Ce défaut est la conséquence de l'absence d'entrelacement de deux séries de fils.
4. **Trame frisée** : Si les fils de chaînes ou de trames sont très torsadés ou détendus.
5. **Limaces** : Lorsque le fil de trame est impur et contient des limaces ou des irrégularités de diamètre.
6. **Couture** : Les extrémités et les pics ne sont pas entrelacés selon l'ordre correct de la structure.
7. **Densité irrégulière** : Si la densité de sélection qui est prise par pouce varie en raison de panne mécanique.
8. **Tissu poilu** : Les fibres dans le fil ont été rendus rugueuses avant, pendant ou après le tissage.
9. **Trous** : Présence des petits trous dans le tissu.
10. **Taches d'huile** : Taches causées par la lubrification des pièces mécaniques de la machine de tricotage.
11. **Défaut de point de départ** : Causé par l'arrêt et le départ d'un métier à tisser.
12. **L'excrétion** : Lorsque un rôle de tissu est teint, la profondeur de teinte de la couleur peut varier d'un endroit à un autre sur la longueur du tissu est connu comme l'excrétion.

13. **Tache blanche** : Si le tissu gris est fait de mélange de coton et de polyester ou d'autres fibres synthétiques à une quantité négligeable, après la teinture de la fibre, le polyester blanc reste, résultant ainsi en des points blancs dans le tissu.
14. **Défaut de teinture** : Durant le processus de teinture, des bandes de couleur plus dense que la couleur supposée, peuvent apparaître.
15. **Tache colorée** : Durant le processus de teinture, des taches colorées peuvent apparaître à cause des nœuds et des fibres impropres dans le tissu.

La diversité des défauts de textile rend la tâche d'inspection plus dure. En effet, la question qui se pose est : quand exactement considérer qu'il s'agit d'un défaut ? De plus, un défaut peut être apparent dans une petite échelle, en d'autres termes lorsqu'il s'agit d'une vision locale alors que dans une grande échelle, ou fenêtre plus large, le même défaut n'est plus visible. L'exemple le plus connu de ce genre de problème est le trou (7ème défaut défini ci-dessus). Selon la taille du trou, ce défaut peut être détecté localement mais pas dans une grande échelle. Le cas inverse aussi constitue un défi à son tour ; un défaut peut-être remarquable sur une grande échelle alors qu'il ne se distingue pas localement. L'exemple le plus fréquent est celui des trames frisées (2ème défaut défini ci-dessus). En regardant la trame dans une grande échelle, nous pouvons remarquer le défaut de texture et nous pouvons remarquer sans doute qu'il s'agit d'un défaut, alors qu'en regardant localement ce défaut n'est plus remarquable.

Le problème que nous essayons de résoudre est la détection automatique de défauts et il ne s'agit en aucun cas de classifier les défauts détectés bien que cette classification est utile. L'être humain est capable de distinguer les défauts dans les deux échelles (grande échelle et petite échelle) grâce à son système de vision complexe. Le but de notre travail est de proposer une méthode automatique qui permet de discriminer les zones défectueuses des zones saines dans les images de tissu. Cette méthode sera appliquée par des systèmes ordonnés et doit assurer une meilleure inspection avec le minimum d'erreurs.

2.3 Inspection traditionnelle de tissu

Selon la littérature [42], il existe deux méthodes d'inspection de textile. La première consiste à inspecter le tissu en cours de production. En d'autres termes, lorsque les machines de tricotage sont entrain de produire les trames, un inspecteur (humain) est chargé de regarder la trame et signaler la présence d'un défaut. Cette inspection est dite préventive puisqu'elle tend à limiter la propagation du défaut. Elle exige le contrôle total sur les paramètres de production. Cette méthode, bien qu'elle semble optimale pour le contrôle de qualité du tissu, est difficile à réaliser pour diverses raisons. Tout d'abord, le processus de tricotage est trop compliqué pour l'arrêter et le redémarrer de façon fluide. De plus la production est assez lente pour garder l'humain attentif à la détection de défauts ; la vitesse de production qui est de 0.3 à 0.5 mètre/minute rend une telle activité ennuyante et ne garantit pas la concentration et l'occupation efficace de l'inspecteur [38]. Le milieu industriel hostile ne joue pas en faveur de cette méthode puisque la chaleur et le bruit produits par les machines de tricotage sont insupportables pour l'inspecteur [38].

La seconde méthode s'applique sur le produit final. Il s'agit donc d'une inspection post-production. Elle consiste à valider la qualité du produit final pour déterminer sa catégorie de qualité et prendre ainsi les décisions appropriées telles que le prix, l'acheteur, etc. Elle est plus facile à appliquer puisqu'il n'y aura pas le problème d'arrêt et de redémarrage de la chaine de production et peut être appliquée dans une phase à part. Les étapes d'inspection post-production sont les suivantes [30] :

1. Tirer les rouleaux de tissu des machines.
2. Dérouler les rouleaux sur une tablette d'inspection. Cette tablette est conçue particulièrement pour cette tâche. Elle est spécialement éclairée et fonctionne à une vitesse relativement haute (généralement de 8 à 20 mètres/minute).
3. Appliquer l'inspection humaine. L'inspecteur ne fait que tourner et arrêter le moteur

de la tablette selon la présence d'un défaut pour noter sa nature et sa localisation.

4. Le résultat de cette inspection se transforme en statistiques qui sont analysées afin de prendre les mesures nécessaires pour réduire les défauts, en se basant sur la nature, l'occurrence et la régularité du défaut. Une réduction du prix de la trame peut être appliquée par l'usine.

Il est à noter que ces statistiques servent aussi à prendre des décisions qui concernent les machines de tricotage, telles que le changement de certaines pièces dans ces machines, appliquer des entretiens supplémentaires ou la fin de vie.

L'inspection traditionnelle est dure à réaliser, non seulement pour des raisons techniques, mais aussi pour des raisons économiques. Une étude menée par Nickolay et Schmalfuà [45] montre que l'investissement dans un système d'inspection automatique de tissu est économiquement intéressant lorsque la réduction des coûts de personnels et les avantages associés sont considérés.

2.4 Inspection automatique de tissu

L'inspection automatique de tissu est basée sur le traitement par ordinateur. Le seul moyen qui permet cette inspection est l'image. En effet, un système d'inspection automatique de tissu prend des images de la trame en cours de production et effectue le traitement sur ces images afin de détecter les anomalies. Figure 2.2 montre l'architecture typique d'un système d'inspection automatique de tissu.

Les caméras installées en parallèle permettent d'acquérir des images de la trame en cours de production. Leur nombre varie selon la nature du tissu à produire et d'autres spécifications techniques des caméras. L'aspect de parallélisme des caméras permet généralement une acquisition plus riche et plus rapide. La largeur de la trame d'un tissu est généra-

lement entre 1.5 et 2 mètres [38]. Une seule caméra permet alors d'obtenir une image à une échelle réduite et certains petits défauts peuvent échapper à la détection. Le système d'illumination situé au-dessous de l'enrouleur de tissu permet une meilleure condition d'acquisition d'images. En effet, le paramètre de lumière est très important. Un excès de lumière ou un manque de lumière donne une image mal éclairée ce qui peut induire l'ordinateur en erreur, le cas échéant, une fausse alerte d'un défaut de teinture (14ème défaut présenté dans la section 1) peut être déclenchée. Le système d'illumination est manipulé par un contrôleur spécifique afin de paramétrer l'illumination du système selon la nature du tissu ou l'illumination globale de l'usine.

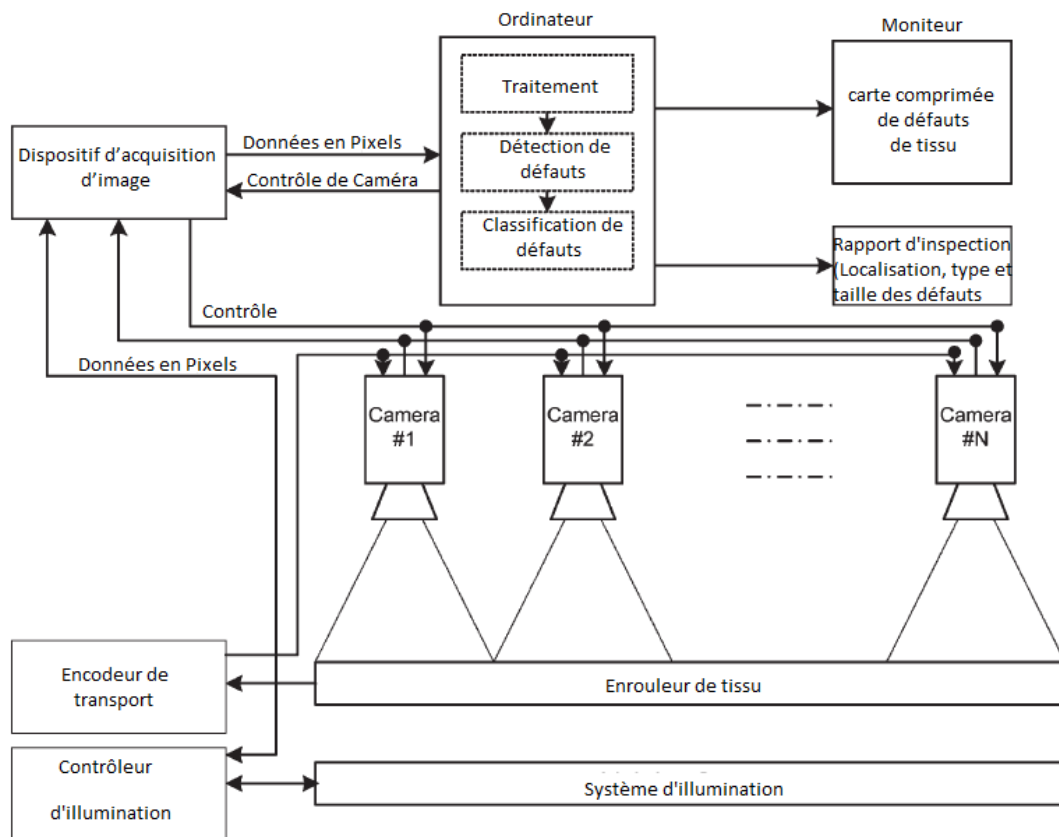


Figure 2.2 – Architecture d'un système typique d'inspection automatique de tissu [42].

Une fois les images sont acquises, elles sont transférées à un dispositif d'acquisition d'images qui permet de rassembler les données sous le format pixel pour obtenir la totalité de l'image dans une bonne résolution et avec la richesse souhaitée.

L'ordinateur effectue le traitement nécessaire sur ces images afin de détecter les défauts et dans un cas optimal, les classer selon leurs natures, leurs types et leurs tailles. Les produits de cette tâche sont la carte de défauts qui indique les coordonnées géométriques des défauts présents dans le tissu et le rapport de statistiques des défauts qui sera analysé par la direction pour prendre les décisions nécessaires.

Le tissu est considéré comme un matériau texturé. Alors s'il s'agit d'un défaut dans un tissu, il devrait absolument être un défaut de texture. Dans la littérature [42], il existe 3 classes de matériaux texturés, à savoir, des matériaux texturés uniformément, des matériaux texturés aléatoirement et des matériaux texturés par motif (pattern). Figure 2.3 présente des exemples des différentes classes de textures. Dans nos travaux, nous nous concentrons sur les tissus uniformément texturés mais l'approche peut être facilement généralisée pour traiter les cas des matériaux texturés par motif.

Figure 2.3 montre les différentes classes de texture dans les matériaux texturés. Selon l'état de l'art de détection de défauts de texture, pour traiter le problème de détection automatisée de défauts dans les tissus uniformément texturés, il existe trois types d'approches, tel que montré dans la même figure.

2.4.1 Les approches statistiques

Les approches statistiques sont basées sur une hypothèse commune ; les régions exemptes de défauts admettent des statistiques stationnaires et s'étalent sur une grande région de l'image. De cette hypothèse, les chercheurs tirent la conclusion que toute région qui admet des statistiques différentes de celles des régions exemptes de défauts (que nous avons

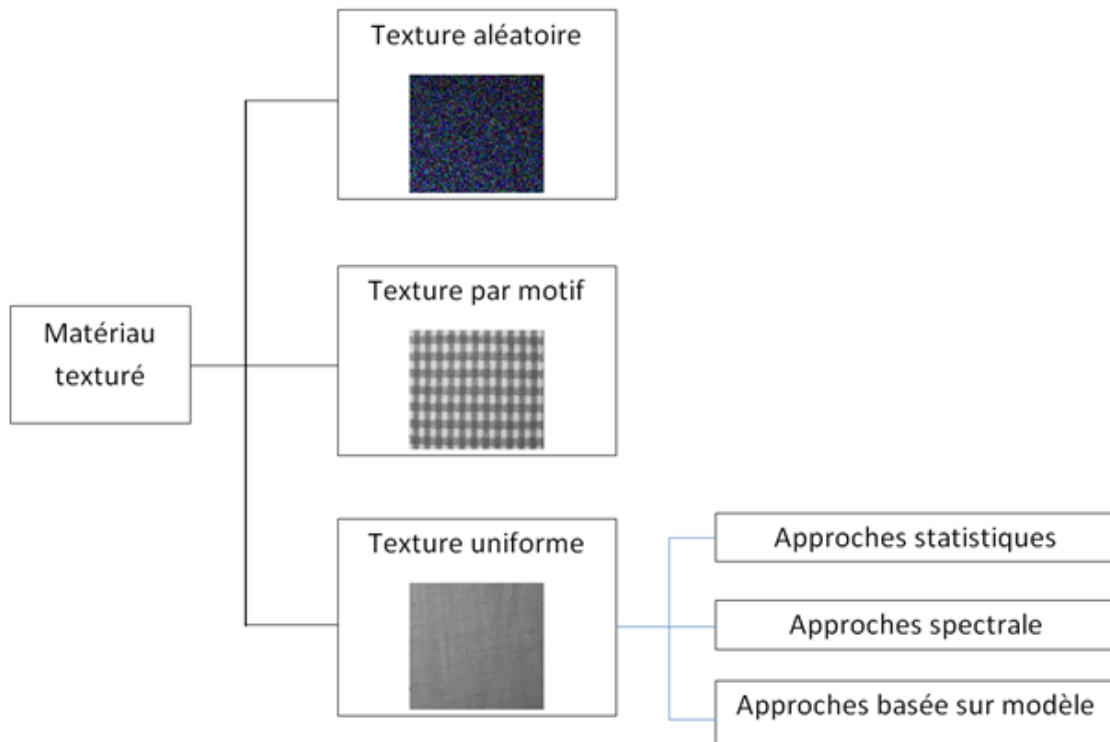


Figure 2.3 – Classification des matériaux texturés et des approches de détection de défauts de texture [42].

appelé des régions saines) est une région porteuse de défauts. Les statistiques utilisées dans ce type d'approches sont multiples telles que l'histogramme de l'image [46, 47, 66], la dimension fractale [22], les statistiques du premier ordre [47, 24], la corrélation croisée [13], la détection de contours [23, 40, 43], les opérations morphologiques [66], la matrice de cooccurrence [24, 59], les filtres propres [61, 1], les fonctions de classement par ordre [26] et les fonctions de transformation linéaire [2]. D'ailleurs, les approches statistiques constituent un grand volume des méthodes utilisées dans le domaine de la détection de défauts de texture [42]. Les résultats atteints par ces approches sont satisfaisants. En contrepartie, la fausse détection est assez élevée. À titre d'illustration, dans [22], les auteurs ont utilisé la dimension fractale pour détecter les défauts de textures. En effet, la

dimension fractale permet de détecter la similarité entre un ensemble de données défini dans un espace de dimensions finis avec un ensemble de ses copies. Cette similarité se mesure par la formule suivante [22] :

$$FD = \frac{\log(N_r)}{\log(\frac{1}{r})}, \quad (2.1)$$

où N_r est le nombre de copies de l'ensemble de données et r est la dimension de réduction.

Cette approche géométrique peut être appliquée sur la classification de textures, voire même la segmentation de textures, du fait que cette dernière représente la répétition d'un ensemble de caractéristiques. Par conséquent, si nous définissons un bloc de l'image de texture comme un bloc sain et nous considérons le reste des blocs de l'image comme étant les copies de ce bloc sain, nous pouvons calculer la dimension fractale de chaque bloc. Ainsi, la mesure de la similarité entre chaque bloc et le bloc sain est effectuée en utilisant l'équation 2.1.

2.4.2 Les approches spectrales

Les approches spectrales sont introduites dans la détection de défauts de textures, suite à l'évaluation des approches statistiques qui ont montré certaines faiblesses. En effet, plusieurs types de défauts peuvent survenir dans une texture sans changer ses caractéristiques statistiques. Les chercheurs ont remarqué une haute périodicité de primitives de texture de base, causée par les fils dans le tissu textile. L'utilisation de cette périodicité permet alors la détection de défauts de texture. Pour ce faire, les chercheurs ont eu recours aux transformées fréquentielles telles que la transformée de Fourier et la transformée en cosinus discrète, et spatio-fréquentielles telles que la transformée en ondelettes discrète [42].

Les approches spectrales constituent à leur tour un grand volet d'applications dans la

littérature et ont produit des résultats satisfaisants dans la détection de défauts. L'analyse en composantes indépendantes (ACI) [8, 56, 57, 58] représente l'une des méthodes qui ont donné de bons résultats dans la détection des défauts dans le textile. En effet, dans [57], les auteurs ont utilisé cette méthode pour discriminer les textures. Cette méthode se repose essentiellement sur la décomposition de la texture finale (un échantillon de la texture saine) en un ensemble fini de textures sources (les textons). Pour mieux expliquer cette méthode, nous en faisons une analogie à une pièce de musique dans laquelle plusieurs instruments sont joués. Le but de cette méthode est de décomposer cette pièce en un ensemble de signaux sources, où chaque signal source représente la musique jouée par un instrument donné, et l'union de tous les signaux sources produit la pièce de musique en question. Cette relation est définie par l'équation suivante :

$$Signal_{final} = \sum_{i=0}^n Signal_{source_i}, \quad (2.2)$$

où n est le nombre de sources qui produisent les différents signaux. Par analogie à l'exemple cité, la texture est formée d'un ensemble de textons. Une fois que les textures sources sont estimées, ces dernières forment une base de textures sources et n'importe quel échantillon de cette texture peut être décomposé sur cette base de textures sources. Ainsi, il devient évident de classer chaque échantillon selon les textures sources exhibées.

2.4.3 Les approches basées sur la modélisation

Les approches basées sur la modélisation ont introduit les modèles probabilistes dans le traitement d'image. La texture est habituellement considérée comme un modèle complexe et peut être définie par un modèle stochastique ou déterministe [42]. Selon Cohen et Fan [20], les textures réelles (telles que la texture des tissus de textile) peuvent être modélisées en tant que procédés stochastiques. Par conséquent, les images texturées sont supposées

être des réalisations (ou des échantillons) provenant des distributions de probabilité sur l'espace des images. L'hypothèse générale de ces approches est la possibilité de résumer la texture dans un modèle statistique. Ce modèle constitue une signature compacte d'une texture donnée. Les zones texturées qui ont la même apparence ont deux modèles similaires. Ainsi, le problème de détection de défauts devient un problème de comparaison de modèles qui est un problème considérablement plus facile à résoudre. En effet, pour comparer deux modèles, plusieurs méthodes ont été utilisées et efficacement implémentées telles que le maximum de vraisemblance et la divergence de Kullback-Leibler (KL) [4]. Une des méthodes les plus utilisées dans la littérature est la modélisation par les champs aléatoires de Gauss-Markov. Dans [25], les auteurs ont utilisé les champs aléatoires de Gauss-Markov afin de modéliser la texture, alors que dans [21], les auteurs ont utilisé ce modèle dans la détection de défauts de texture.

2.5 Critères d'évaluation des algorithmes de détection de défauts de texture

Dans tout algorithme développé, il est nécessaire d'évaluer la méthode élaborée de façon objective. Dans le domaine informatique en général, tout algorithme produit est évalué par des indicateurs qui changent d'un contexte à l'autre. Parmi ces critères, nous pouvons citer la complexité, la précision, l'utilisation optimale des ressources, etc. Dans le domaine d'inspection automatique de textile, il n'existe pas de critères standards d'évaluation des méthodes de détection de défauts de texture. En effet, dans chaque travail publié dans le passé, les auteurs utilisent des critères personnalisés afin d'évaluer leurs méthodes. Bien que ces critères semblent théoriquement viables, leur variété et différentes personnalisations ne permettent pas de comparer ces méthodes ni de les évaluer de façon objective. De plus, la plus part des travaux effectués ont été testés sur différentes bases de données

d'images, ce qui élimine la possibilité de la comparaison visuelle des résultats de différents algorithmes. Le troisième problème est l'expression limitée des résultats d'une méthode. En effet, dans plusieurs travaux (ex. [22]), nous remarquons que l'ensemble de données est limité et il n'y a aucune référence indiquant des expérimentations en annexe papier ou électronique.

Dans ce qui suit, nous allons citer quelques travaux précédents. Dans [22], les auteurs ont atteint un taux de détection de 96%, mais la précision est mauvaise et les fausses alarmes sont très élevées. Nous remarquons donc que le critère qui a été pris en considération concerne uniquement les blocs défectueux. La détection correcte des blocs sains n'était pas prise en considération. Donc, les fausses alertes ont été ignorées alors qu'elles induisent à la perte de l'usine du fait qu'un tissu sain est détecté comme défectueux. Ceci influence profondément la décision concernant une trame de produit final. De plus, l'ensemble de données utilisé dans les travaux est très réduit (16 images dont 8 sans défauts et 8 avec défauts), ce qui est insuffisant pour évaluer cette méthode.

Dans [57], les auteurs ont exprimé des taux de détection de 96%, alors que dans des travaux suivants [56], ont indiqués des taux de 92% . De plus, les critères d'évaluation de leurs algorithmes étaient personnalisés et ont changé d'un article à l'autre. En effet, dans les premiers travaux [57], les auteurs ont défini une certaine précision et l'évaluation était exprimée en termes de pourcentage de ce critère. Dans d'autres travaux [56, 58], ce critère ne figure plus. En contrepartie, dans ces travaux l'évaluation est basée sur le nombre de blocs non détectés correctement dans un ensemble global de blocs d'une image.

2.6 Sommaire du chapitre

Plusieurs travaux dans la détection de défauts de texture ont été proposés durant les deux dernières décennies. Cependant, les articles qui traitent spécifiquement du problème de détection de défauts dans le domaine de textile ne sont pas assez nombreux. Dans les travaux qui existent, l'évaluation des approches n'est pas évidente. Tout d'abord, il n'existe pas de critères d'évaluation en commun entre ces travaux. Certains utilisent la précision de la détection [22, 57], alors que d'autres [56, 58] utilisent le nombre de défauts détectés. Ces évaluations tendent à être subjectives et ne permettent pas aux nouveaux travaux d'être comparés aux anciens. De plus, l'application de ces méthodes à des différents ensembles d'images ne permet pas non plus de les comparer. Malgré cette absence de critères d'évaluation standardisée, nous pouvons remarquer que chaque classe d'approches présente des points forts et des points faibles, ce qui donne l'intuition que la combinaison de plusieurs types d'approches peut donner une méthode hybride capable de détecter les défauts de texture de façon meilleure.

Chapitre 3

Étude des transformées multirésolutions

3.1 Généralités

Dans le domaine du traitement d'image, la notion d'analyse multirésolution apparaît souvent. Elle incorpore et unifie des techniques développées pour différentes disciplines dont le codage en sous-bandes de signaux, les filtres miroirs en quadrature en reconnaissance de la parole et l'extraction de primitives dans les représentations pyramidales d'images [34]. L'interprétation d'une image revient à détecter les primitives visuelles qui la composent. Ces primitives peuvent être entre autres des arêtes, des contours ou des coins et admettent des dimensions variées tels que la longueur, la largeur, la profondeur, l'épaisseur, etc. Pour détecter ces dernières, nous avons besoin de plusieurs opérations de bas niveau [62] comme le filtrage, la dérivée, le sous-échantillonnage, etc.

Durant les dernières décennies, plusieurs chercheurs ont travaillé sur les transformées multirésolutions pour remédier à la limite majeure de la transformée de Fourier (FFT)

qui ne garde que l'information fréquentielle et perd l'information spatiale de l'image. Les transformées multirésolutions permettent, non seulement l'interprétation de l'information fréquentielle, mais aussi sa localisation dans l'image. Les transformées multirésolutions visent essentiellement à assurer les caractéristiques suivantes :

- **La multirésolution** : La représentation devrait permettre aux images d'être successivement approximées, d'une résolution grossière à une résolution fine.
- **La localisation** : Les éléments de base dans la représentation doivent être localisés dans le domaine spatial et fréquentiel.
- **L'échantillonnage critique** : Pour certaines applications (ex. la compression), la représentation devrait constituer une base ou un cadre avec une redondance de petite taille.
- **La directionnalité** : La représentation doit contenir des éléments de base orientés à une variété de directions, beaucoup plus que les quelques directions qui sont offertes par les ondelettes séparables.
- **L'anisotropie** : Pour capturer des contours lisses dans les images, la transformée devrait avoir des éléments de base qui permettent de faire la recherche directionnelle de contours. Ceci est réalisé en utilisant, lors de la décomposition, une variété de formes allongées avec différentes directions.

Dans ce chapitre, nous présentons trois types de transformées qui ont été utilisées fréquemment dans l'analyse de texture, à savoir, la transformée en ondelettes, la transformée en contourlettes et sa variante redondante [4, 6, 67]. Il existe encore plusieurs articles dont les auteurs ont utilisé au moins une de ces trois transformées dans leurs travaux visant essentiellement à analyser et traiter les textures. Le rapport de recherche [10] regroupe l'utilisation de ces trois transformées dans l'extraction de caractéristiques de texture pour la recherche par le contenu.

La première section de ce chapitre présente la transformée en ondelettes discrètes. Nous

introduisons dans la seconde section la pyramide Laplacienne et la transformée en contourlettes. La troisième section présente la transformée en contourlettes redondantes. Nous finirons ce chapitre par une conclusion.

3.2 La transformée en ondelettes

La transformée en ondelettes discrète (DWT) bidimensionnelle permet la représentation d'une image dans le domaine spatio-fréquentiel. Elle est basée sur l'analyse multirésolution qui décompose hiérarchiquement un signal (image) en deux parties. La première partie, de basse fréquence, est une sorte de moyenne du signal d'origine communément appelée image d'approximation. La seconde partie est un ensemble de sous-bandes constituant les détails de l'image ; à chaque niveau de résolution, les détails sont organisés sous forme de 3 sous-bandes orientées *horizontalement* (H), *verticalement* (V) et *diagonalement* (D). Ces détails sont communément appelés les coefficients ondelettes.

À l'origine, les ondelettes ont été introduites par Alfred Haar en 1909. Elles ont été appliquées pour représenter des signaux monodimensionnels. Mallat [44] a étendu l'application de la transformée en ondelettes aux images (ou signaux bidimensionnels). Il a ainsi introduit l'algorithme rapide de décomposition/reconstruction par ondelettes. Cet algorithme est récursif et se base essentiellement sur deux opérations :

- **Le filtrage** : convolution du signal avec un filtre passe-bas (h_0) ou un filtre passe-haut (g_0).
- **Sous-échantillonnage** : réduction du nombre d'échantillons du signal. En effet, un sous-échantillonnage horizontal (1 : 2) de l'image revient à éliminer une colonne sur deux, ce qui réduit le nombre de pixels par ligne à la moitié.

L'algorithme de [44] est schématisé dans Figure 3.1 et s'explique comme suit :

Soit S_j l'image d'approximation à un niveau de résolution j et soit D_j^X la sous-bande d'orientation X , où $X \in \{H, V, D\}$, extraite au niveau de résolution j . Comme le montre la figure, l'image S_j passe en entrée de l'algorithme. Elle subit les deux filtrages *passer-haut* et *passer-bas*. Les deux images résultantes subissent un sous-échantillonnage sur les lignes. Les deux images sous-échantillonnées sont filtrées chacune par un filtre *passer-haut* et un filtre *passer-bas* pour en produire 4 images. Ces dernières sont sous-échantillonnées de nouveau donnant ainsi 4 images de même taille : une image d'approximation (S_{j+1}) et 3 images de détails $D_{j+1}^X, X \in \{H, V, D\}$.

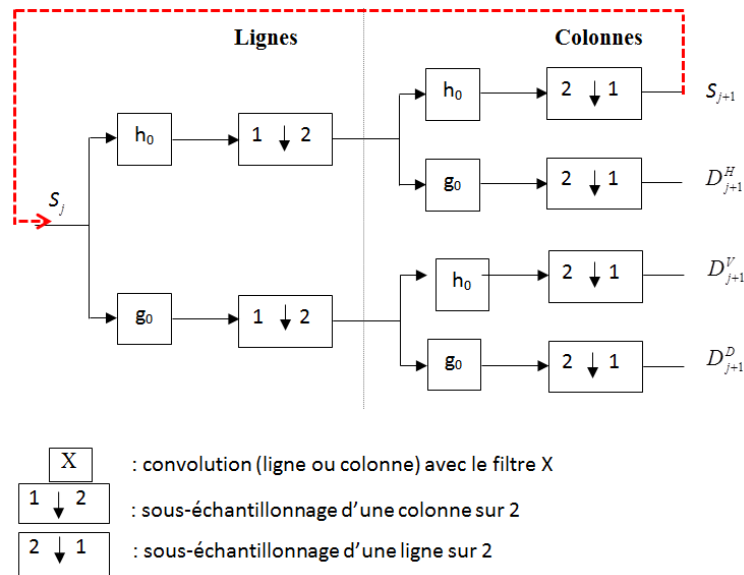


Figure 3.1 – Schéma de décomposition récursive en ondelettes et l'obtention des quatre sous-bandes.

En plus de bénéficier d'un algorithme de décomposition/reconstruction rapide, la transformée en ondelettes offre plusieurs autres avantages ; la représentation d'image en ondelettes est compacte puisque le nombre total des coefficients ondelettes est égal au nombre de pixels de l'image associée. Les coefficients ondelettes sont des coefficients réels contrairement à la transformée de Fourier (FFT) qui fournit des coefficients complexes. Notons aussi que la reconstruction de l'image à partir de sa représentation en ondelettes est par-

faite et ceci démontre que toute l'information contenue dans l'image est conservée dans sa transformée en ondelettes. Nous présentons dans Figure 3.2 un exemple d'application de la transformée en ondelettes à trois niveaux sur une image naturelle.

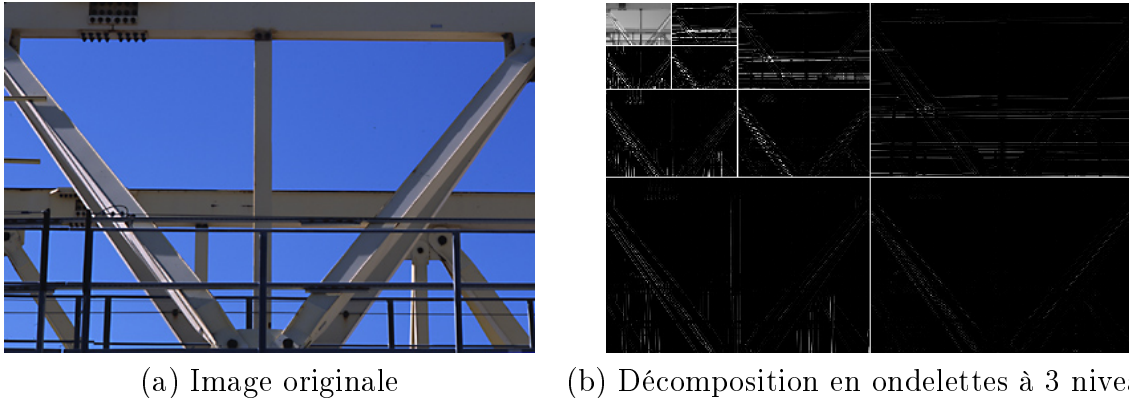


Figure 3.2 – Exemple d'application de la transformée en ondelettes à 3 niveaux sur une image naturelle.

Les ondelettes ont une capacité limitée dans la représentation de l'information directionnelle de l'image, ce qui engendre une faiblesse à la détection des contours lisses. Cette limite provient du nombre restreint d'orientations capturées par les filtres ondelettes lors de la décomposition. En effet, les sous-bandes ondelettes ont soit une orientation horizontale, verticale ou diagonale. De plus, les sous-bandes diagonales des ondelettes sont généralement ignorées dans les traitements basés sur la transformée en ondelettes. L'information contenue dans les sous-bandes diagonales est considérée non pertinente (considérée comme du bruit).

3.3 La transformée en contourlettes

La transformée en contourlettes (CT) a été introduite par Minh N. Do et al. [27, 29] afin de combler la limite majeure de la transformée en ondelettes discrètes bidimensionnelle.

En effet, nous expliquons dans les sous-sections 3.3.1 et 3.3.2 comment la transformée en contourlettes dépasse les performances de la transformée en ondelettes.

La décomposition en contourlettes comporte deux parties distinctes ; la décomposition en pyramide Laplacienne pour créer les niveaux de résolution de l'image et la décomposition en sous-bandes directionnelles par un banc de filtres bidimensionnels directionnels.

3.3.1 La pyramide Laplacienne

La pyramide Laplacienne est introduite en 1983 par Peter J. Burt et Edward H. Adelson [18]. Le but des chercheurs était de trouver une technique permettant de supprimer la forte corrélation entre un pixel et son voisinage. Ceci permet un meilleur codage de l'image puisqu'elle assure l'optimisation du nombre de bits nécessaire pour l'encodage. Cette technique combine les caractéristiques de prédiction et les méthodes de transformation tout en conservant la simplicité et la localité des calculs. La technique de prédiction est relativement simple à implémenter et se considère adaptée aux caractéristiques locales de l'image. La transformée à son tour, permet une meilleure compression de données tout en gardant un temps de calcul raisonnable [18].

Supposons que nous allons calculer la pyramide Laplacienne d'une image $S_0(i, j)$. Pour ce faire, nous appliquons un filtrage passe-bas suivi d'un sous-échantillonnage à cadence 2 selon chaque dimension pour obtenir une image filtrée passe-bas que nous notons $S_I(i, j)$ et dont la taille totale est réduite de 4 par rapport à l'image originale $S_0(i, j)$. L'image d'erreur de prédiction que nous notons $L_0(i, j)$ est obtenue par l'équation suivante :

$$L_0(i, j) = S_0(i, j) - S'_I(i, j) \quad (3.1)$$

L'image $S'_I(i, j)$ est une version interpolée de $S_I(i, j)$ et ayant donc la même taille que l'image $S_0(i, j)$. À son tour, l'image $S_I(i, j)$ est filtrée par un filtre passe-bas et sous-

échantillonnée pour donner une image $S_2(i, j)$. Ainsi, une seconde image d'erreur de prédiction notée $L_1(i, j)$ est calculée en adaptant l'équation 3.1. En répétant ce processus, nous obtenons une séquence d'images d'erreur de prédiction $L_0(i, j), L_1(i, j), L_2(i, j), \dots, L_n(i, j)$ tel que la taille de chaque image est le quart de taille de son prédécesseur. Cette séquence est la pyramide Laplacienne de l'image $S_0(i, j)$.

Au lieu d'encoder l'image $S_0(i, j)$, il est mieux d'encoder $L_0(i, j)$ et $S_1(i, j)$. En effet, $L_0(i, j)$ est largement décorrélée et sa représentation binaire nécessite moins de bits que l'image $S_0(i, j)$. De plus, l'image $S_1(i, j)$ est filtrée en passe-bas et peut être alors encodée à un taux d'échantillonnage réduit. Appliquons ce processus plusieurs fois sur les images passe-bas notées $S_k(i, j), k \in \{0, \dots, n\}$, nous bénéficions des 2 avantages cités ci-dessus plusieurs fois de façon récursive.

Les valeurs des pixels dans chaque image d'erreur de prédiction dans la pyramide correspondent à la différence entre deux fonctions pseudo-Gaussiennes ou connexes, convoluées avec l'image originale. La différence entre ces deux fonctions est similaire à l'opérateur Laplacien couramment utilisé dans l'amélioration de l'image [53], d'où l'appellation Laplacienne de cette pyramide. Dans Figure 3.3 nous montrons le fonctionnement de l'algorithme de la pyramide Laplacienne pour produire l'approximation et l'erreur de prédiction (partie (b)) ainsi que le résultat final de cet algorithme (partie (a)).

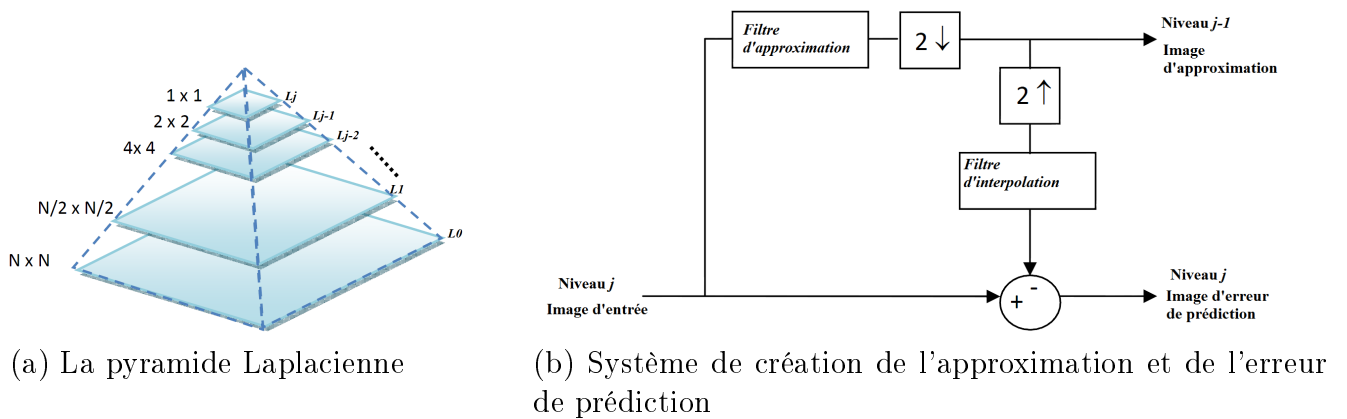


Figure 3.3 – Schéma de décomposition en pyramide Laplacienne.

3.3.2 Le filtrage bidimensionnel directionnel

La transformée en contourlettes permet un nombre différent et flexible de directions à chaque niveau Laplacien, tout en réalisant l'échantillonnage presque critique. Ceci est assuré par l'utilisation d'un banc de filtres bidimensionnels qui ont différentes orientations. Avec un tel ensemble riche de fonctions de base, la transformée en contourlettes peut représenter un contour lisse avec moins de coefficients par rapport à la transformée en ondelettes. De plus, la transformée en contourlettes utilise des bancs de filtres itérés, ce qui rend son algorithme rapide et efficace [51].

Pour mieux expliquer ce que nous avons énoncé, nous présentons dans Figure 3.4 montre un exemple d'application de la transformée en ondelettes et la transformée en contourlettes sur une image, illustrant ainsi le fonctionnement de ces deux transformées. La partie (a) des exemples de cinq images à base d'ondelettes 2-D. La partie (b) montre des exemples de quatre images à base de contourlettes. Dans la partie (c), nous présentons une illustration montrant comment les ondelettes avec des supports carrés ne peuvent pas capturer les points de discontinuités, tandis que les contourlettes ayant des supports allongés peuvent capturer des segments linéaires des contours, et donc peut effectivement

représenter un contour lisse avec moins de coefficients [51].

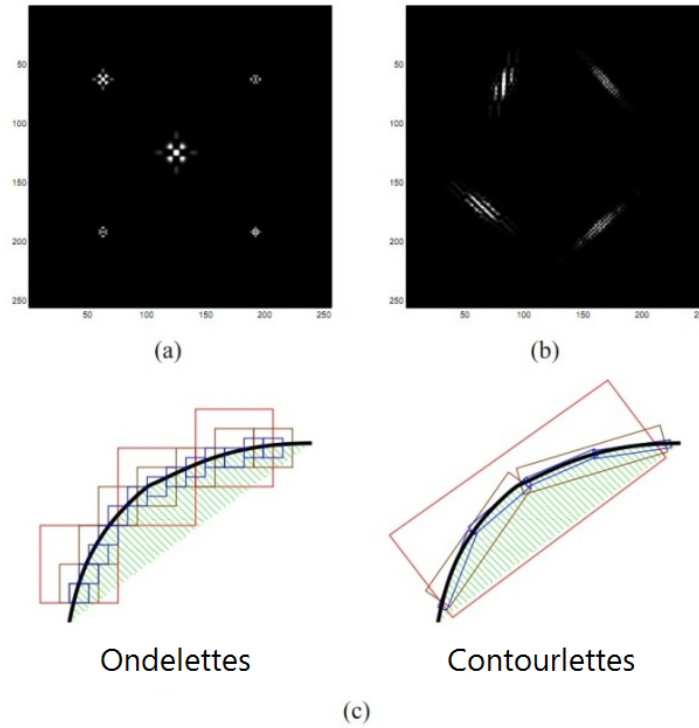


Figure 3.4 – Représentation du fonctionnement des ondelettes et des contourlettes pour les images [51].

Figure 3.5 montre l'aspect de flexibilité du nombre de directions pour chaque niveau de résolution. En effet, la figure en question est une représentation fréquentielle en contourlettes ayant 4 niveaux de résolutions ($L = 4$) et les différentes directions fréquentielles pour chaque niveau de résolution ; le premier et le second niveau de résolution ont 4 directions alors que le troisième et le quatrième niveau de résolution ont 8 directions. L'image d'approximation (basses fréquences) ne subit pas de décomposition directionnelle. Nous remarquons aussi que le nombre de directions est toujours une puissance de 2 ($8 = 2^3$ et $4 = 2^2$).

Dans Figure 3.6, nous montrons l'exécution de l'algorithme de la transformée en contour-

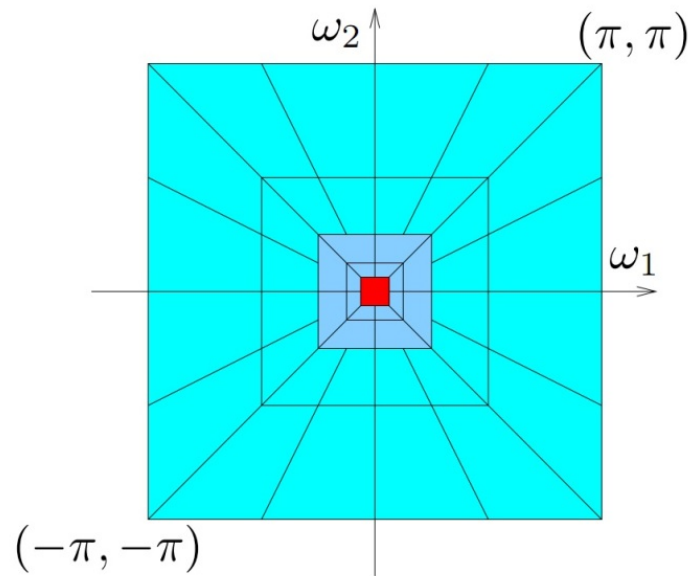
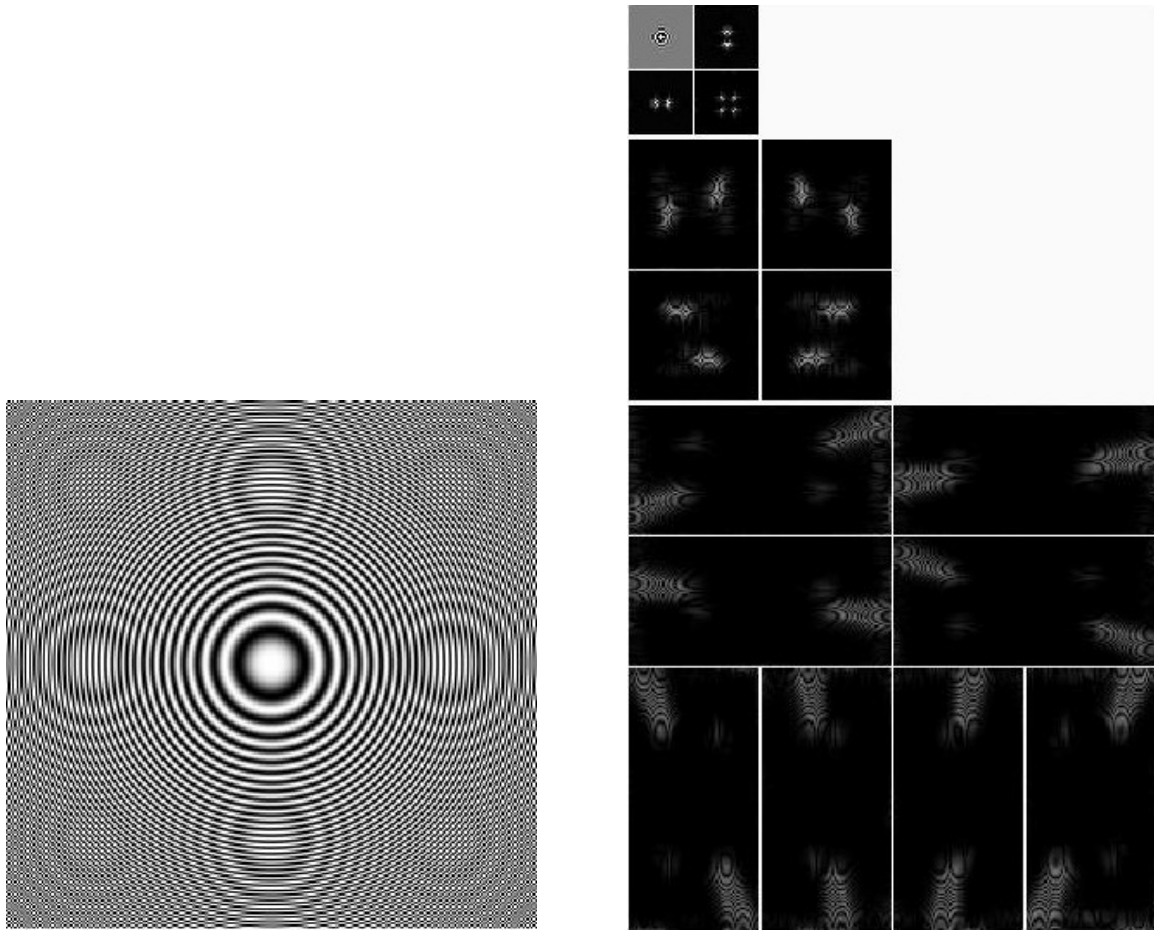


Figure 3.5 – Représentation fréquentielle de la décomposition en contourlettes ayant 4 niveaux de résolution ($L = 4$).

lettes sur une image appelée 'zoneplate'. L'image passée comme paramètre d'entrée de l'algorithme de la transformée en contourlettes est représentée par (a) tandis que (b) représente sa transformée en contourlettes. À première vue, il est clair qu'il s'agit d'une transformée en contourlettes à 2 niveaux ($L = 2$). Nous remarquons aussi que le premier niveau admet 4 directions fréquentielles alors que le deuxième niveau admet 8 directions fréquentielles.

Dans cet exemple, la pyramide Laplacienne a 2 niveaux et 1 approximation. Le banc de filtres contourlettes a subdivisé L_0 en 8 sous-bandes (2 niveaux). L_1 est subdivisée en 4 sous-bandes (1 niveau) et l'image d'approximation est subdivisée en 4 sous-bandes (1 niveau aussi).

Pour illustrer l'aspect de redondance dû à la pyramide Laplacienne nous avons effectué le calcul suivant :



(a) L'image 'zoneplate'

(b) Décomposition en contourlettes à 2 niveaux

Figure 3.6 – Exemple d'application de la transformée en contourlettes à 2 niveaux.

Nombre de pixels de l'image 'zoneplate' :

$$256 * 256 = 65536 \text{ pixels}$$

Nombre de pixels produits par la transformée en contourlettes :

$$(32 * 32) * 4 + (64 * 64) * 4 + (64 * 128) * 8 = 86106 \text{ pixels}$$

$$\text{Taux de redondance} = \frac{\text{nombre de pixels produits par la transformée en contourlettes}}{\text{nombre de pixels de l'image passée en entrée}} \quad (3.2)$$

$$\text{Taux de redondance} = 86106/65536 = 1.3138 \simeq 4/3$$

3.4 La transformée en contourlettes redondantes

Suite à une décomposition d'image en contourlettes, il est clair que les bandes résultantes n'ont pas toutes la même taille aussi bien sur un même niveau de résolution qu'à travers les différents niveaux de résolution.

Lorsqu'il s'agit d'un traitement mettant en coopération les différentes sous-bandes d'une contourlette, il est souhaitable d'avoir des sous-bandes de la même taille afin d'éviter les problèmes d'interpolation et de mise en correspondance entre les différents niveaux de résolution. À partir de cette exigence, une nouvelle variante de la transformée en contourlettes a vu le jour sous le nom de la transformée en contourlettes redondantes [9].

Nous avons indiqué dans la partie 2 que la redondance provient de la pyramide Laplacienne et peut atteindre un taux de sur-échantillonnage jusqu'à 1/3 donnant un espace total de stockage allant à 4/3 de la taille de l'image d'origine. Plus de redondance peut être obtenue en éliminant toute opération de sous-échantillonnage dans le système pyramidal Laplacien ; en utilisant L filtres passe-bas appropriés pour créer L approximations passe-bas de l'image. La différence entre chaque approximation et sa version ultérieure passe-bas est une bande passante de l'image. Le résultat final est un système de Pyramide Laplacienne Redondante (PLR) avec $L+1$ niveaux de tailles égales ; une image d'approximation grossière et L images passe-bande. Chaque image passe-bande est filtrée par un banc de filtres directionnels (BFD). Le filtrage effectué est un filtrage bidimensionnel et

les filtres directionnels sont des filtres pseudo-Gaussiens. Le facteur de redondance est défini par L . En appliquant le même principe sur le niveau de décomposition d-directionnel avec échantillonnage critique sur chaque bande passante PLR, nous obtenons une transformée en contourlettes redondantes avec $L * D$ sous-bandes directionnelles de tailles égales, en plus de l'image d'approximation grossière. Le symbole D représente le nombre de directions souhaité, et chaque sous-bande contourlettes est représentée par une sous-image. Pour mieux comprendre le fonctionnement de la décomposition en contourlettes redondante, nous exposons dans Figure 3.7 le schéma de son algorithme.

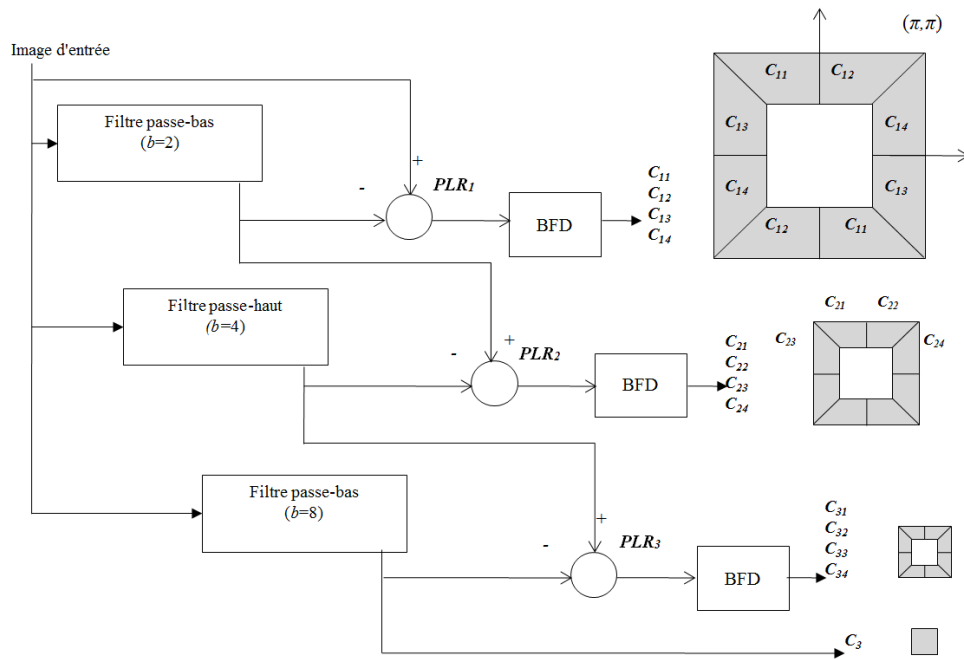
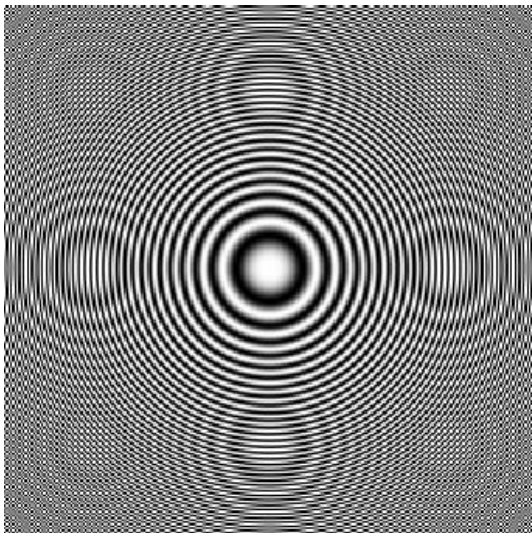
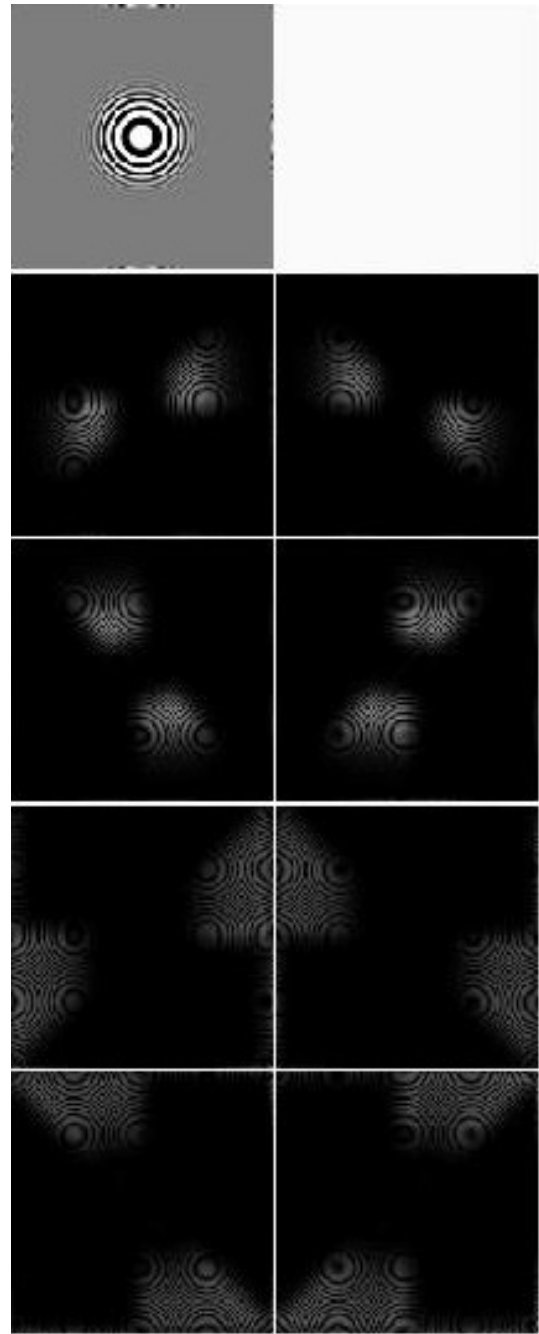


Figure 3.7 – Schéma de décomposition en contourlettes redondantes.

Dans Figure 3.8, nous présentons l'application de la transformée en contourlettes redondantes sur la même image de Figure 3.6. La partie (a) représente l'image passée en entrée de l'algorithme de la transformée en contourlettes redondantes, tandis que la partie(b) représente les sous-bandes de la transformée.



(a) L'image 'zoneplate'



(b) Décomposition en contourlettes redondantes

Figure 3.8 – Exemple d'application de la transformée en contourlettes redondantes.

3.5 Sommaire du chapitre

Dans ce chapitre, nous avons étudié trois types de transformées multirésolution : les ondelettes, les contourlettes et les contourlettes redondantes. Les ondelettes, bien qu'elles soient toujours un outil puissant et largement utilisé dans le traitement et l'analyse d'images, elles admettent des limites relatives à la directionnalité. Les contourlettes viennent dans le but d'améliorer le comportement des ondelettes. Par conséquent, les contourlettes fournissent un bon support pour l'analyse multirésolution et ceci grâce à la flexibilité de choisir la directionnalité des sous-bandes. Ainsi, la détection des contours lisses est meilleure. Nous avons aussi présenté dans ce chapitre la transformée en contourlettes redondantes qui est une variante de la transformée en contourlettes permettant d'avoir des sous-bandes de la même taille, facilitant ainsi le traitement hiérarchique de l'image.

Chapitre 4

Analyse en Composantes Indépendantes (ACI) appliquée à la détection de défauts de texture

4.1 Introduction

L'analyse en composantes indépendantes (ACI) est une méthode d'analyse de données qui fait usage des statistiques, des réseaux de neurones et du traitement du signal, etc [37]. Cette méthode a été initialement introduite pour séparer les différentes sources indépendantes qui composent un signal mixé, de façon aveugle, c'est-à-dire, sans avoir aucune information préliminaire sur les sources. Alors on peut conclure que l'ACI est une méthode d'estimation.

Ci-après, nous présentons dans une première partie les principes fondamentaux de l'ACI. Dans une seconde partie, nous analysons une méthode de détection de défauts de texture basée sur l'ACI et appliquée sur les images texturées de textile [56, 57, 58]. Nous montrons



Figure 4.1 – Exemple d’une texture d’un tissu de textile.

l’application de cette méthode sur notre base de données et les résultats correspondants. Ces résultats serviront ultérieurement comme référence de comparaison. Nous finissons ce chapitre par une conclusion présentant l’apport de cette méthode ainsi que quelques critiques.

4.2 Analyse en Composantes Indépendantes

Comme cité dans l’introduction, l’ACI [8, 37, 56, 57, 58] vise à séparer les sources composant un signal mixé. Citons, à titre d’exemple, un morceau de musique. Nous voulons séparer le son produit par chaque instrument ainsi que le chant produit par le chanteur. L’ACI permet cette séparation sans aucune information préliminaire sur les sources, c’est-à-dire, sans spécifier par exemple les instruments qui contribuent à la production de la pièce de musique ou la plage de fréquence de la voix du chanteur. C’est la raison pour laquelle cette séparation est dite *aveugle*.

Dans le domaine du traitement d’image, le signal à traiter est bidimensionnel. Chaque fenêtre de l’image est considérée comme une combinaison linéaire de composantes qui

sont statistiquement indépendantes [8]. La texture est caractérisée par la répétition d'un certain motif. Cette répétition de motif assure une régularité dans l'image globale. Nous pouvons conclure ainsi que les mêmes composantes qui combinent une fenêtre (bloc) sont les mêmes qui produisent une autre avec des pondérations semblables. Ainsi, pour une image de texture, il suffit de trouver les composantes et les pondérations pour caractériser une image. La formule de l'analyse en composantes indépendante est donnée par [8, 37, 56, 57, 58] :

$$X = A * S, \tag{4.1}$$

où X est le signal d'origine, A est la matrice contenant les composantes indépendantes en vecteurs colonnes et $*$ est l'opération de multiplication de matrices. La combinaison linéaire de ces composantes à l'aide d'un vecteur de coefficients donne un bloc de texture. Dans cette formule, le vecteur de coefficients (ou vecteur de caractéristiques) est noté par S . Notons que dans l'équation 4.1, seul le signal original X est connu a priori. Le rôle de l'algorithme de l'ACI est d'estimer A et S en se basant sur l'observation X constituée d'un ensemble de signaux originaux (ou signaux d'apprentissage). Alors, pour décomposer un signal image selon ses composantes indépendantes, il suffit d'appliquer la formule suivante [8, 37, 56, 57, 58] :

$$S = W * X, \tag{4.2}$$

où S représente la décomposition du signal X dans la base des composantes indépendantes, X est le signal de départ et W est le pseudo-inverse de la matrice A , dite matrice de démixage. Il est clair, mathématiquement, que W est l'inverse de A , c'est-à-dire $W = A^{-1}$. Tout de même, on a utilisé un autre symbole qui est le W . Ceci est dû au fait que dans certains cas, la matrice A n'admet pas d'inverse. Dans ces cas-là, on a

recours à la matrice pseudo-inverse. En résumé, la matrice W peut être l'inverse ou le pseudo-inverse de la matrice A .

4.3 Détection des défauts de texture de tissus basée sur l'ACI

Dans [57, 58], les auteurs ont présenté une méthode de détection de défauts de texture de tissus se basant sur l'ACI. En effet, leur système implanté est composé de deux modules principaux. Le premier consiste en un apprentissage qui s'effectue hors-ligne. L'expression hors ligne veut dire qu'il n'y a pas de contraintes temporelles puisqu'il s'agit d'une préparation de certains paramètres. Le deuxième module est une comparaison entre les paramètres calculés dans ce module et ceux calculés dans le modules d'apprentissage. Le deuxième module est dit 'détection en ligne', qui n'est rien d'autre que la phase de détection des défauts de textile. Nous présentons dans ce qui suit un diagramme résumant le système avec ses deux modules et nous commentons par la suite les détails techniques sous la forme d'un algorithme.

Dans [58], les auteurs ont spécifié un prétraitement à appliquer sur toutes les images avant de procéder à l'estimation des composantes indépendantes et la détection de défauts. Le premier prétraitement consiste à redimensionner toutes les images avec un facteur de 0.5 en appliquant un filtrage d'interpolation bicubique. Le deuxième prétraitement consiste à normaliser les images à une moyenne nulle et variance unitaire.

Algorithm 1 *Algorithme de détection de défauts de texture basé sur l'ACI.*

Phase hors-ligne

1. Sélectionner aléatoirement 10000 blocs d'images exempts de défauts de taille $32 * 32$.
2. Organiser les blocs sélectionnés sous la forme de vecteurs colonnes de la matrice X .
3. Utiliser un algorithme d'estimation ACI pour estimer les composantes indépendantes sous la forme de vecteurs colonnes de la matrice A et calculer la matrice pseudo-inverse W .
4. Choisir une image exempte de défauts et la découper en blocs de taille $32 * 32$ chevauchés de 16 pixels en ligne et 16 pixels en colonnes.
5. Calculer les vecteurs de caractéristiques S_i correspondants aux blocs de l'image en utilisant l'équation 4.2.
6. Calculer le vecteur S_{moyen} en moyennant tous les vecteurs colonnes S_i de la matrice S .

Phase en ligne

1. Choisir l'image de test à traiter et la découper en blocs de taille $32 * 32$ chevauchés de 16 pixels en ligne et 16 pixels en colonnes.
2. Calculer les vecteurs de caractéristiques SI_i de l'image de test en utilisant l'équation 4.2.
3. Calculer les distances Euclidiennes entre les vecteurs colonnes de SI_i et le vecteur S_{moyen} en appliquant la formule :

$$Distance_i = ((S_{moyen} - SI_i)^T * (S_{moyen} - SI_i))^{1/2} \quad (4.3)$$

4. Classifier chaque bloc selon la formule de seuillage suivante : $Bloc_i =$
 $\begin{cases} \text{contenant un défaut,} & Distance_i \geq \alpha. \\ \text{exempt de défauts,} & \text{sinon.} \end{cases}$
tel que :

$$\alpha = D_m + \eta * IQR, \quad (4.4)$$

où D_m est la valeur médiane du vecteur de distance, IQR est l'intervalle interquartile du vecteur de distance et η est une constante déterminée expérimentalement.

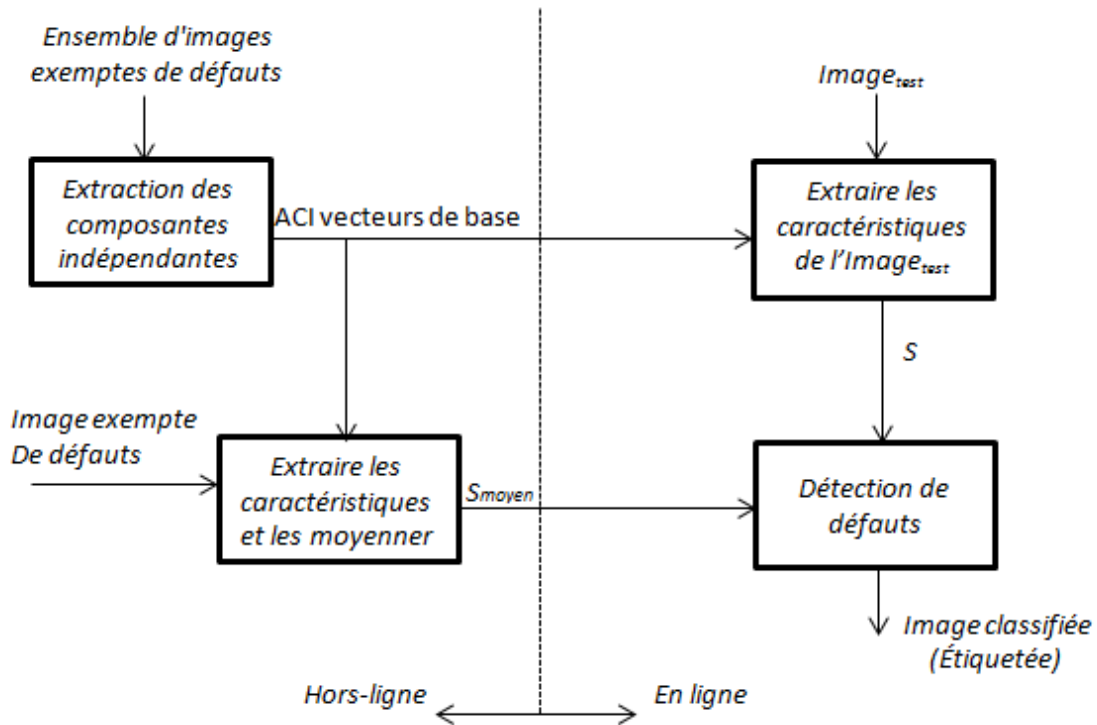
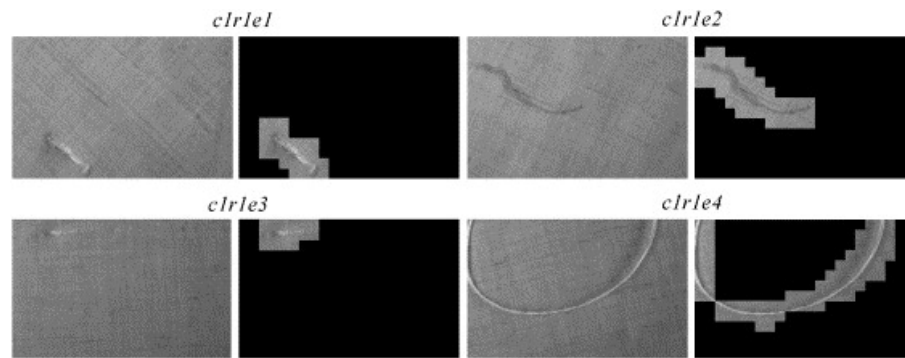


Figure 4.2 – Diagramme du système de détection de défauts basé sur l’ACI [57].

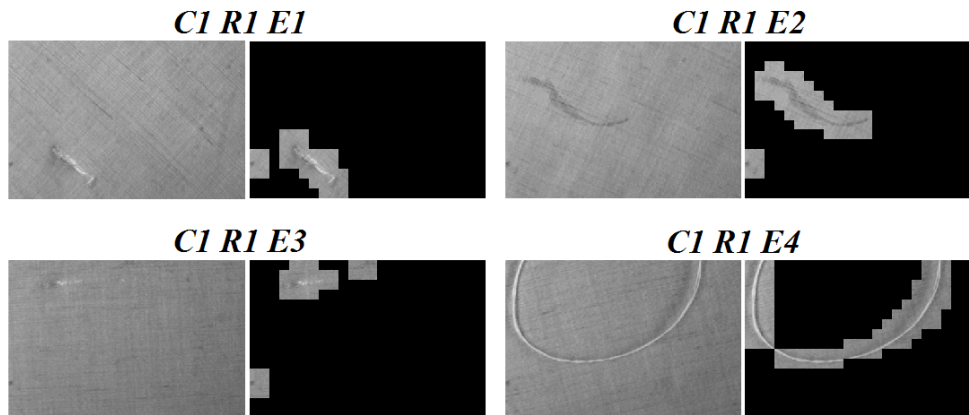
4.4 Application sur un ensemble d’images

Nous avons suivi les travaux présentés dans [58] afin de reproduire les résultats obtenus sur la base de données TILDA [71]. Notons dans ce contexte, que le schéma présenté dans Figure 4.2 est cité dans [57] et a été respecté dans [58] avec quelques modifications au niveau des paramètres tel que le type d’images à introduire dans la phase d’apprentissage. En effet, dans [57] les auteurs ont introduit 19 images représentant des défauts divers et une image exempte de défauts alors que dans [58] les auteurs ont introduit 50 images exemptes de défauts. Ceci est expliqué par le fait que le système ne peut pas apprendre tous les types de défauts (nombre infini de défauts qui peuvent se présenter dans un tissu). Alors, il est plus pratique au système d’apprendre l’absence du défaut dans un tissu. Nous

avons appliqué le même prétraitement cité dans [58] ainsi que les mêmes étapes présentées dans Algorithme 1. Pour se comparer aux résultats présentés dans [58] nous avons choisi des images de test qui ont été présentées dans ce papier, sur lesquels nous avons appliqué notre implémentation de l'algorithme 1. Dans Figure 4.3, nous montrons les résultats obtenus par notre implémentation et les résultats présentés dans [58].



(a) Résultats obtenus dans [58]



(b) Résultats obtenus par notre implémentation

Figure 4.3 – Comparaison des résultats obtenus.

L'interprétation de Figure 4.3 se fait de gauche à droite et par paire d'image. Chaque paire contient l'image originale et l'image après la détection. Dans l'image de détection, le noir représente l'espace exempt de défauts. Nous remarquons ici qu'il y a quelques différences au niveau de la détection. Ceci est dû essentiellement à la différence des images à traiter.

D'ailleurs nous pouvons remarquer la petite translation des images présentée dans [58] surtout dans l'image C1R1E4.

4.5 Critiques

La démarche théorique de cette méthode de détection est basée sur deux phases principales : une phase d'apprentissage et une phase de détection.

La première critique concerne essentiellement le type de l'apprentissage. Dans [57], les auteurs ont introduit quelques types de défauts dans la phase d'apprentissage. Ils ont conclu dans [58] qu'il est impossible de faire une telle démarche du fait que le nombre de combinaisons des défauts qui peuvent se présenter dans un tissu est infini. Pour cela, ils ont décidé que le système doit apprendre l'absence du défaut, du fait que l'absence des défauts représente un seul cas qui est le cas d'un tissu exempt de défauts.

La deuxième critique concerne la décision concernant l'image elle-même. En effet, les auteurs passent directement à la localisation des défauts et ne précisent en aucun cas est-ce que l'image porte déjà un défaut ou elle est exempte de défauts. Ceci constitue un inconvénient majeur dans ce système. Un système de détection de défaut doit analyser une image d'un morceau de tissu afin de décider qu'il contient des défauts ou non, et la classifier en image saine ou en image défectueuse. En plus de ce processus de décision, il est pertinent de préciser, pour les images défectueuses, la localisation de ses défauts. Figure 4.4 présente le schéma d'un système de détection de défauts basée sur l'ACI et contenant le compartiment de classification des images et de localisation des défauts. Le système présenté dans Figure 4.4 répond aux besoins de l'industrie de textile et permet, d'une part, une meilleure gestion de la phase d'inspection et, d'autre part, d'élaborer des statistiques plus avancées pour des utilisations diverses.

La troisième critique porte sur les critères d'évaluation de l'algorithme de détection de défauts basé sur l'ACI. Ces derniers ont varié d'une publication à une autre. En effet, dans [57] et [56] les auteurs ont utilisé un critère d'évaluation qu'ils ont appelé *Taux de détection* qui n'est d'autre que l'*exactitude*. Il est calculé comme suit :

$$Taux\ de\ détection = 100 * (N_{CC} + N_{DD}) / N_{Total}, \quad (4.5)$$

où N_{CC} est le nombre de blocs exempts de défauts qui ont été détectés correctement et N_{DD} représente le nombre de blocs contenant au moins un défaut et qui ont été détectés défectueux. N_{Total} représente le nombre de blocs total dans une image.

Dans [58], les auteurs n'ont pas utilisé ce critère d'évaluation. Ils ont préféré montrer l'aspect visuel des détections réussies, tel que présenté dans la Figure 4.3. Malgré que les résultats exposés montrent l'efficacité de l'algorithme implémenté, il est toujours difficile de comparer cette méthode avec d'autres en absence de statistiques objectives ou de critères quantifiés. Les seules statistiques mentionnées dans [58] concernent le nombre de blocs détectés comme défectueux dans des images saines de la base de données utilisée [71].

La dernière critique concerne la taille des images et le prétraitement effectués sur ces images. Bien que la normalisation des images garantie l'élimination des problèmes d'illumination lors de l'acquisition, ceci peut, par la même occasion, réduire la visibilité de certains défauts de textiles causés par l'intensité de la coloration. La détection de tels défauts devient alors impossible. De plus, l'opération de redimensionnement de l'image avec un filtrage bicubique suivi d'un sous-échantillonnage nous fait perdre de l'information puisque l'image obtenue après un tel prétraitement constitue une image de basse fréquence et ne conserve plus les détails fins de hautes fréquences qui peuvent être porteurs d'informations utiles pour une meilleure détection.

Malgré toutes les critiques présentées dans cette section, nous pouvons dire que la méthode de détection de défauts basée sur l'ACI représente une méthode puissante, rapide et efficace et nécessite une révision au niveau de l'implémentation afin d'améliorer sa détection. Ceci peut être le sujet de recherche future.

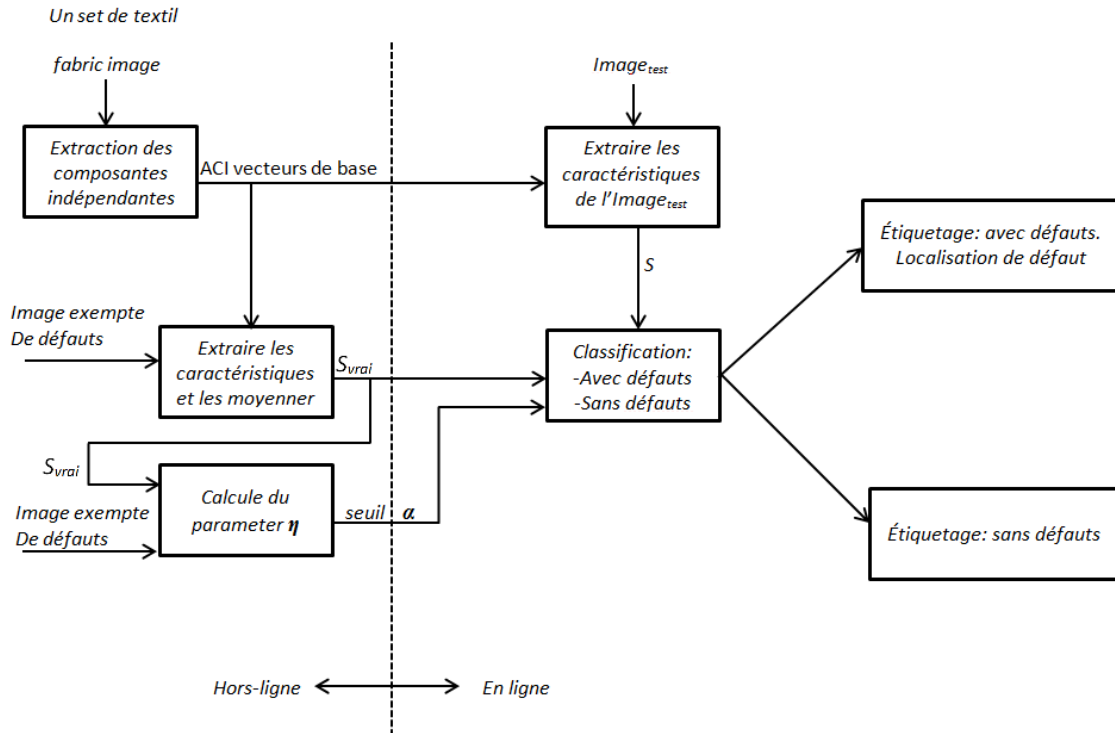


Figure 4.4 – Proposé du nouveau système de détection des défauts de textiles basé sur l'ACI.

4.6 Sommaire du chapitre

Dans ce chapitre, nous avons présenté une méthode de détection des défauts de textiles qui se base sur l'ACI. Nous avons expliqué le cadre théorique et mathématique de cette méthode et nous l'avons implémenté d'une manière fidèle par rapport à ce qui a été cité

dans [57, 58]. Suite aux expérimentations effectuées et aux résultats obtenus, des pistes d'amélioration nous sont alors apparues et qui concernent essentiellement la méthode de seuillage des distances et la classification des images en tant que telles. Cette méthode qui assure de bonnes détections selon [58] sera considérée comme une méthode de référence avec laquelle nous allons comparer notre nouvelle méthode de détection de défauts de texture.

Chapitre 5

Modélisation avec les mélanges finis de Gaussiennes généralisées

5.1 Introduction

Dans l'analyse de texture, plusieurs méthodes ont été utilisées. À titre d'exemple, nous citons les transformées multirésolutions qui ont été exposées dans le chapitre 3. Ces transformées représentent les informations spatio-fréquentielles de l'image qui la décrivent de façon précise et compacte. Le nombre de coefficients des transformées multirésolutions est de grande taille (généralement le même nombre de pixels de l'image ou plus). L'utilisation de ces coefficients en tant que tel pour la mesure de similarité entre les images n'est pas optimal, voire non efficace. En effet, les transformées présentées dans le chapitre 3 ne sont pas invariantes à la translation et à la rotation. Par conséquent, comparer une texture avec une autre qui lui est semblable mais qui a subi une rotation ou une translation ne donne pas la bonne similarité. Bien qu'il existe d'autres transformées qui sont invariantes à la translation [32] ou à la rotation [48], elles restent toujours inefficaces pour la dis-

crimination de textures. La transformée en ondelettes redondantes [32] est invariante à la translation mais elle n'est pas invariante à la rotation. L'approche présentée dans [48] est invariante à la rotation mais elle n'est pas invariante à la translation. De plus, ces méthodes sont inefficaces dans le cas de présence de bruit dans les images. Nous pouvons conclure alors que la comparaison pixel à pixel d'une image, ou coefficient à coefficient des sous bandes d'une transformée multirésolution d'une image, n'est pas une méthode efficace pour la discrimination de textures.

Une des solutions consiste à ramener le problème de discrimination de texture en un problème de mesure de similarité entre les modèles qui représentent les images. Pour calculer un modèle d'une image, nous pouvons utiliser la modélisation statistique des données (l'image dans le cas général, la texture dans notre cas) [42]. En effet, la distribution des niveaux de gris d'une texture suit une loi de probabilité [27]. Lorsque les paramètres de cette loi sont finis et bien définis, ils permettront de décrire entièrement la texture et de façon compacte. Ces paramètres constituent le modèle ou la signature d'une texture [4, 27].

Do et Vetterli [27] ont utilisé les Gaussiennes généralisées pour modéliser les sous-bandes de la transformée en ondelettes d'une image. Leur méthode utilise la divergence de KL pour calculer la similarité entre les modèles et a été appliquée avec succès dans la recherche de textures. Néanmoins, cette approche admet des limites que nous présentons au fur et à mesure dans la section 5.2.

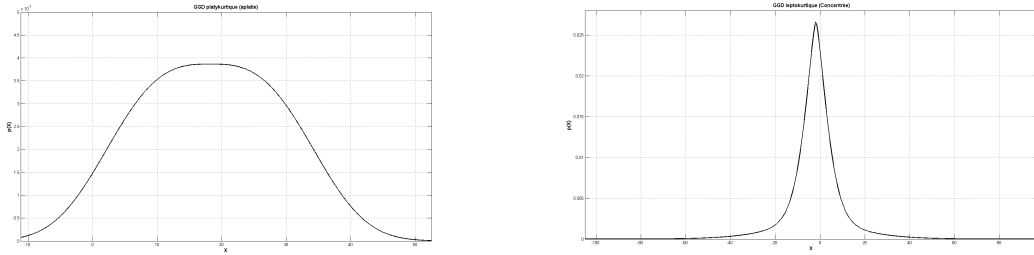
Dans ce chapitre, nous présentons une approche de modélisation efficace qui permet une meilleure discrimination de textures. Cette méthode se base sur les mélanges finis de Gaussiennes généralisées et utilise la divergence de KL pour le calcul de similarité. Nous exposons dans une première section la théorie de la modélisation statistique de données basée sur les mélanges finis de Gaussiennes généralisées. Dans la seconde section, nous montrons l'application de cette approche pour la modélisation et la détection de défauts

dans les textures de textiles.

5.2 Le modèle de mélanges finis de Gaussiennes généralisées

5.2.1 Avantages

Le principal avantage de la modélisation statistique des transformées multirésolutions de l'image est de fournir une représentation parcimonieuse et compacte pour les applications. Le problème de discrimination de texture peut être reformulé en un problème de mesure de similarité entre les modèles statistiques qui décrivent ces textures. Cette approche est relativement plus simple à implémenter que les méthodes basées sur l'énergie [27]. Actuellement, le modèle qui a marqué le maximum de succès dans la discrimination de texture est le modèle de mélanges finis de Gaussiennes généralisées [4]. En effet, la Gaussienne généralisée est meilleure que la Gaussienne du fait qu'elle a un paramètre supplémentaire communément connu par le *coefficient d'aplatissement* ou le *kurtosis*, qui permet de contrôler la forme de la distribution. Cette propriété offre une meilleure convenance pour la représentation de distributions aplaties (*platikurtiques*) et aigues (*leptokurtiques*) [3, 16]. Dans Figure 5.1, nous présentons un exemple de courbes *platikurtiques* et *leptokurtiques*.



(a) Distribution *platykurtique* ($\beta = 2.95$) (b) Distribution *leptokurtique* ($\beta = 1.73$)

Figure 5.1 – Exemple de distribution platykurtique et leptokurtique.

Maintenant que nous avons justifié le choix du modèle de mélanges finis des Gaussiennes généralisées comme approche de modélisation, nous expliquons dans ce qui suit son fonctionnement.

5.2.2 Estimation de paramètres

La distribution de Gaussienne généralisée pour une variable aléatoire unidimensionnelle $\mathbf{x} \in \mathbb{R}$ est définie dans sa forme générale comme suit [16] :

$$p(x|\mu, \sigma, \beta) = \frac{\beta \sqrt{\frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}}}{2\sigma\Gamma(1/\beta)} \exp\left(-A(\beta) \left|\frac{x - \mu}{\sigma}\right|^\beta\right), \quad (5.1)$$

où $A(\beta) = \left[\frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}\right]^{\frac{\beta}{2}}$, $\Gamma(\cdot)$ désigne la fonction gamma. μ and σ sont respectivement la moyenne et la déviation standard de la distribution. Le paramètre β ajuste l’aplatissement de la forme de la distribution. En effet, c’est grâce à ce terme que la Gaussienne généralisée est flexible dans l’ajustement de la forme. Si $\beta \rightarrow \infty$ la distribution devient uniforme, alors que dans le cas où $\beta \rightarrow 0$, la distribution devient une fonction delta centrée à μ . Si $\beta = 2$ la fonction devient une Gaussienne. S’il s’agit de données multimodales, la distribution marginale d’une variable aléatoire $\mathbf{x} \in \mathbb{R}$ se décrit comme suit [7] :

$$p(x|\vec{\theta}) = \sum_{i=1}^K \pi_i p(x|\mu_i, \sigma_i, \beta_i), \quad (5.2)$$

tel que $0 < \pi_i \leq 1$ et $\sum_{i=1}^K \pi_i = 1$. $\vec{\theta}$ désigne l'ensemble des paramètres du modèle $\{\pi_i, \mu_i, \sigma_i, \beta_i, i = 1, \dots, K\}$. Le but dans la modélisation est de trouver les paramètres d'un modèle ajustant le mieux la forme des données tout en gardant une complexité raisonnable de ce dernier (nombre de composantes). Les paramètres sont les suivants :

- K : le nombre de composantes. C'est à dire, le nombre de Gaussiennes généralisées dans un modèle.
- π_i : le poids de la composante i .
- μ_i : la moyenne de la composante i .
- σ_i : l'écart type de la composante i .
- β_i : le paramètre de forme de la composante i .

Pour estimer le nombre optimal de composantes K , l'algorithme de modélisation utilise la méthode de longueur minimale de message (*'Minimum message length (MML)'*) [49, 64]. Il est prouvé dans [49, 64] que le principe MML offre un moyen optimal pour la sélection du modèle, par ce qu'il offre un bon compromis entre la complexité du modèle et l'ajustement aux données [7, 64]. Wallace et Freeman [65] ont préconisé l'utilisation du MML comme moyen optimal pour coder les données, car il sélectionne le modèle avec ses paramètres qui donne la longueur global de message la plus courte. Un modèle plus complexe a besoin d'une longueur de message plus grande pour coder les paramètres du modèle, mais donne un meilleur ajustement pour les données. D'autre part, un modèle simple a besoin d'une longueur plus courte, mais peut diminuer la précision d'ajustement des données. Par conséquent, le MML offre un compromis naturel entre la complexité du modèle et la qualité de l'ajustement [4]. Le MML d'un ensemble de données $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ est donné comme suit :

$$MessL \simeq -\log p(\vec{\theta}) + \frac{1}{2} \log |I(\vec{\theta})| + \frac{c}{2} (1 + \log \frac{1}{12}) - \log p(\mathcal{X}|\vec{\theta}), \quad (5.3)$$

où, $p(\vec{\theta})$, $I(\vec{\theta})$ et $p(\mathcal{X}|\vec{\theta})$, désignent respectivement, la distribution a priori des paramètres $\vec{\theta}$, la matrice (attendu) d'information de Fisher et la vraisemblance des données. $|I(\vec{\theta})|$ désigne dans cette formule le déterminant de la matrice $I(\vec{\theta})$. La constante $c = 4K$ donne le nombre total des paramètres du modèle. Le chiffre 4 provient du nombre de paramètres par composante de Gaussienne généralisée qui sont notés par $\vec{\theta}$. L'initialisation des paramètres de $\vec{\theta}$ se fait comme suit [4] : Les probabilités a priori des composantes du modèle sont initialisés suivant cette formule $p(\pi_1, \dots, \pi_K) = \prod_{i=1}^K \pi_i^{-1/2}$. La moyenne et l'écart type de chaque composante sont initialisés respectivement par $p(\mu_i) = \frac{1}{2 \times \sigma_{pop}}$ et $p(\sigma_i) = \frac{1}{\sigma_{pop}}$ tel que $0 \leq \sigma_i \leq \sigma_{pop}$ et $\mu_{pop} - \sigma_{pop} \leq \mu_i \leq \mu_{pop} + \sigma_{pop}$ et en supposant que μ_{pop} et σ_{pop} sont la moyenne et l'écart type tirés des données initiales \mathcal{X} . $p(\beta_i) = \frac{1}{Kur(\mathcal{X})}$ dans l'intervalle $0 \leq \beta_i \leq Kur(\mathcal{X})$ tel que $Kur(\mathcal{X})$ représente le 'Kurtosis' de la donnée \mathcal{X} . L'équation 5.3 est minimisée à l'aide de l'algorithme expectation-maximisation, communément connu sous le nom 'Expectation Maximization' (EM)[7]. C'est l'implémentation du maximum de vraisemblance de données. L'étape de l'expectation consiste à calculer la probabilité a priori des composantes du modèle. L'étape de maximisation consiste à la mise à jour des paramètres du modèle dans le but d'augmenter la vraisemblance des données avec le modèle calculé. Nous obtenons ainsi les formules de mise à jour suivantes :

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^n p(\theta_k | x_i) \quad (5.4)$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^n p(\theta_k | x_i) |x_i - \mu_k|^{\beta_k - 2} x_i}{\sum_{i=1}^n p(\theta_k | x_i) |x_i - \mu_k|^{\beta_k - 2}} \quad (5.5)$$

$$\hat{\sigma}_k = \left[\frac{\beta_k A(\beta_k) \sum_{i=1}^n p(\theta_k | x_i) |x_i - \mu_k|^{\beta_k}}{\sum_{i=1}^n p(\theta_k | x_i)} \right]^{1/\beta_k} \quad (5.6)$$

où θ_k désigne les paramètres de la composante k du modèle et $A(\beta_k)$ est défini au début de cette section dans la formule 5.1. La probabilité a priori $p(\theta_k | x_i)$ est donnée par $p(\theta_k | x_i) = \frac{\pi_k p(\mathbf{x}_i | \theta_k)}{\sum_{j=1}^K \pi_j p(\mathbf{x}_i | \theta_j)}$. Le paramètre β_k de chaque composante K est estimée simultanément avec

les autres paramètres en utilisant la méthode de Newton-Raphson :

$$\hat{\beta}_k \simeq \beta_k - \left\{ \frac{\partial^2 \log(p(\mathcal{X}|\vec{\theta}))}{\partial \beta_k^2} \right\}^{-1} \frac{\partial \log(p(\mathcal{X}|\vec{\theta}))}{\partial \beta_k}, \quad (5.7)$$

où $p(\mathcal{X}|\vec{\theta})$ est la vraisemblance de la génération des données par le modèle. Dans le script suivant, nous présentons un résumé de l'algorithme d'estimation du modèle de mélanges finis des Gaussiennes généralisées.

Algorithm 2 *Algorithme de sélection de modèles de mélanges finis des Gaussiennes généralisées.*

```

for  $K = K_{max} \rightarrow K_{min}$  do
  repeat
    1. Estimer le paramètre  $\mu_k$  utilisant Eq. (5.4).
    2. Estimer le paramètre  $\sigma_k$  utilisant Eq. (5.5).
    3. Estimer le paramètre  $\pi_k$  utilisant Eq. (5.6).
    3. Estimer le paramètre  $\beta_k$  utilisant Eq. (5.7).
  until (Convergence)
  Calculer la longueur de message utilisant Eq. (5.3).
end for
Retourner le modèle optimal  $K^*$  avec ses paramètres.

```

Notons ici que K_{min} et K_{max} peuvent être fixés pour toutes les estimations et que le test de plusieurs valeurs de K permet d'obtenir le K optimal.

La distribution des coefficients d'une transformée multirésolution peut être multimodale (*le nombre de pics* ≥ 2). La distribution peut être aussi asymétrique, à queue lourde et/ou encore fortement pointue. Dans tous ces cas-là, il est difficile, voire impossible de trouver une distribution unimodale qui peut ajuster efficacement la forme de la distribution de données à modéliser. Les mélanges finis de Gaussiennes généralisées [4] permettent de dénouer toutes ces contraintes résultants ainsi en un meilleur ajustement de la forme de la distribution.

Pour montrer la force de la modélisation par les mélanges finis de Gaussiennes généralisées, nous présentons dans Figures 5.3 et 5.4 la modélisation des coefficients de la

transformée en contourlettes d'une image de texture de la base de texture 'Vistex' [70]. Figure 5.2 représente le premier Bloc de taille $128 * 128$ de l'image 'Fabric.0000' extraite de l'album 'Fabric'.

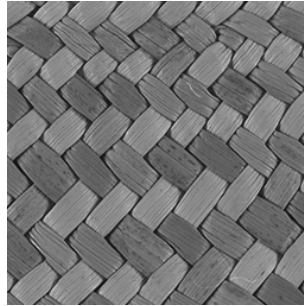


Figure 5.2 – Image de texture extraite de Vistex [70].

Les histogrammes des coefficients de la transformée sont représentés par les courbes de surfaces denses et de couleur bleue (surfaces remplies), alors que les modèles de mélanges finis de Gaussiennes généralisées de ces coefficients sont représentés par les courbes en lignes continues de couleur noire. Nous adoptons la même présentation dans toutes les présentations de modélisation qui suivent. Figures 5.3 et 5.4 montrent comment ce modèle est capable d'ajuster les formes des histogrammes des coefficients correspondants.

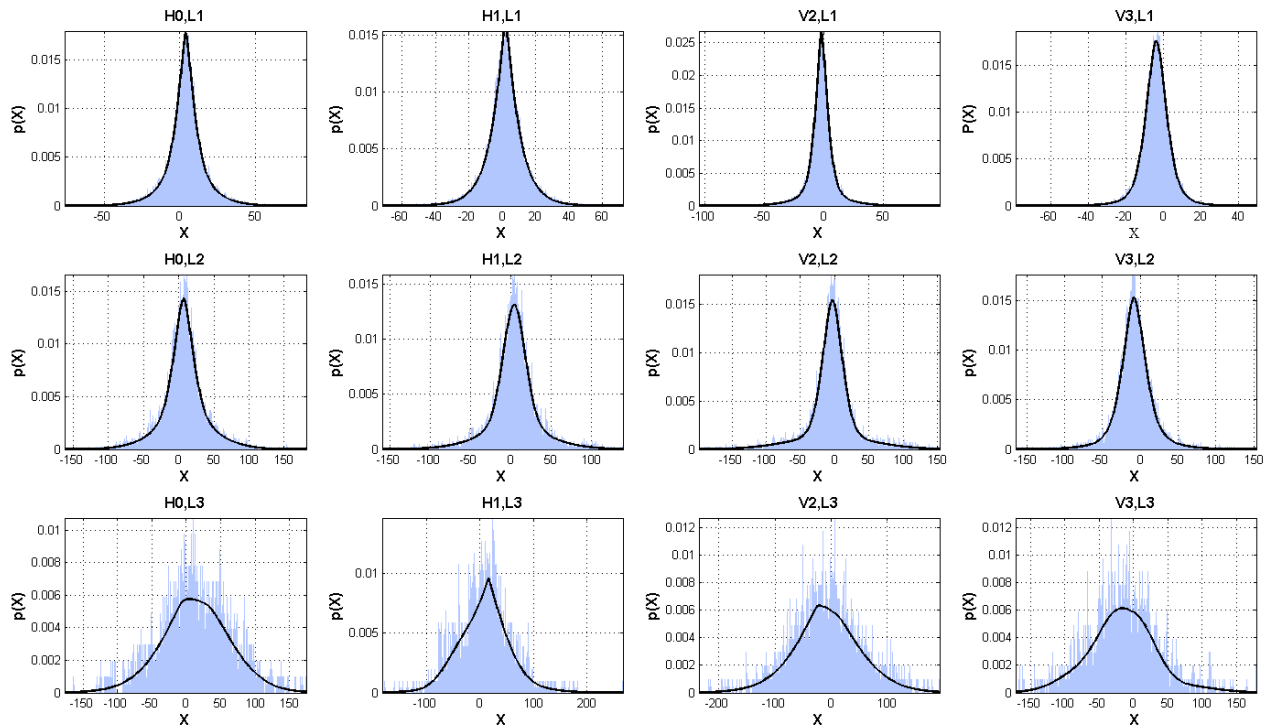


Figure 5.3 – Histogrammes des sous-bandes contourlettes de l'image Fabric et leurs modélisations MoGG.

Dans Figure 5.3, les histogrammes des sous-bandes $(H0, L1)$, $(H1, L1)$, $(V2, L1)$ et $(V3, L1)$ sont *leptokurtiques*. Le modèle MoGG ajuste parfaitement la forme de la distribution des données. Les histogrammes des sous-bandes $V3, L1$, $V3, L2$, $V3, L3$ ne sont pas centrés autour de zéro et pour autant le modèle de mélanges finis de Gaussiennes généralisées à bien ajusté la forme de leurs distributions. Les histogrammes des sous-bandes $(H0, L3)$ et $(V3, L3)$ sont relativement *platykurtiques* et la modélisation MoGG ajuste parfaitement la forme. L'ajustement de la forme des distributions à queue lourde est bien assuré par le modèle de mélanges finis des Gaussiennes généralisées. En effet, les histogrammes des sous-bandes $(H0, L1)$, $(H1, L1)$, $(V2, L1)$ et $(V3, L1)$ sont à queue lourde et nous voyons la bonne modélisation des données avec notre modèle.

Dans Figure 5.4, l’histogramme de l’image d’approximation est multimodal ; *le nombre de pics* ≥ 2 . D’ailleurs, nous remarquons la présence de 3 régions. Le modèle a bien ajusté la forme de la distribution. Il est évident de remarquer que n’importe quel modèle unimodal est incapable d’assurer un tel ajustement précis de la forme de telles données.

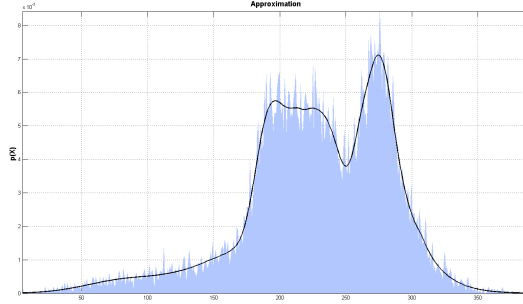


Figure 5.4 – Histogramme de l’image d’approximation de l’image Fabric et sa modélisation MoGG.

5.2.3 Calcul de similarité entre MoGGs

Do et al. [27] ont développé une formule exacte de la divergence de KL entre deux Gaussiennes généralisées centrées. Les auteurs ont montré également que cette divergence est meilleure dans la comparaison des modèles comparant aux méthodes basées sur l’énergie et la distance Euclidienne. Cependant, dans le cas de mélanges de Gaussiennes généralisées cette formulation exacte de la divergence de KL ne peut pas être développée [33, 36]. Dans [4], l’auteur a développé une approximation de la divergence de KL entre deux MoGG basée sur la méthode d’échantillonnage de Monte Carlo [52]. La technique s’explique comme suit : Étant données deux modèles de MoGG $f(x) = \sum_{i=1}^K \pi_i p(x|\theta_i)$ et $g(x) = \sum_{j=1}^M \omega_j p(x|\theta_j)$, la divergence de KL entre f et g notée par KL est définie comme suit :

$$KL(f||g) = \int f \log \left(\frac{f}{g} \right) dx = \int \sum_{i=1}^K \pi_i f_i \log \left(\frac{\sum_{i=1}^K \pi_i f_i}{\sum_{j=1}^M \omega_j g_j} \right) dx, \quad (5.8)$$

Où f_i et g_i représentent respectivement $p(x|\theta_i)$ et $p(x|\theta_j)$.

L'approximation de la KL par l'intégration de Monte Carlo est définie par :

$$KL_{mc}(f||g) = \frac{1}{n} \sum_{i=1}^n \log \frac{f(x_i)}{g(x_i)} \approx KL(f||g), \quad (5.9)$$

La méthode d'approximation par échantillonnage vise à générer un échantillon suffisamment grand $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ tiré indépendamment de la distribution f afin d'approximer la divergence de KL en utilisant une intégration de Monte Carlo.

5.3 Application à la détection de défauts de texture dans les textiles

Dans cette section, nous présentons l'application du modèle de mélanges finis de Gaussiennes généralisées pour modéliser la distribution des sous-bandes de la transformée en ondelettes. L'algorithme est constitué de deux phases. La première est une phase d'apprentissage, dans laquelle nous récupérons les modèles des sous-bandes multirésolutions d'un bloc exempt de défauts. La seconde phase consiste à calculer les modèles des sous-bandes multirésolutions de l'image à traiter et calculer la divergence de KL entre ces modèles et les modèles du bloc exempt de défauts. Pour effectuer les expérimentations nécessaires, nous avons utilisé une base de données de textures de textile appelée '*TILDA*' [71].

5.3.1 Phase d'apprentissage

La phase d'apprentissage de l'algorithme de détection de défauts de texture dans les textiles consiste à calculer les modèles de distributions des sous-bandes multirésolutions

d'un tissu exempt de défauts pour les comparer avec les modèles des autres images. La divergence de KL ainsi obtenue sera utilisée pour déterminer si l'image en question est exempte de défauts ou non. Dans notre algorithme, le traitement se fait par bloc. Ce choix est indispensable pour la détection des différents types et tailles de défauts et la localisation de ces derniers en cas de leur présence. Nous expliquerons ces deux raisons dans la phase de détection. La première tâche dans cette phase est relativement simple et consiste à choisir un bloc d'image exempt de défaut. Dans la base de données TILDA [71], pour chaque type de texture il y a 50 images exemptes de défauts. Nous choisissons alors une image à partir de laquelle nous choisissons un bloc qui sert comme référence. Nous expliquons aussi le choix de la taille du bloc, qui d'ailleurs, doit être fixée durant tout le traitement, dans la phase de détection. Une fois le bloc de référence choisi, nous appliquons la transformée multirésolution sur ce bloc pour récupérer les sous-bandes multirésolutions. Le nombre de niveaux de la transformée est aussi un autre paramètre à expérimenter. Maintenant que les sous-bandes multirésolutions sont produites, nous appliquons l'algorithme de modélisation par les mélanges fini des Gaussiennes généralisées afin de récupérer les modèles nécessaires.

5.3.2 Phase de détection

Comme cité dans la phase d'apprentissage, le traitement effectué par notre algorithme se fait par bloc. Deux raisons nous poussent à opter pour ce choix. Premièrement, le traitement par bloc nous permet de localiser le défaut en cas de son présence. En effet, si le bloc traité est détecté comme défectueux, il sera étiqueté par l'algorithme et nous obtenons ainsi les localisations des blocs défectueux dans une image. Deuxièmement, si l'image à traiter contient un défaut pointu tel que présenté dans Figure 5.6, l'application de l'algorithme de détection sur cette image globale de taille 768×512 pixels produit des modèles qui convergent vers les modèles de tissus sain du fait que ce défaut ne change pas

de façon remarquable la distribution des sous-bandes multirésolution indépendamment du type de la transformée en multirésolution utilisée. Figure 5.5 représente la modélisation par les mélanges finis des Gaussiennes généralisées des sous-bandes de la transformée en ondelettes d'une image exempte de défauts. Nous remarquons que la plupart des modèles des deux images sont similaires et le calcul de la divergence de KL donnera une distance relativement petite.

La taille des blocs est un autre paramètre à fixer. Si la taille des blocs est relativement grande, la localisation des défauts n'est pas optimale et certains défauts qui sont pointus ou qui se présentent dans les bords des blocs ne seront pas détectés par l'algorithme. Si la taille des blocs est plus petite que la taille des filtres de la transformée en multirésolution, le traitement sera impossible et les résultats obtenues seront totalement erronés. Nous montrons dans Figure 5.5 la modélisation pour une image exempte de défauts et dans Figure 5.6 la modélisation d'une image contenant un défaut pointu. En comparant les modèles obtenus nous concluons qu'ils sont similaires et nous déduisons que la divergence de KL sera minime. Ainsi, l'image contenant un défaut est considérée comme exempte de défauts.

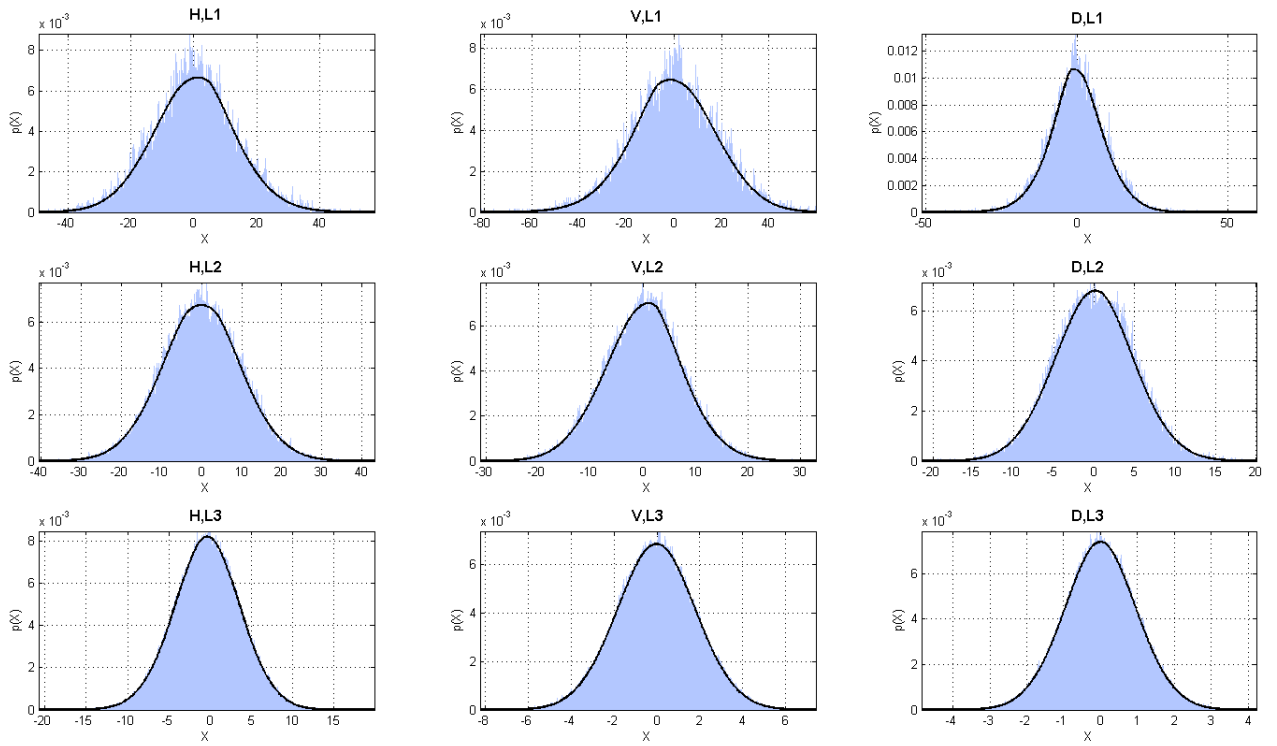
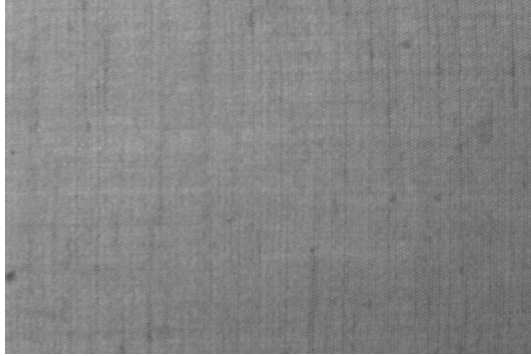


Figure 5.5 – La modélisation par les mélanges finis des Gaussiennes généralisées des sous-bandes de la transformée en ondelettes d’une image exempte de défauts.

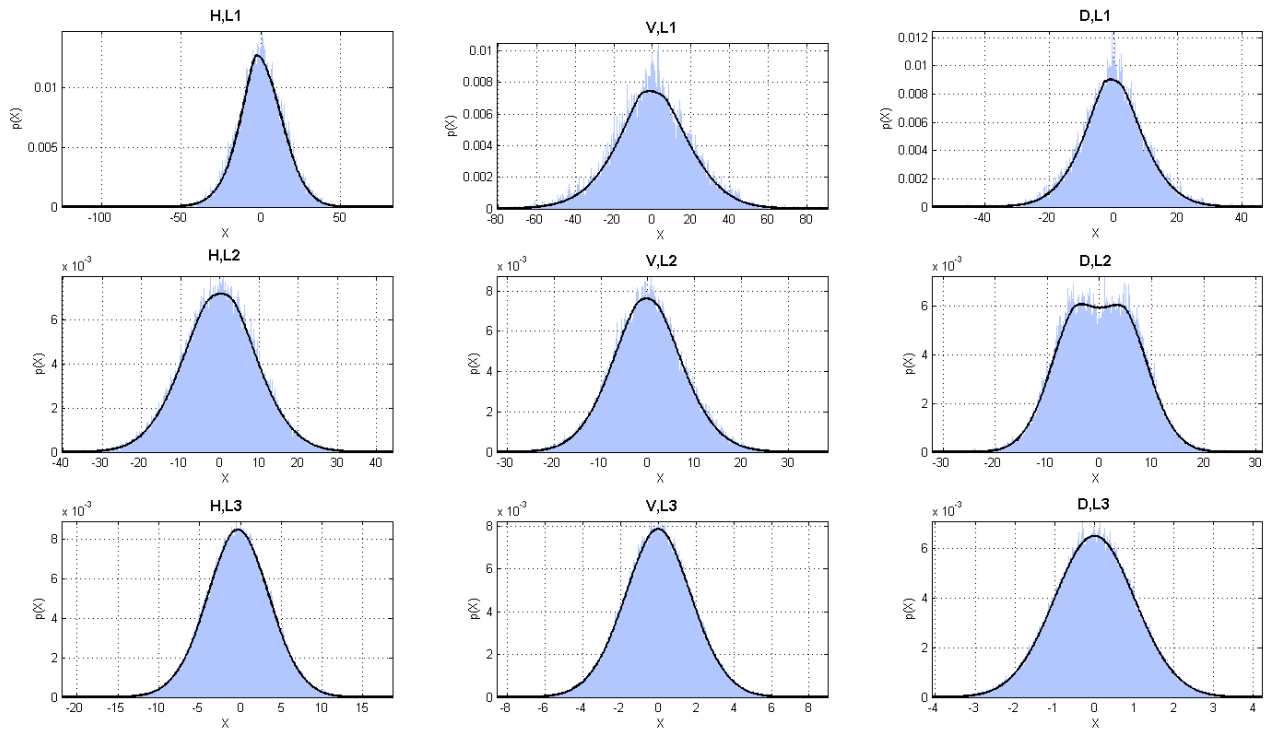
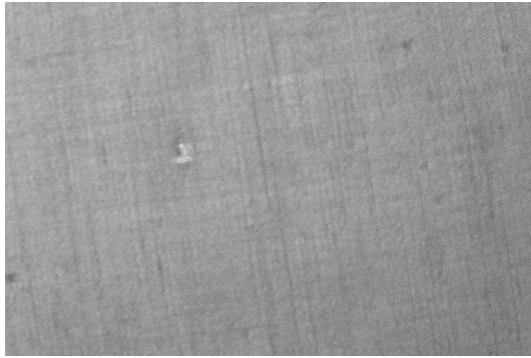


Figure 5.6 – La modélisation par les mélanges fins des Gaussiennes généralisées des sous-bandes de la transformée en ondelettes d’une image contenant un défaut pointu.

La phase de détection consiste alors à découper une image en blocs de taille préalablement fixée, et ensuite, appliquer l'algorithme de modélisation par les mélanges finis des Gaussiennes généralisée afin d'obtenir les modèles de chaque bloc. Une fois tous les modèles sont calculés, nous calculons la divergence de de KL entre les modèles de chaque bloc et les modèles du bloc fixé comme référence. Si la divergence de KL est petite le bloc est étiqueté comme exempt de défauts, sinon il est étiqueté contenant au moins un défaut. Dans Figure 5.7, nous présentons un schéma global de l'algorithme de détection de défauts en se basant sur la modélisation des sous-bandes de transformées multirésolutions par les mélanges finis de Gaussiennes généralisées.

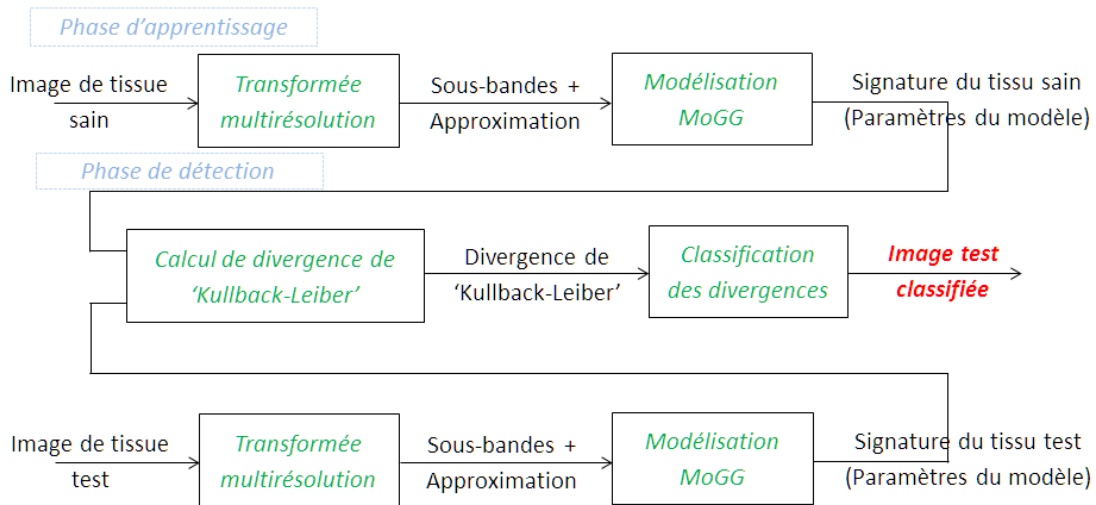


Figure 5.7 – L'algorithme de détection de défauts basé sur la modélisation MoGG des sous-bandes des transformées multirésolutions.

Figure 5.8 montre les résultats de l'application de l'algorithme de détection de défauts de texture présenté dans Figure 5.7. Les blocs qui sont cadrés en noir et marqués au centre par le signe (+) représentent les blocs défectueux qui ont été détectés par l'algorithme. Les autres représentent les blocs exempts de défauts. Le découpage de cette image en blocs de taille 128×128 pixels produit 24 blocs répertoriés en 4 lignes et 6 colonnes. Il est évident de remarquer dans cette image que la totalité de la 3^{eme} ligne représente un défaut

alors que le reste de l'image est exempt de défauts. L'algorithme de détection de défauts appliqué sur cette image est basé sur la transformée en ondelettes et la modélisation MoGG. Nous remarquons que l'algorithme a échappé la détection d'un bloc défectueux et a détecté un bloc exempt de défaut comme défectueux. Alors, nous pouvons conclure que l'algorithme est parvenu à faire 22 détections correctes sur les 24 détections qu'il devrait faire.

Pour illustrer l'importance du principe de découpage en bloc de l'image à traiter nous présentons dans deux figures séparées la modélisation des sous-bandes de la décomposition en ondelettes d'un bloc contenant un défaut et d'un bloc exempt de défaut. Dans Figure 5.8, le bloc de coordonnées (1, 1) est exempt de défauts. Nous présentons la modélisation de ces sous-bandes ondelettes, contourlettes et contourlettes redondantes respectivement dans Figure 5.9, . Le bloc de coordonnées (3, 3) est un bloc contenant un défaut et sa modélisation est présentée dans Figure 5.12.

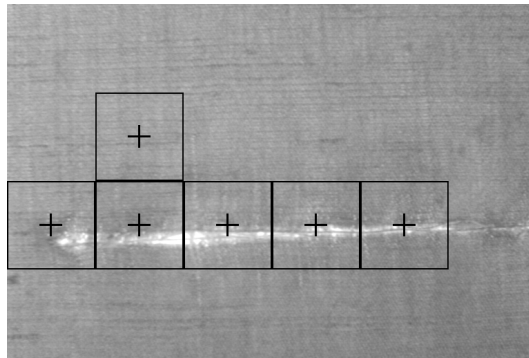
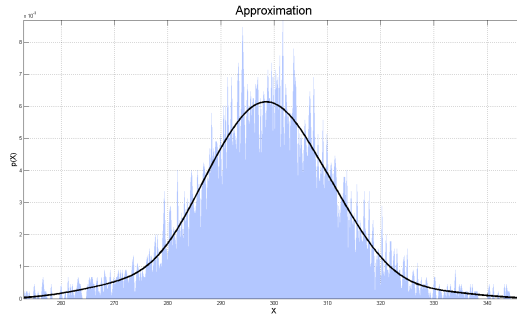
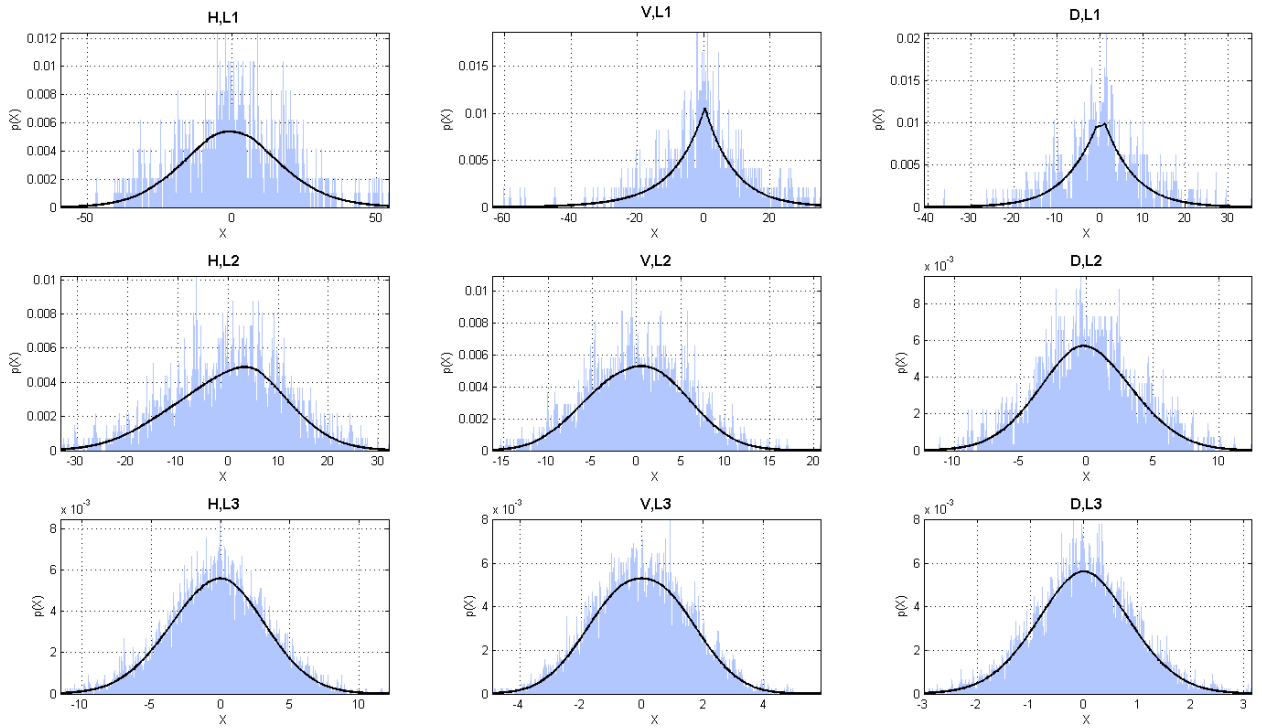


Figure 5.8 – Exemple de détection de défauts.

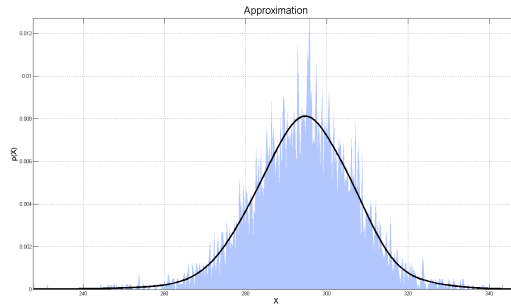


(a) Modélisation de l'image d'approximation

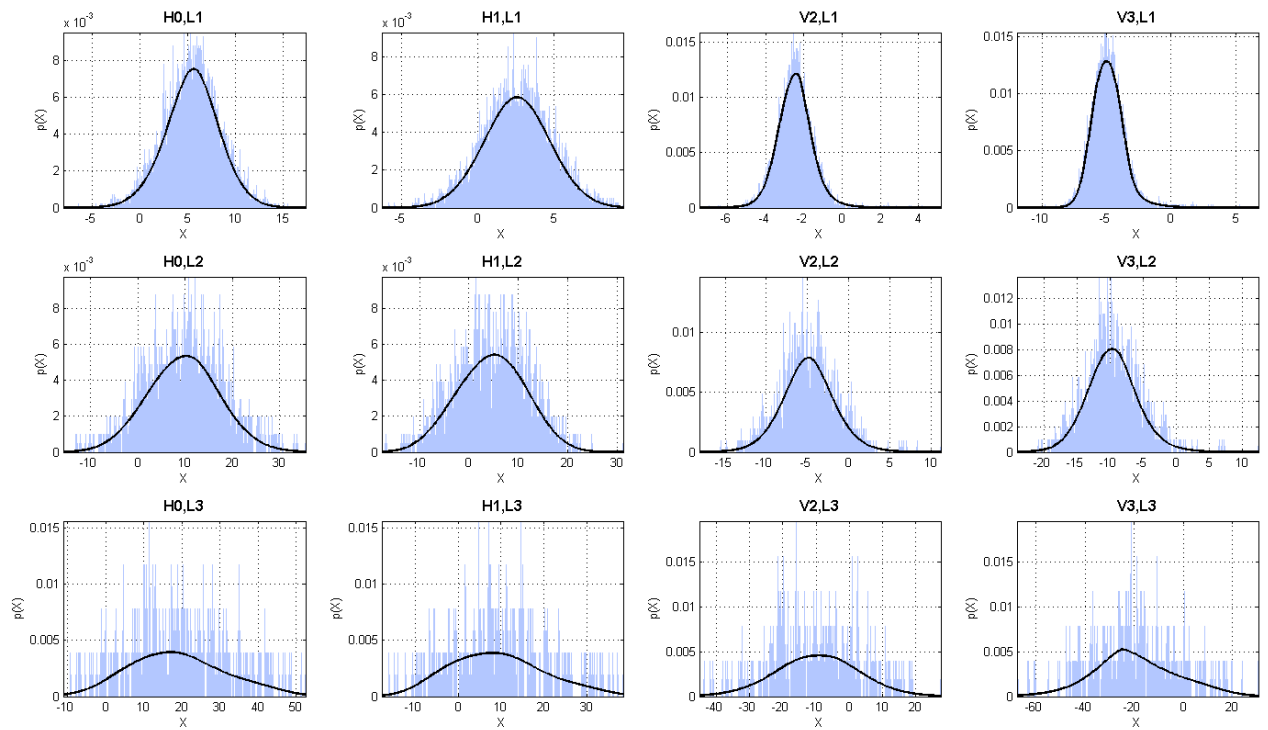


(b) Modélisation des sous-bandes ondelettes

Figure 5.9 – La modélisation MoGG des sous-bandes ondelettes du bloc(1,1) de l'image présentée dans la Figure 5.8.

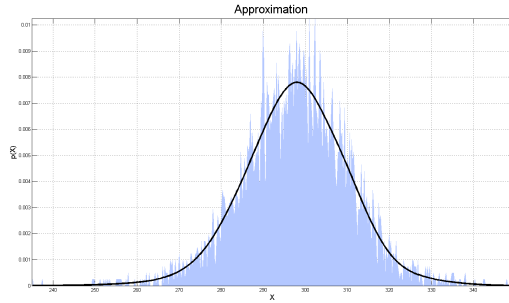


(a) Modélisation de l'image d'approximation

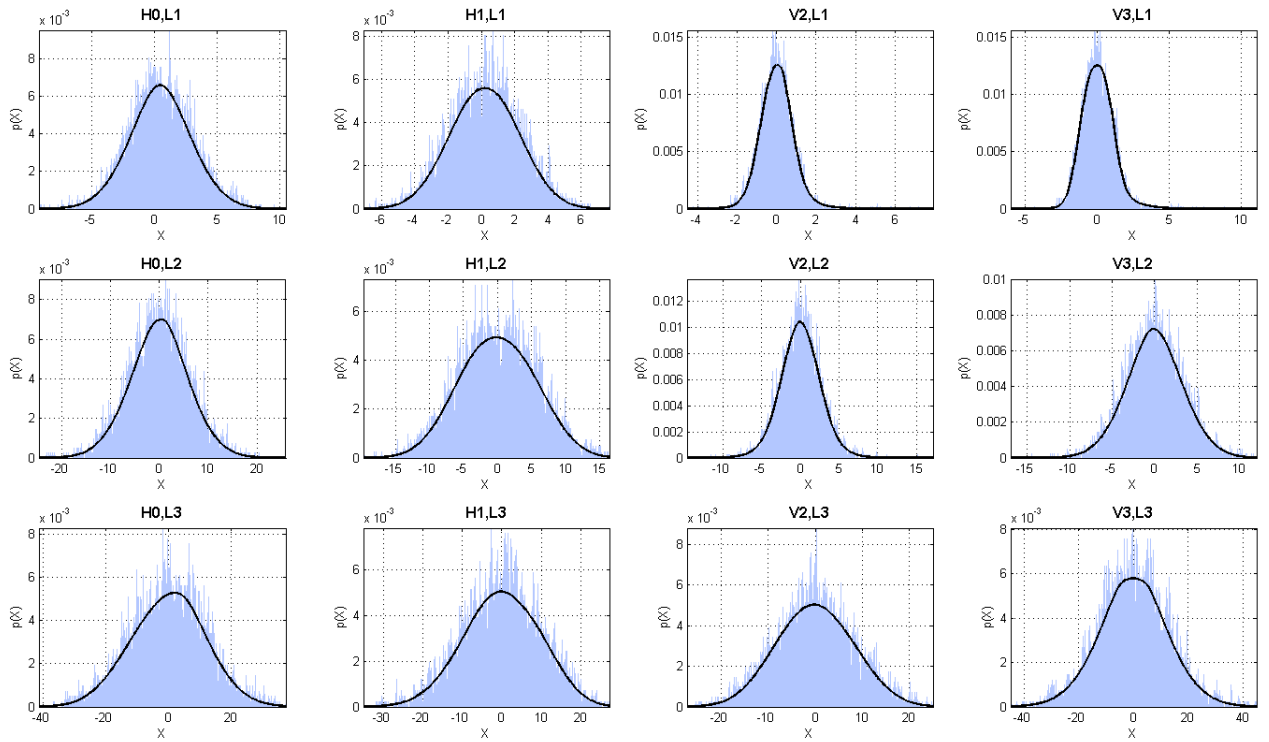


(b) Modélisation des sous-bandes contourlettes

Figure 5.10 – La modélisation MoGG des sous-bandes contourlettes du bloc(1,1) de l'image présentée dans la Figure 5.8.

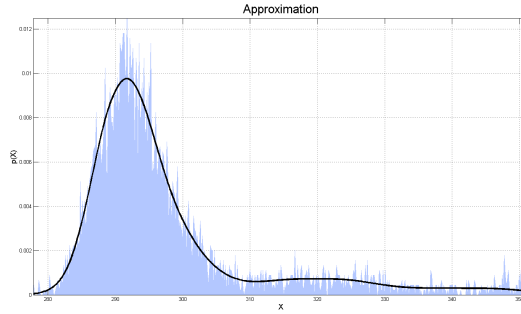


(a) Modélisation de l'image d'approximation

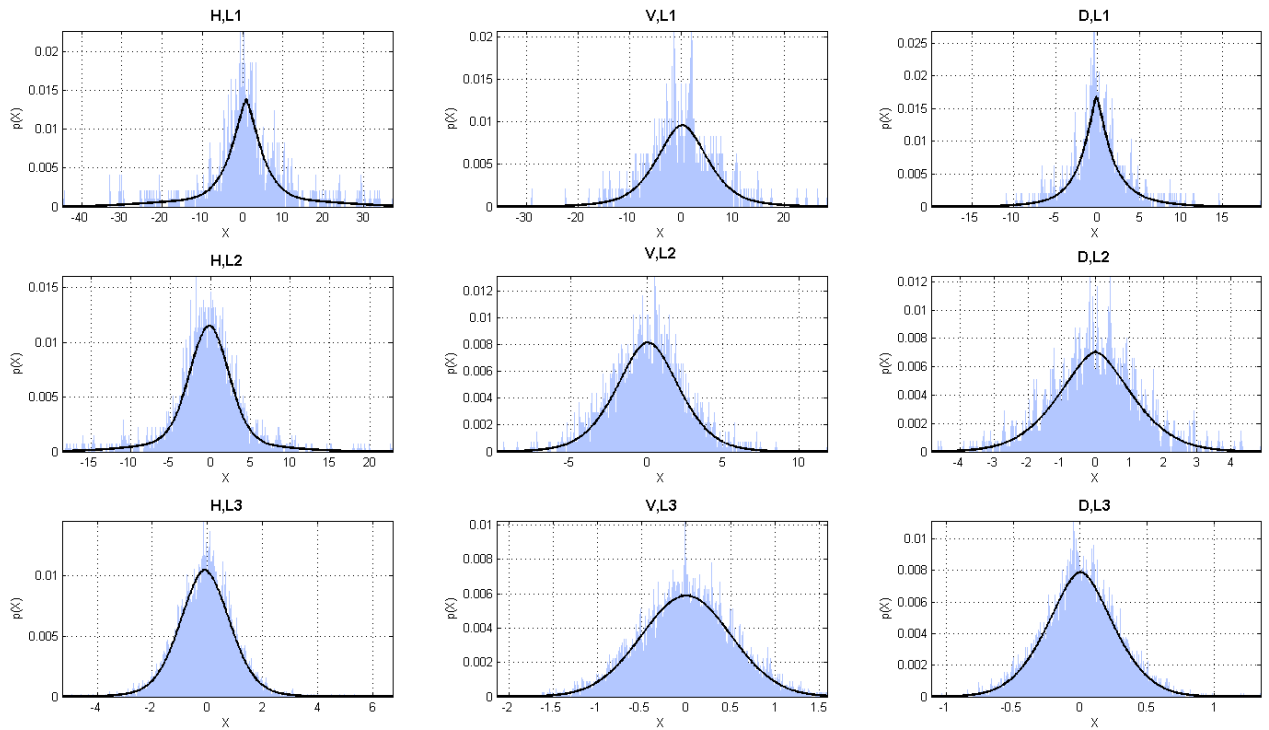


(b) Modélisation des sous-bandes contourlettes redondantes

Figure 5.11 – La modélisation MoGG des sous-bandes contourlettes redondantes du bloc(1,1) de l'image présentée dans la Figure 5.8.



(a) Modélisation de l'image d'approximation



(b) Modélisation des sous-bandes ondelettes

Figure 5.12 – La modélisation MoGG des sous-bandes ondelettes du bloc(3,3) de l'image présentée dans la Figure 5.8.

Si nous comparons les modèles MoGGs obtenus, nous remarquons qu'il y a une grande différence entre les modèles du bloc exempt de défauts (voir Figure 5.9) et les modèles du bloc défectueux (voir Figure 5.12). Lors du calcul de la divergence de KL, nous obtenons une grande valeur qui exprime une divergence importante. De plus, nous remarquons que la totalité des modèles représentent un mélange de plusieurs Gaussiennes généralisées (aux moins 2). Ceci se remarque essentiellement dans l'image d'approximation du bloc défectueux (voir partie (a) Figure 5.12). De ce fait, nous concluons que le traitement de l'image doit être effectué inévitablement par bloc.

5.4 Sommaire du chapitre

Dans ce chapitre, nous avons présenté la méthode de modélisation par les mélanges finis de Gaussiennes généralisés tout en exposant les avantages qu'elle présente par rapport aux méthodes existantes dans l'état de l'art. Les résultats de l'application de cette méthode dans le domaine de recherche d'images basée sur le contenu [4] encouragent son utilisation dans le domaine de la détection de défauts. En effet, la détection de défauts de texture peut être traitée comme un problème de comparaison entre images de texture saine et images de texture avec défauts.

Chapitre 6

Expérimentations

6.1 Introduction

Dans le chapitre 3, nous avons présenté les transformées multirésolutions et leurs avantages. Nous avons présenté aussi dans le chapitre 5 la modélisation statistique des données ainsi que l'algorithme de détection de défauts de texture dans les tissus. Dans ce chapitre, nous présentons différentes études expérimentales afin de comparer notre méthode à la méthode de référence présentée dans le chapitre 4. De plus, nous montrons la transformée multirésolution la plus appropriée à la détection de défauts de textures dans les tissus. D'un autre côté, nous présentons la méthode d'apprentissage pour le calcul automatique des seuils des divergences de KL, la taille optimale des blocs ainsi que l'apport du traitement par chevauchement de blocs.

6.2 Outils de travail

Dans nos travaux, nous utilisons le logiciel Matlab qui est un environnement interactif de calcul numérique. Il permet le développement des algorithmes d'analyse de données grâce à ses fonctions mathématiques intégrées. Cet environnement contient beaucoup de bibliothèques spécifiques. Matlab englobe une bibliothèque spécifique pour le traitement et l'analyse d'image.

Nous appliquons l'algorithme de détection de défauts sur des images de textile. Pour ce faire, nous aurons recours à la base de données TILDA [71]. TILDA est une base de données de texture de textile qui a été développée en 1996 dans le cadre de l'analyse de la texture par le groupe de travail de la DFG (Deutsche Forschungsgemeinschaft, *Fondation Allemande de Recherche*). Dans cette base, nous trouvons 8 types de textures répertoriées en quatre classes ('C1'...'C4'). Chaque classe contient 3 répertoires dont un est vide et les deux autres contiennent un type de texture (R1, R2, R3). Dans chaque répertoire de texture, il y a 8 sous-répertoires, dont chacun contenant 50 images de format TIFF et de taille 768×512 pixels. Le premier sous-répertoire, toujours nommé 'E0' contient des images de textures exemptes de défauts, alors que les 7 autres répertoires ('E1'...'E7') contiennent les images de textures avec défauts. Les défauts sont capturés avec des angles de rotation différents et aléatoires. Figure 6.1 présente l'arborescence de la base de données TILDA. Le choix de cette base de données comme outil de travail se justifie par l'utilisation de cette dernière dans plusieurs travaux de recherche sur la détection de défauts [56, 57, 58], ce qui permet la comparaison des résultats obtenus par notre méthode. De plus, il est difficile de trouver d'autres bases de données de défauts de textile sur le marché. La construction d'une base de données de défauts de textile est fastidieuse et nécessite des ressources coûteuses, tels que le matériel d'acquisition d'image, l'accord avec une usine de confection de tissu et le temps nécessaire pour un tel

projet, ce qui nous pousse à l'utilisation d'une base de données déjà existante. TILDA [71] est devenue de nos jours une référence dans le domaine de détection de défauts de textile.

L'implémentation et l'exécution des différents algorithmes sont effectuées sur une station de travail équipée d'un processeur quadricoeur Intel Core I7 cadencé à 2.93 GHz, 16 Go de mémoire volatile, une carte graphique NVIDIA Quadro FX 1800 et doté d'un système d'exploitation Microsoft Windows 7 professionnel 64× pixels et un logiciel Matlab 64-bit version R2011b.

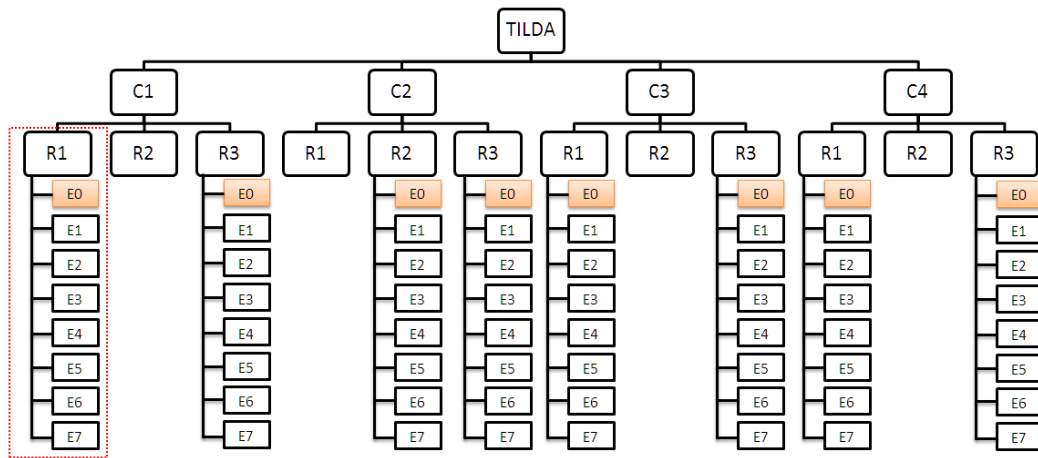
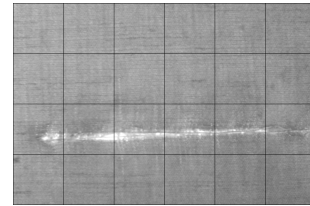
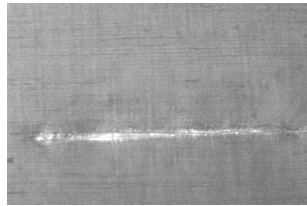


Figure 6.1 – Arborescence de la base de données TILDA.

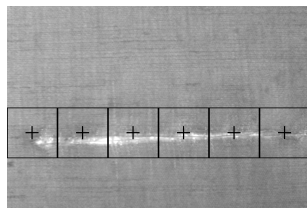
6.3 Méthode d'évaluation

Notre méthode de détection de défauts de texture dans les tissus est basée sur deux phases principales. La première phase est celle de la transformée multirésolution des blocs d'image qui produit les différentes sous-bandes multirésolutions, alors que la seconde phase constitue la modélisation par les mélanges fins de Gaussiennes généralisées. TILDA

ne contient pas la vérité terrain sur les défauts des images de textures. En effet, les seules images dont la vérité terrain est connue sont les images des répertoires 'E0' (images exemptes de défauts). La construction de la vérité terrain pour les images contenant des défauts est un exercice fastidieux et nécessite du temps. De plus, la vérité terrain construite est basée sur une évaluation subjective. En outre, dans nos travaux nous utilisons différentes tailles de blocs (64×64 pixels et 128×128 pixels), et le traitement peut être avec ou sans chevauchement de blocs. Alors, il faut construire pour une même image, les vérités terrains correspondant à ces paramètres. Nous avons extrait de la base TILDA, 6 images exemptes de défauts (à partir du répertoire 'C1/R1/E0') et 24 images contenant divers défauts (à partir des répertoires 'C1/R1/E1'...'C1/R1/E7'). Nous avons construit la vérité terrain de chacune des 30 images choisies afin de pouvoir évaluer et comparer les différentes variantes de l'algorithme. Figure 6.2 montre un exemple de construction de la vérité terrain d'une image de textile contenant un défaut.



(a) Image de TILDA contenant un défaut (b) Découpage en blocs de 128×128 pixels



0	0	0	0	0	0
0	0	0	0	0	0
1	1	1	1	1	1
0	0	0	0	0	0

(c) Image avec la vérité terrain

(d) Matrice de vérité terrain

Figure 6.2 – Exemple de construction de la vérité terrain d'une image contenant un défaut.

Lors de l'application de notre méthode de détection de défauts, nous affectons à chaque bloc de l'image la classe correspondante selon la divergence de KL obtenue. Si la diver-

gence est supérieure à un seuil δ_d , le bloc est considéré comme défectueux et l'étiquette affectée est 1, sinon le bloc est considéré exempt de défauts et l'étiquette affectée est 0. À partir des classifications des blocs de l'image, nous construisons la matrice de détection de l'image. Ainsi, l'évaluation de la performance de l'algorithme devient une simple comparaison entre la matrice de détection et la matrice de vérité terrain de l'image. La comparaison s'effectue avec l'opérateur logique *XOR* (Ou exclusif). Lorsque la valeur de classification affectée par l'algorithme est différente de la valeur de la vérité terrain, le résultat de la comparaison donne la valeur logique 1, qui veut dire qu'il s'agit d'une erreur de classification de bloc. Sinon le résultat de la comparaison donne la valeur logique 0, qui veut dire qu'il existe une concordance entre la détection et la vérité terrain. Dans le Tableau 6.1, nous présentons la table de vérité de l'opérateur logique *XOR* et les significations correspondantes à notre problématique.

Valeur de la matrice de vérité terrain (V)	Valeur de la matrice de détection (D)	(V) <i>XOR</i> (D) : signification
0 : Bloc exempt de défauts	0 : Bloc exempt de défauts	0 : Le bloc est exempt de défauts et est bien détecté comme exempt de défauts.
0 : Bloc exempt de défauts	1 : Bloc contenant des défauts	1 : Erreur de détection ; le bloc est exempt de défauts alors qu'il est détecté comme porteur de défauts.
1 : Bloc contenant des défauts	0 : Bloc exempt de défauts	1 : Erreur de détection ; le bloc est défectueux alors qu'il est détecté exempt de défauts.
1 : Bloc contenant des défauts	1 : Bloc contenant des défauts	0 : La détection est correcte ; le bloc contient des défauts et l'algorithme l'a bien détecté.

Tableau 6.1 – Table de vérité de l'opérateur logique *XOR*.

La comparaison de la matrice de détection et la matrice de vérité terrain peut être effectuée par le calcul de distance de Hamming [35]. Ainsi, les formules de taux de détection ainsi que le taux d'erreurs se présentent comme suit :

$$\text{Taux de détection} = \frac{(K \times N) - \sum_{i=1}^K \sum_{j=1}^N D_{ij} \oplus V_{ij}}{(K \times N)}, \quad (6.1)$$

$$\text{Taux d'erreurs} = \frac{\sum_{i=1}^K \sum_{j=1}^N D_{ij} \oplus V_{ij}}{(K \times N)}, \quad (6.2)$$

où D et V représentent respectivement la matrice de marquage et la matrice de vérité terrain de l'image et K et N représentent respectivement le nombre de lignes et le nombre de colonnes des matrices. Le symbole \oplus représente l'opérateur logique *XOR*.

6.4 Intégration des différentes transformées multirésolutions et classification des divergences de KL

Les transformées multirésolutions étudiées dans le chapitre 3 présentent différentes propriétés et ne produisent pas les mêmes résultats. L'objectif de cette section est de montrer la transformée multirésolution la plus appropriée pour la détection de défauts de texture ainsi que la meilleure méthode de classification. Nous avons créé 3 variantes de l'algorithme de détection de défauts de texture dans les tissus. La première utilise la transformée en ondelettes discrètes (DWT), la seconde utilise la transformée en contourlettes (CT) et la troisième utilise la transformée en contourlettes redondantes (RCT). Les trois variantes utilisent la même modélisation statistique, à savoir, la modélisation par les mélanges finis de Gaussiennes généralisées (MoGG). Nous avons choisi manuellement un bloc d'image exempt de défauts noté \mathcal{B}_r . Pour éliminer les problèmes de variation d'illumination, probablement dus aux conditions d'acquisition des images, nous procédons à normaliser tous les blocs par rapport au bloc de référence \mathcal{B}_r . Appelons pour chaque bloc d'une même image \mathcal{B}_i sa moyenne μ_i et sa déviation standard σ_i . Pour chaque bloc d'une même image nous appliquons la formule suivante :

$$\mathcal{B}_i = \frac{(\mathcal{B}_i - \mu_i)}{\sigma_i} \times \sigma_M + \mu_M, \quad (6.3)$$

où μ_M et σ_M représentent respectivement la médiane des moyennes et la médiane des déviations standards de tous les blocs d'une image et du bloc de référence \mathcal{B}_r . Nous avons calculé la divergence de KL entre les modèles de chaque bloc des images choisies et les modèles du bloc de référence \mathcal{B}_r pour obtenir un ensemble de divergences de KL.

La tâche suivante consiste à classifier les divergences de KL obtenues en deux classes ; 1) la classe qui représente les blocs exempts de défauts et 2) la classe qui représente les blocs contenant des défauts. Deux alternatives se présentent. La première alternative consiste à utiliser un algorithme de classification par seuillage pour classifier les divergences et déterminer ainsi le seuil δ_d . Nous avons appliqué 3 algorithmes de classification par seuillage, qui sont : K-moyennes [12], Otsu [34] et Otsu médian [14]. Ces méthodes intègrent en leur sein un calcul automatique du seuil δ_d , ainsi que l'opération de classification en utilisant le seuil calculé. La deuxième alternative consiste à utiliser un classificateur probabiliste tel que le classificateur de Bayes. Le classificateur probabiliste n'utilise pas un seuil pour identifier l'appartenance d'une donnée à une classe, mais estime plutôt la distribution statistique de chaque classe en utilisant les données d'apprentissage puis, calcule les probabilités d'appartenance à chacune de ces classes. La donnée en question sera affectée à la classe ayant la probabilité la plus élevée. Cette option exige une phase d'apprentissage supervisé qui consiste à choisir à partir de la vérité terrain déjà construite des divergences qui représentent des blocs exempts de défauts et des divergences qui représentent des blocs contenant au moins un défaut et les marquer pour le classificateur probabiliste. Ainsi, le classificateur de Bayes se basera sur cet apprentissage supervisé pour décider lors de l'injection des données de test s'il s'agit d'un bloc sain ou défectueux. Pour ce faire, nous avons appliqué 2 variantes du classificateur de Bayes ; à savoir, le classifica-

teur de Bayes avec la loi normale, et le classificateur de Bayes avec les mélanges finis de Gaussiennes généralisées.

6.4.1 Seuillage des divergences de KL avec l’algorithme des K-moyennes

L’algorithme des K-moyennes [12] est un algorithme itératif de regroupement non supervisé de données par moyennes. En effet, cet algorithme permet de séparer les données en groupes selon leurs distances par rapport aux moyennes de ces groupes. Dans notre cas, nous avons 2 groupes de données ; 1) les blocs exempts de défauts et 2) les blocs contenant au moins un défaut. Nous avons appliqué l’algorithme des K-moyennes sur les divergences de KL et les résultats obtenus sont présentés dans le tableau suivant :

Transformée multirésolution	DWT		CT		RCT	
	128 × 128	64 × 64	128 × 128	64 × 64	128 × 128	64 × 64
Taille de bloc en pixels						
Niveau =1	89.72%	93.75%	93.89%	95.73%	93.89%	96.49%
Niveau =1 + approximation	90.42%	93.78%	93.89%	96.11%	93.89%	96.63%

Tableau 6.2 – Classification des divergences de KL avec l’algorithme des K-moyennes.

Trois conclusions sont tirées du Tableau 6.2. Premièrement, les performances des transformées en contourlettes dépassent celles des ondelettes (DWT) et les contourlettes redondantes (RCT) dépassent légèrement les contourlettes standards (CT). Deuxièmement, l’image d’approximation combinée avec le premier niveau de transformée multirésolution donne généralement de meilleurs résultats. Troisièmement, la taille de bloc 64 × 64 pixels est meilleure que la taille 128 × 128 pixels. Ceci s’explique par le poids du bloc par rapport à l’image entière. En effet, dans une image de tailles 768 × 512 pixels, s’il y a une erreur de classification d’un bloc de taille 128 × 128 pixels, l’erreur est de $1 \div 24 = 4.16\%$ alors

que dans le cas où la taille est de 64×64 pixels, l'erreur est de $1 \div 96 = 1.04\%$. D'ailleurs, l'utilisation de blocs de taille 64×64 pixels est visuellement meilleure, tel que montré dans Figure 6.5.

6.4.2 Seuillage des divergences de KL avec l'algorithme d'Otsu

La méthode de seuillage d'Otsu est très appliquée dans la segmentation d'images et plus précisément, lorsqu'il s'agit de séparer le fond du reste de l'information de l'image [34]. La méthode de seuillage d'Otsu est appliquée sur l'histogramme de l'image en assurant une grande variance interclasses. Il peut être utilisé dans la classification des données en deux classes du fait que la segmentation est un exemple de classification. Dans notre cas, nous avons construit l'histogramme des divergences de KL et nous avons appliqué la méthode de seuillage d'Otsu. Les résultats obtenus sont présentés dans le tableau suivant :

Transformée multirésolution	DWT		CT		RCT	
	Taille de bloc en pixels	Taille de bloc en pixels	Taille de bloc en pixels	Taille de bloc en pixels	Taille de bloc en pixels	Taille de bloc en pixels
Niveau =1	89.17%	93.99%	93.33%	94.17%	96.15%	96.91%
Niveau =1 + approximation	89.31%	94.03%	93.61%	95.69%	96.98%	97.05%

Tableau 6.3 – Classification des divergences de KL avec l'algorithme d'Otsu.

D'après Tableau 6.3, les performances des contourlettes redondantes (RCT) sont les meilleures. De plus, les contourlettes en général donnent un meilleur taux de détection que les ondelettes (DWT). Nous remarquons aussi que la taille des blocs 64×64 pixels est meilleure et que l'image d'approximation améliore remarquablement la détection de défauts tel que le montre Figure 6.6.

6.4.3 Seuillage des divergences de KL avec l’algorithme d’Otsu médian

La méthode de seuillage d’Otsu médian est une variante de la méthode de seuillage d’Otsu [14]. Otsu médian assure à son tour un maximum de séparation interclasse mais en utilisant la médiane de données au lieu de la moyenne. L’application de cette méthode de seuillage sur les divergences de KL nous a donné les résultats suivants :

Transformée multirésolution	DWT		CT		RCT	
Taille de bloc en pixels	128 × 128	64 × 64	128 × 128	64 × 64	128 × 128	64 × 64
Niveau =1	90.83%	94.17%	93.47%	97.33%	93.89%	97.36%
Niveau =1 + approximation	91.39%	94.27%	94.17%	97.40%	94.03%	97.40%

Tableau 6.4 – Classification des divergences de KL avec l’algorithme d’Otsu médian.

Les résultats présentés dans Tableau 6.4 confirment les conclusions tirées dans la sous-section 6.4.1 et la sous-section 6.4.2. Les transformées en contourlettes (standards (CT) et redondantes (RCT)) donnent des résultats meilleurs que la transformée en ondelettes (DWT). Ceci peut s’expliquer par la richesse de la sélectivité directionnelle dans les contourlettes. En effet, dans le chapitre 3, nous avons expliqué que les transformées en contourlettes (standards et redondantes) permettent la localisation de l’information dans différentes directions grâce aux filtres bidimensionnels directionnels alors que la transformée en ondelettes se limite à 3 directions. De plus, ces filtres bidimensionnels directionnels permettent la détection des contours lisses, ce qui permet d’améliorer la discrimination de texture et la détection des défauts. Nous remarquons aussi que, généralement, la transformée en contourlettes redondantes dépasse légèrement la transformée en contourlettes standards. Ceci est dû essentiellement à l’utilisation des filtres pseudo-Gaussiens dans la transformée en contourlettes redondantes. De plus, la transformée en contourlettes

redondantes offre plus de précision aux détails exhibés dans les sous-bandes grâce à la redondance d'information dans les sous-bandes. En effet, toutes les opérations de sous-échantillonnage sont éliminées dans le système pyramidal Laplacien, produisant ainsi des sous-bandes de même taille. Nous remarquons aussi que la taille des blocs 64×64 pixels résulte en une meilleure localisation de défauts tel que présenté dans Figure 6.7

6.4.4 Classification des divergences de KL avec le classificateur de Bayes

Nous avons expliqué dans le début de la section 6.4 que l'application d'un classificateur supervisé exige une phase d'apprentissage. Dans cette phase, le classificateur en question apprend avec des données correctement classifiées. En d'autres termes, il faut introduire pour le classificateur des divergences de KL représentant des blocs exempts de défauts et des divergences de KL représentant des blocs contenant au moins un défaut. De plus, pour chacune de ces divergences, il faut préciser le groupe de données auquel elle appartient. Cette tâche est supervisée et nécessite la possession de la vérité terrain des données en question. Ensuite, il faut préciser le modèle statistique qui constitue la base de la classification. Notons aussi qu'il faut introduire suffisamment de données au classificateur afin d'obtenir de meilleurs résultats. Nous avons expérimenté deux classificateurs de Bayes avec deux modèles différents ; le premier se base sur la loi normale et le second se base sur le modèle de mélanges finis de Gaussiennes généralisées. La spécification du modèle du classificateur est une étape essentielle. En effet, dans la phase d'apprentissage du classificateur, il faut préciser le modèle statistique suivi par les données de chaque classe. Dans le chapitre 5, nous avons présenté et expliqué la modélisation statistique de données. Le classificateur utilise les paramètres des modèles des classes pour calculer les probabilités d'appartenance d'une donnée (dans notre cas une divergence de KL) à

chacune de ces classes. Une fois les probabilités sont calculées, le classificateur affecte la donnée en question à la classe correspondante. Pour améliorer les performances de la phase d'apprentissage du classificateur ainsi que la détection de défauts, l'algorithme peut avoir recours à un ou plusieurs blocs de référence. En effet, à la fin de chaque itération de la phase d'apprentissage, l'algorithme calcule le taux d'erreurs de détection et le compare avec celui de l'itération précédente. Si le taux d'erreur de détection est plus petit et non nul, l'algorithme cherche un nouveau bloc de référence. Le nouveau bloc de référence doit être un bloc exempt de défaut et qui a une divergence de KL maximale par rapport au bloc de référence de l'itération précédente. Si le taux d'erreur de détection est plus grand de celui de l'itération précédente l'algorithme s'arrête et retourne les blocs de référence retenus aux itérations précédentes. L'apprentissage du classificateur de Bayes est structuré dans l'algorithme 3. Nous avons utilisé les 30 images dont nous avons la vérité terrain, tel qu'expliqué dans la Section 6.3. Nous avons fixé manuellement un bloc exempt de défauts comme un bloc de référence. Ensuite, nous avons estimé les modèles MoGG des sous-bandes de la transformée en contourlettes redondantes de chaque bloc. Nous avons calculé la similarité des modèles de chaque bloc avec les modèles du bloc de référence pour obtenir les divergences de KL. Nous avons divisé ces divergences ainsi que leur vérité terrain correspondante en deux sous-ensembles ; 1) Ensemble d'apprentissage du classificateur et 2) Ensemble de test. Nous avons introduit l'ensemble d'apprentissage au classificateur et nous avons précisé son modèle ; 1) Loi normale (Gaussienne) ou 2) Mélanges finis de Gaussiennes généralisées (MoGG). Nous avons utilisé le classificateur pour classifier l'ensemble de test. L'étape suivante consiste à calculer le taux d'erreurs de détection et sélectionner le nouveau bloc de référence, le cas échéant, ou arrêter l'itération de l'algorithme. L'arrêt des itérations est effectué dans deux cas possibles ; le premier est lorsque le taux d'erreurs d'une itération est supérieur à celui de l'itération précédente et le second est lorsque le taux d'erreurs est nul (cas optimal).

Algorithm 3 *Algorithme d'apprentissage du classificateur de Bayes.*

Input : Ensemble d'images, Vérité terrain associée, Modèle du classificateur (Loi normale || MoGG).

Output : L'ensemble des Blocs de référence, le classificateur de Bayes.

1. Diviser les images en blocs ; Appliquer la transformée RCT ; Estimer les modèles MoGG ;
2. Choisir un bloc de référence (\mathcal{B}_1) ; initialiser $i = 1$;
3. Calculer les divergences de KL (\mathcal{KL}_i) entre le bloc de référence (\mathcal{B}_1) et les blocs des images ;
4. Construire le modèle du classificateur de Bayes (en utilisant les divergences de KL et la vérité terrain associée) ;
5. Classifier les divergences de KL (\mathcal{KL}_i) ; calculer le Taux d'erreurs(i) ;

if (Taux d'erreurs(i) \neq 0) **then**

repeat

 6. $i = i + 1$;

 7. Chercher le nouveau bloc de référence \mathcal{B}_i (Bloc sain détecté comme défectueux et ayant la divergence de KL maximale) ;

 8. Calculer les divergences de KL (\mathcal{KL}_i) entre le bloc de référence (\mathcal{B}_i) et les blocs des images.

 9. Calculer pour chaque bloc la divergence de KL minimale ($\mathcal{KL}_{min} = \min\{\mathcal{KL}_j\}_{j=1..i}$) ;

 10. Classifier les divergences de KL minimales (\mathcal{KL}_{min}) ; calculer le Taux d'erreurs(i) ;

until (Taux d'erreurs(i) \geq Taux d'erreurs(i - 1) || Taux d'erreurs(i) = 0).

end if

if (Taux d'erreurs(i) \neq 0) **then**

 11. Retourner les blocs de référence distincts $\{\mathcal{B}_1 \dots \mathcal{B}_{i-1}\}$ et le classificateur de Bayes ;

else

 12. Retourner les blocs de référence distincts $\{\mathcal{B}_1 \dots \mathcal{B}_i\}$ et le classificateur de Bayes ;

end if

Dans le tableau suivant, nous présentons la classification des données d'apprentissage avec deux classificateurs de Bayes : le premier utilise le modèle de la loi normale et le second utilise le modèle de mélanges finis de Gaussienne généralisées (MoGG).

Modèle du classificateur	MoGG(a)	MoGG(b)	Loi normale
Blocs défectueux	782	782	782
Blocs sains	782	9568	9568
Nb de blocs de référence retenus / Nb de blocs de référence calculés lors de l'apprentissage	2 / 3	1 / 2	1 / 2
Erreurs	[478 ; 458 ; 463]	[442 ; 724]	[289 ; 454]
Meilleur taux d'erreurs	4.42%	4.27%	2.79%

Tableau 6.5 – Statistiques de taux d'erreurs lors de la phase d'apprentissage des classificateurs de Bayes avec différents modèles statistiques.

Dans Tableau 6.5, la colonne MoGG(a) exprime les résultats de la classification de Bayes avec le modèle MoGG en utilisant un échantillon de 1564 données (divergences de KL), alors que la colonne MoGG(b) exprime les résultats de classification de Bayes avec le modèle MoGG en utilisant un échantillon de 10350 données. Il est clair dans la comparaison de deux colonnes que le meilleur taux d'erreurs est celui de la colonne MoGG(b). Ceci confirme que l'utilisation d'un grand nombre de données dans l'apprentissage influence positivement les résultats de classification. Il est à noter aussi dans ce contexte que l'ensemble de données de test peut inclure les données utilisées dans la phase d'apprentissage qui est d'ailleurs le cas pour la colonne MoGG(b). Nous remarquons aussi que le classificateur de Bayes avec la loi normale est meilleur que ceux qui utilisent le modèle MoGG, que ce soit en taux d'erreurs ou en nombre de blocs de référence qui seront utilisés ultérieurement dans la détection de défauts dans les images de test. Le classificateur de Bayes avec la loi normale donne un taux de détection de 97.21 % avec l'utilisation d'un seul bloc de référence. Rappelons que l'utilisation de moins de blocs de référence donne un meilleur temps d'exécution. Pour mieux exprimer les résultats exposés dans le Tableau 6.5, nous proposons le graphe suivant :

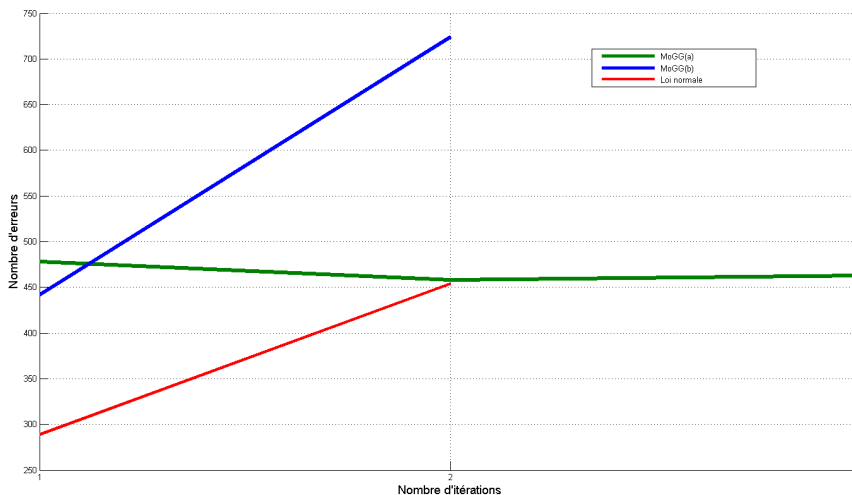


Figure 6.3 – Nombre d’erreurs des différents classificateurs de Bayes versus le Nombre d’itérations lors de la phase d’apprentissage.

La Figure 6.3 montre que le meilleur classificateur de Bayes est celui qui est basé sur la loi normale. En effet, ce dernier donne le meilleur taux d’erreurs de classification avec le minimum de blocs de référence, assurant ainsi le meilleur temps d’exécution. Nous présentons dans la Figure 6.4 la modélisation des divergences de KL tel que présenté dans Tableau 6.5. D’après la Figure 6.4, il est clair que le meilleur modèle qui permet de représenter les divergences de KL obtenues est le modèle de la loi Normale. Ceci s’explique essentiellement par le fait que les divergences de KL qui représentent les blocs exempts de défauts suivent une distribution normale.

Après avoir récupéré les blocs de référence ainsi que le classificateur avec le modèle spécifié, nous appliquons ce classificateur pour classifier les blocs du reste des images appartenant à la même famille de texture de textile. Nous avons pris chaque image qui appartient à la famille de texture de textile en question et nous l’avons subdivisée en blocs. Ensuite, nous avons appliqué à chaque bloc, la transformée RCT suivie de la

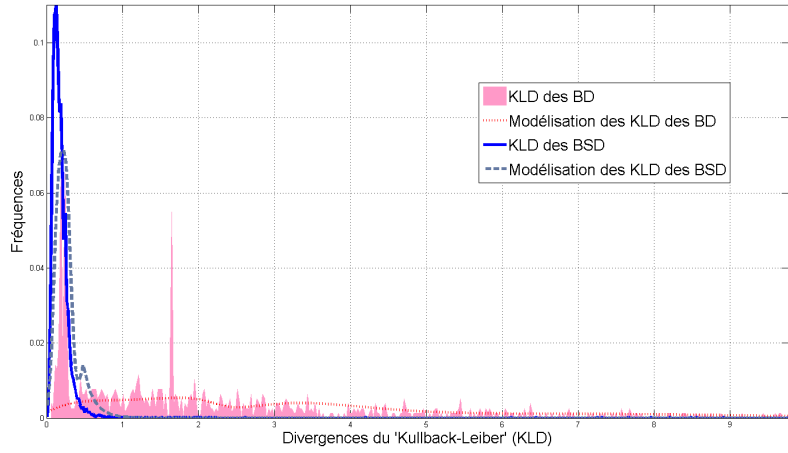
modélisation MoGG. Puis, nous avons calculé la divergence de KL entre chaque bloc de l'image et et chacun des blocs de référence retournés par l'algorithme 3. Nous choisissons la divergence minimale de KL pour représenter le bloc en question. Enfin, nous appliquons le classificateur de Bayes pour classifier les divergences de KL minimales calculées. Ainsi, nous obtenons une image classifiée. Nous présentons dans ce qui suit l'algorithme de détection de défauts avec le classificateur de Bayes sur les images de test.

Algorithm 4 *Algorithme de détection de défauts basée sur la modélisation MoGG des sous-bandes de la transformée RCT et le classificateur de Bayes.*

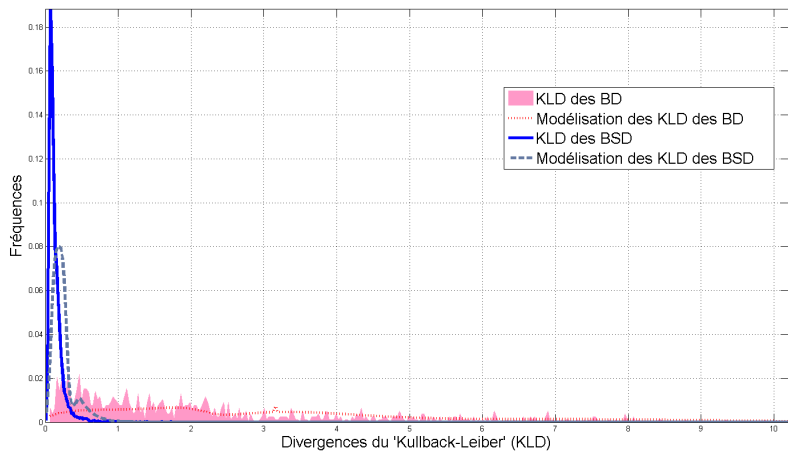
Input : Image de test de la même famille que l'ensemble des images d'apprentissage, le classificateur de Bayes de l'algorithme 3 et ses n blocs de référence $\{\mathcal{B}_1 \dots \mathcal{B}_n\}$.

Output : image de test classifiée.

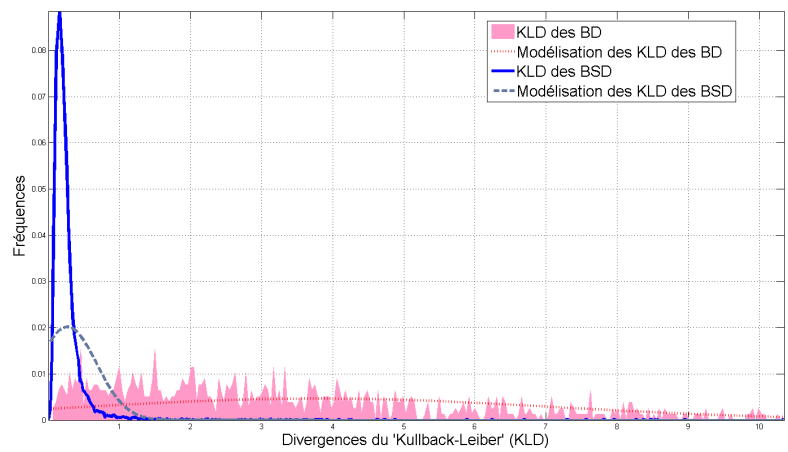
1. Diviser l'image de test en blocs ; Appliquer la transformée RCT ; Estimer les modèles MoGG ;
 2. Calculer les divergences de KL entre chaque bloc de l'image de test et chacun des blocs de référence $\{\mathcal{B}_1 \dots \mathcal{B}_n\}$;
 3. Calculer pour chaque bloc de l'image de test la divergence de KL minimale $\mathcal{KL}_{min} = \min\{\mathcal{KL}_j\}_{j=1..n}$;
 4. Classifier les divergences de KL (\mathcal{KL}_{min}) en utilisant le classificateur de Bayes ;
-



(a) MoGG(a)



(b) MoGG(b)



(c) Loi Normale

Figure 6.4 – Modélisations des divergences de KL correspondantes au Tableau 6.5.

6.5 Influence de la taille des blocs

En se référant aux Tableaux 6.2, 6.3 et 6.4, nous remarquons que l'utilisation de blocs de taille 64×64 pixels est meilleure que l'utilisation des blocs de taille 128×128 pixels. En effet, les blocs de taille plus petite permettent de mieux cerner le défaut en question, et produisent ainsi une meilleure localisation des défauts. Cependant, la taille des blocs doit être supérieure à la taille des filtres utilisés dans les transformées multirésolutions, sinon les résultats seront altérés du fait que les transformées multirésolutions traiteront les informations de bord des blocs. D'ailleurs, la taille des filtres utilisés par les transformées multirésolutions influence le nombre de niveaux de décomposition. Les meilleures performances ont été obtenues par l'utilisation du premier niveau de décomposition combiné avec l'image d'approximation. Dans Figures 6.5, 6.6 et 6.7, nous montrons quelques exemples qui illustrent les conclusions tirées dans les sections 6.4 et 6.5.

Sur le plan visuel, on peut remarquer dans les exemples présentés dans Figures 6.5, 6.6 et 6.7, que l'efficacité de détection des contourlettes dépasse largement celle des ondelettes et que les contourlettes redondantes donnent de meilleurs résultats que les contourlettes standards. De plus, la taille de bloc plus petite donne une meilleure localisation du défaut. Notons que la taille 64×64 pixels est la taille minimale avec laquelle nous pouvons procéder, du fait que les filtres des contourlettes et des contourlettes redondantes sont de la taille 49×49 pixels.

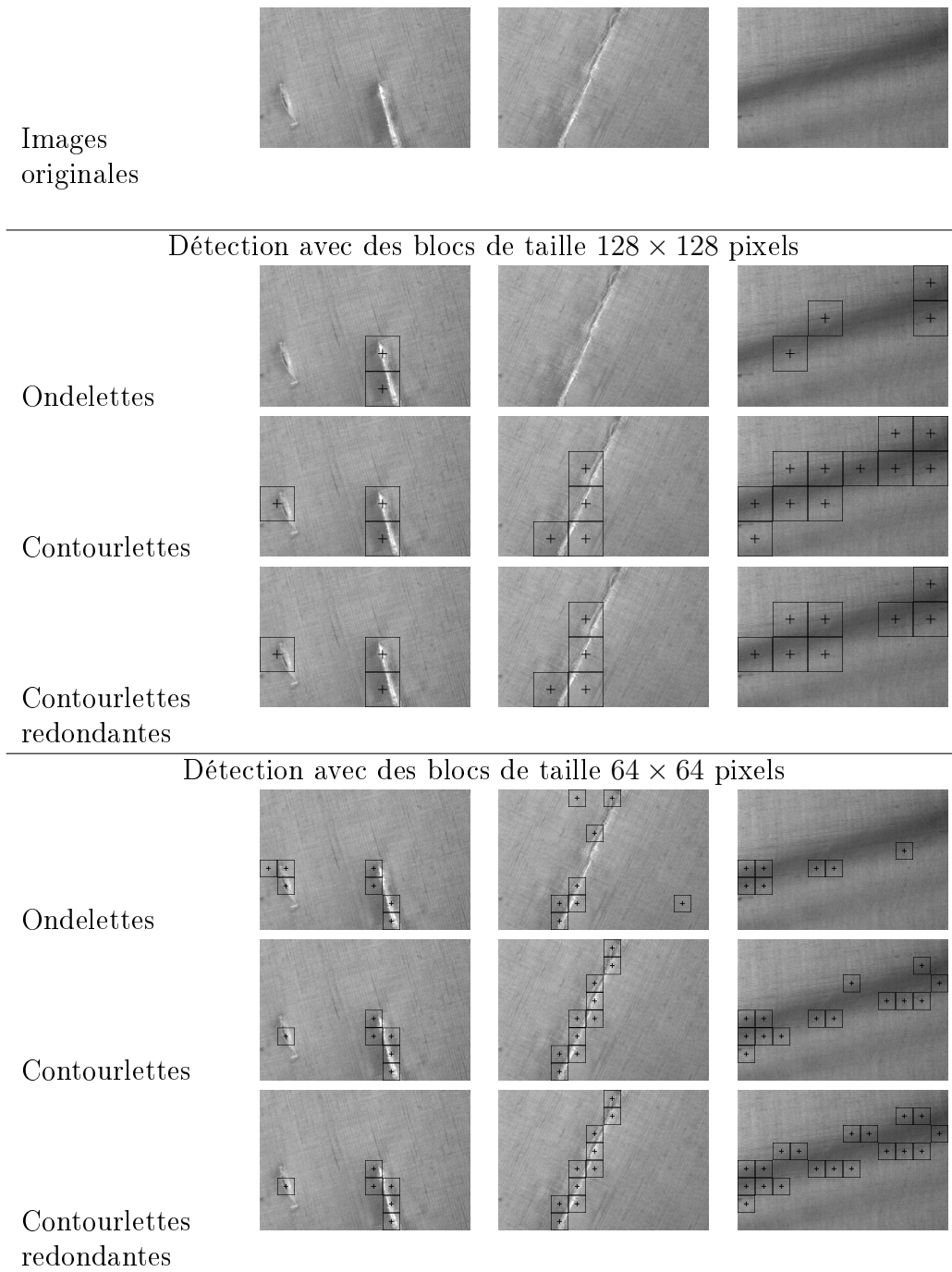


Figure 6.5 – Exemples de détection de défauts avec la méthode de classification par seuillage de K-moyennes.

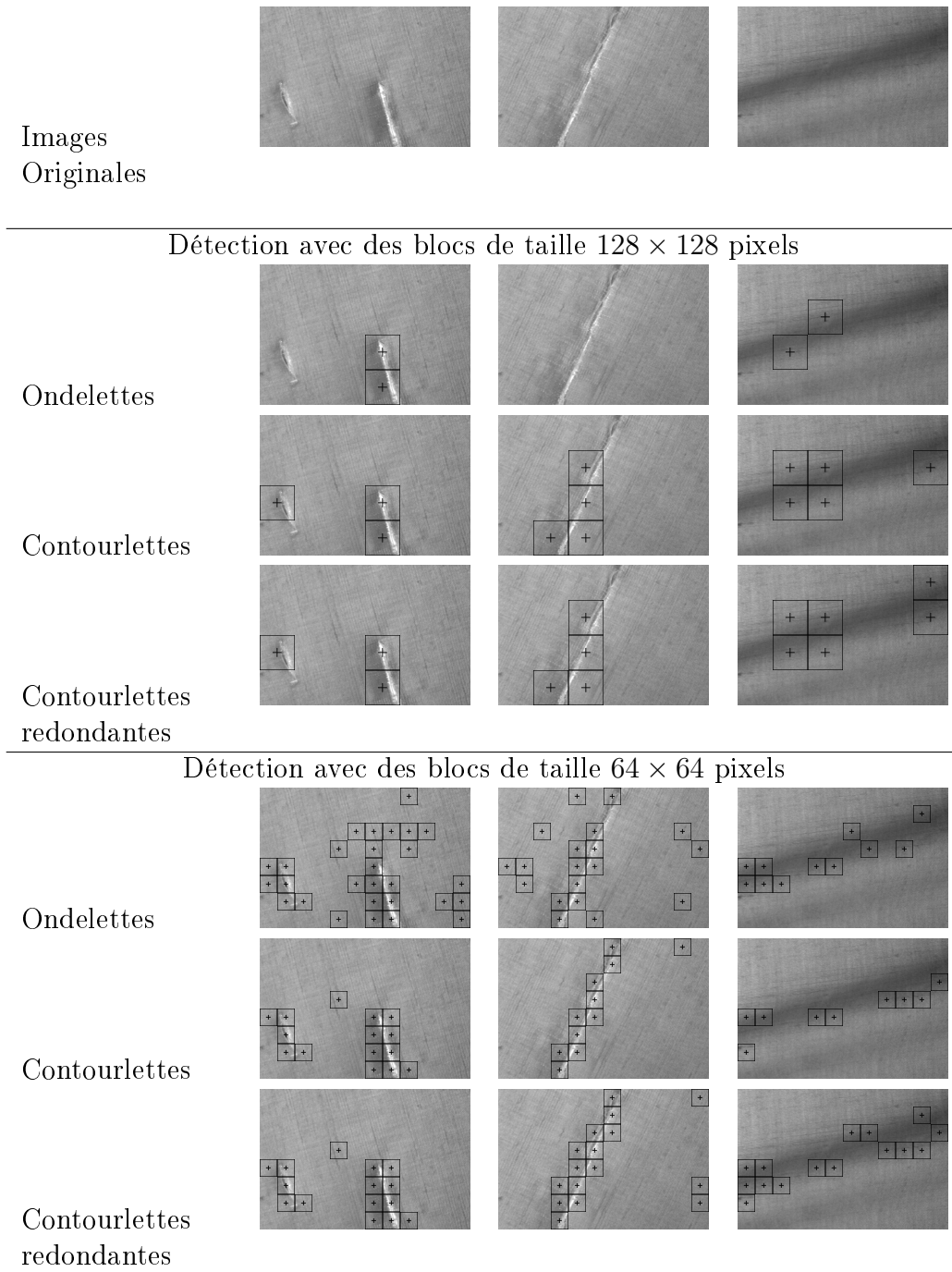


Figure 6.6 – Exemples de détection de défauts avec la méthode de classification par seuillage d’Otsu.

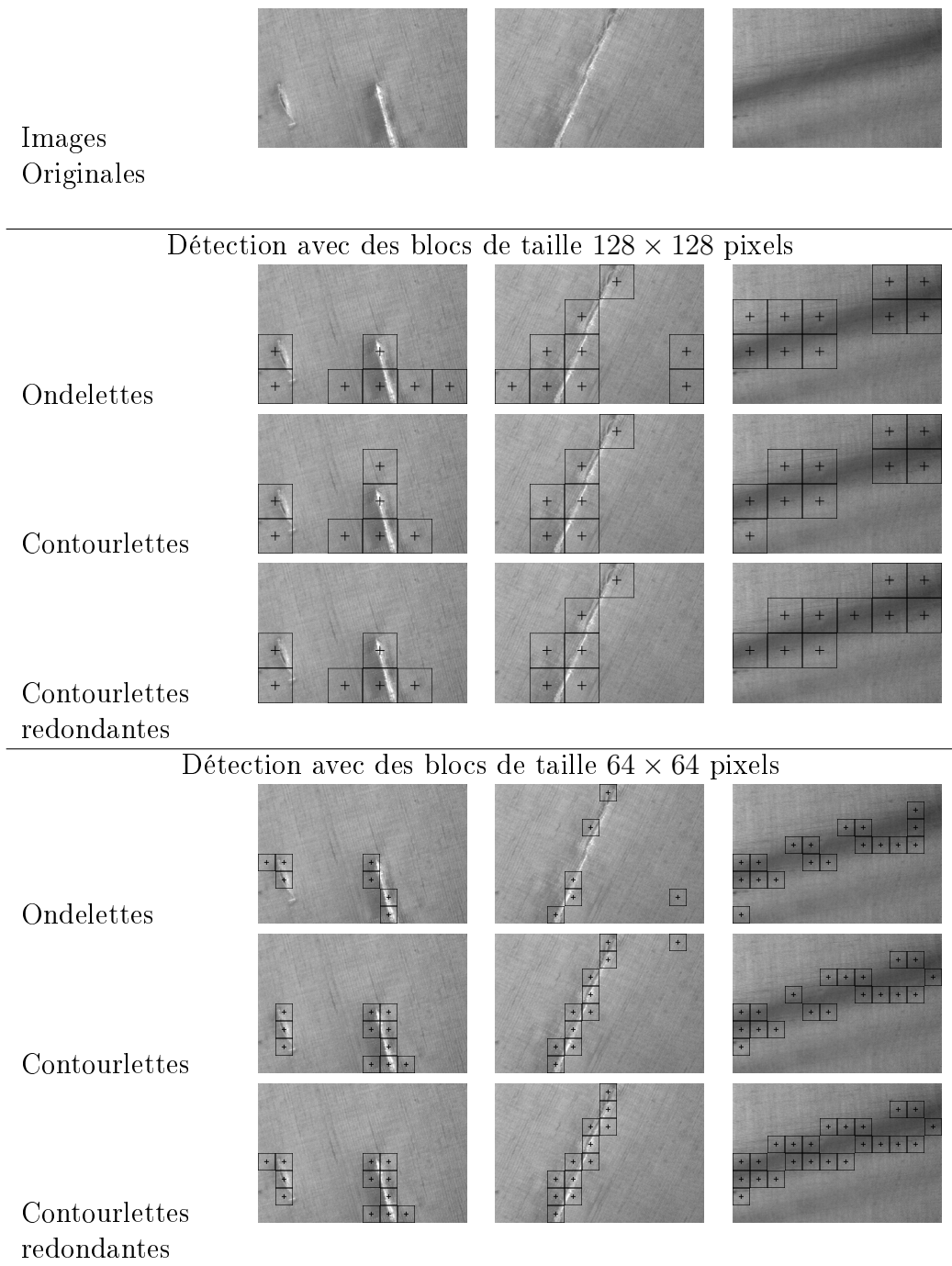


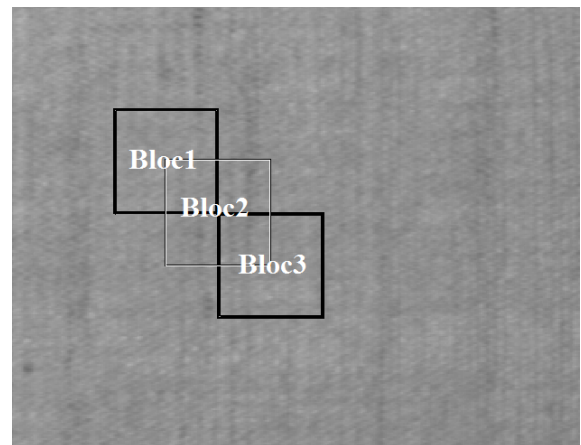
Figure 6.7 – Exemples de détection de défauts avec la méthode de classification par seuillage d’Otsu médian.

6.6 Traitement par chevauchement de blocs

Pour augmenter l'efficacité de notre méthode en termes de taux de détection de défauts, nous procédons par le traitement des images par chevauchement de blocs. En effet, le chevauchement de blocs permet de traiter le quart d'un bloc d'image 4 fois dans 4 voisinages différents. Bien que ce concept implique la redondance, il assure une décision plus exacte concernant les différentes parties composantes d'un bloc, ainsi qu'une meilleure localisation de défauts. Figure 6.8 explique le concept de traitement par chevauchement de blocs et montre comment il permet une meilleure localisation de défauts.

1	2	2	2	2	2	2	2	2	2	2	1
2	4	4	4	4	4	4	4	4	4	4	2
2	4	4	4	4	4	4	4	4	4	4	2
2	4	4	4	4	4	4	4	4	4	4	2
2	4	4	4	4	4	4	4	4	4	4	2
2	4	4	4	4	4	4	4	4	4	4	2
2	4	4	4	4	4	4	4	4	4	4	2
1	2	2	2	2	2	2	2	2	2	2	1

(a) Matrice de chevauchement

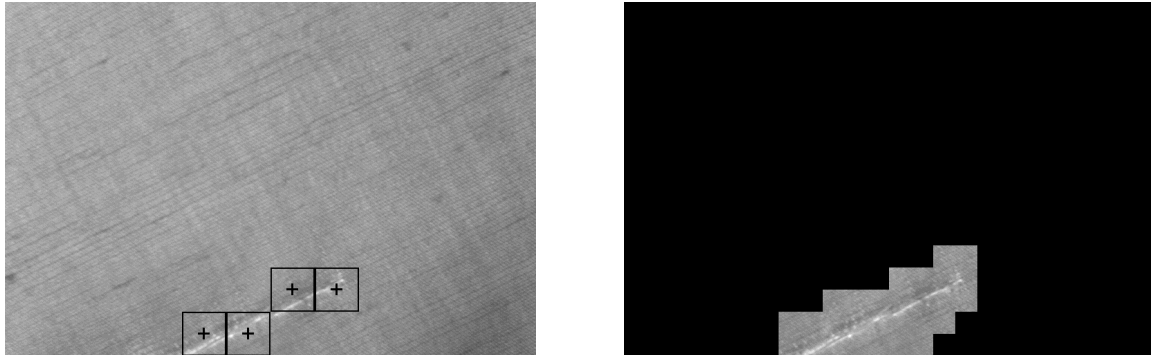


(b) Application sur une image de texture

Figure 6.8 – Schéma de traitement d'image par chevauchement de blocs.

Nous remarquons que les blocs des quatre coins de l'image sont traités une seule fois alors que les blocs des bords sont traités deux fois. Les blocs de l'intérieur sont traités chacun 4 fois mais dans 4 blocs différents. Nous avons expliqué que dans le cas de la présence d'un bout de défauts dans les bords d'un bloc est difficile à détecter du fait qu'il n'exhibe pas remarquablement les fréquences dans les sous-bandes des transformées multirésolutions. Le traitement par chevauchement ramène une partie du défaut le rendant ainsi détectable par les transformées multirésolutions. Dans Figure 6.9, nous présentons la détection de

défauts avec et sans chevauchement pour la même image.



(a) Détection sans chevauchement de blocs (b) Détection avec chevauchement de blocs

Figure 6.9 – Exemple de traitement d’image avec et sans chevauchement de blocs de taille 64×64 pixels.

En regardant la partie (a) de Figure 6.9, nous remarquons qu’une partie du défaut n’a pas été détectée. Ceci est dû essentiellement à la présence de certaines parties du défaut dans les bords des blocs avec de petites proportions. Le traitement par chevauchement de blocs permet de résoudre ce problème tel que le montre la partie (b) de la même figure.

6.7 Comparaison avec la méthode de détection basée sur l’ACI

Les résultats obtenus par la méthode de détection de défauts de texture basée sur la modélisation MoGG et les contourlettes redondantes sont mieux que les résultats obtenus par la méthode de détection de défauts de texture basée sur l’ACI. En effet, dans [57], les auteurs ont obtenu un taux de détection de défauts de 96.74%. Dans [56], les mêmes auteurs ont intégré la transformée en ondelettes discrètes pour avoir une détection de défauts basée sur l’ACI des sous-bandes ondelettes des blocs des images de textile. Dans

ces travaux, plusieurs variantes ont été implémentées et les études empiriques ont montré des taux de détections inférieurs à 97%, alors que notre méthode basée sur les contourlettes redondantes et la modélisation MoGG combinée avec le classificateur par seuillage d'Otsu médian, a atteint un taux de détection de 97.40%.

Dans [58], les auteurs ont utilisé le traitement par chevauchement pour réduire les taux d'erreur de détection de leur méthode basée sur l'ACI. Dans leur étude expérimentale, les auteurs n'ont pas spécifié le taux de détection de défauts, mais ils ont établi une comparaison en utilisant différentes mesures d'erreurs de détection. Pour la classe des images exemptes de défauts, ils ont calculé le nombre de blocs qui ont été classés défectueux. Ainsi, le taux d'erreurs de détection est facilement calculé du fait que le nombre total de blocs d'une image est un paramètre connu. Pour les images contenant des défauts, ils ont calculé le nombre d'images où les défauts n'ont pas été détectés. Nous avons utilisé le principe de chevauchement dans notre méthode de détection afin d'améliorer les résultats de détection et pouvoir se comparer à la méthode de détection de défauts basée sur l'ACI. Pour ce faire, nous avons construit deux tableaux ; le premier présente les résultats de la détection de défauts de texture basée sur l'ACI [58] et le second présente les résultats de la détection de défauts de texture basée sur la modélisation par les mélanges finis de Gaussiennes généralisées des sous-bandes de la transformée en contourlettes redondantes et le classificateur de Bayes.

Classe Tilda C1R1	E0	E1	E2	E3	E4
Taux de détection en terme d'image	0/50 (0%)	50/50 (100%)	50/50 (100%)	48/50 (96%)	49/50 (98%)
Taux de détection en terme de bloc	17123/17250 (99.61%)	-	-	-	-

Tableau 6.6 – Statistiques de taux de détection basée sur l'ACI.

Classe Tilda C1R1	E0	E1	E2	E3	E4
Taux de détection en terme d'image	41/50 (82%)	48/50 (96%)	50/50 (100%)	45/50 (90%)	45/50 (90%)
Taux de détection en terme de bloc	17240/17250 (99.94%)	-	-	-	-

Tableau 6.7 – Statistiques de taux de détection basée sur les contourlettes redondantes et le modèle MoGG.

D'après Tableaux 6.6 et 6.7, on peut noter que la méthode basée sur les contourlettes redondantes et le modèle MoGG agit mieux sur les images exemptes de défauts. En effet, l'algorithme a mal classé 10 blocs sur un ensemble de 17250 blocs et qui sont répartis sur 9 images alors que l'algorithme de détection basé sur l'ACI a fait 67 erreurs réparties sur toutes les images exemptes de défauts. Pour les images contenant des défauts, l'algorithme de détection de défauts basé sur les contourlettes redondantes et le modèle MoGG s'est trempé sur 12 images (sur un total de 200 images défectueuses), alors que l'algorithme basé sur l'ACI s'est trempé uniquement sur 3 images. Notons dans ce contexte que méthode de détection de défauts basée sur les contourlettes redondantes et le modèle MoGG s'est trempée sur des images où le(s) défaut est mineur. Nous présentons dans Figure 6.10 les images où notre méthode de détection de défauts n'a pas détecté les défauts et a considéré que l'image est totalement saine. Nous remarquons d'après la figure que les défauts sont lisses et/ou de proportion petite et par conséquent, sont difficiles à détecter même par l'œil humain.

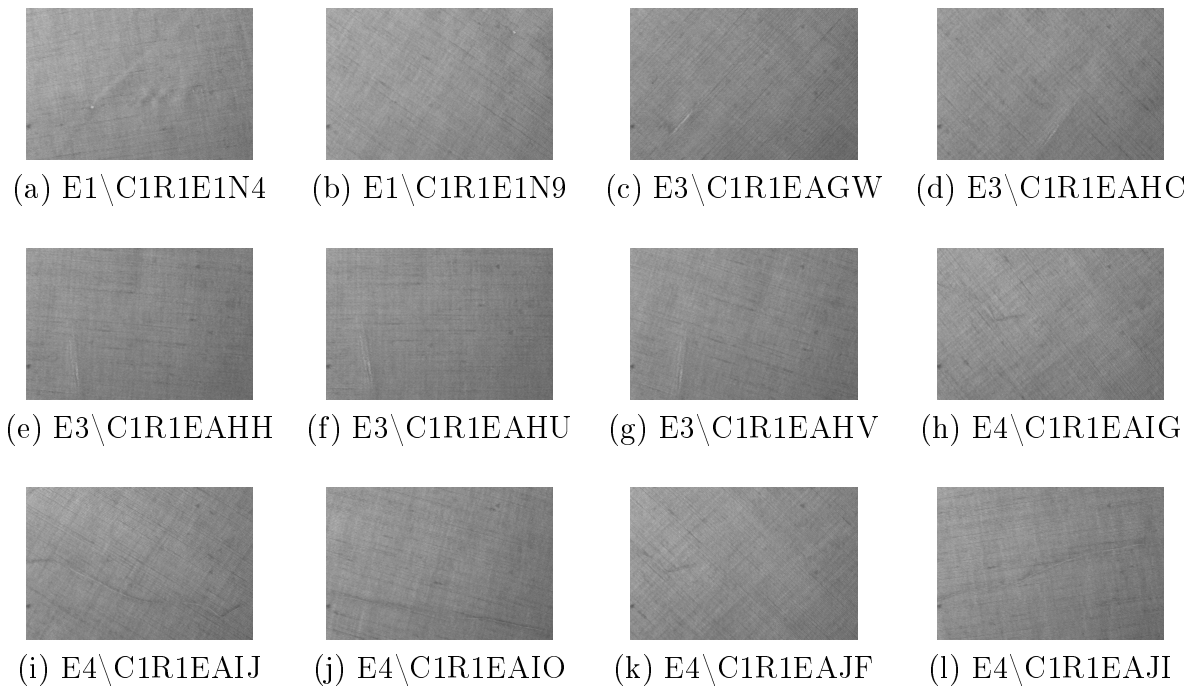


Figure 6.10 – Les 12 images défectueuses et mal détectées par l’algorithme basé sur la transformée en contourlettes redondantes et la modélisation par les mélanges finis de Gaussiennes généralisées.

Si nous comparons le taux de réussite total en terme d’images, nous trouvons que la méthode de détection de défauts basée sur les contourlettes redondantes et le modèle MoGG réalise un taux de détection de $((41 + 48 + 50 + 45 + 45)/250 = 91.6\%)$ alors que la méthode de détection de défauts basée sur l’ACI réalise un taux de détection de $((0+50+50+48+49)/250 = 78.8\%)$. Il est à noter aussi que la détection correcte des blocs exempts de défauts est plus importante dans l’industrie de textile du fait que la fréquence de défauts est généralement petite alors que la fréquence de la production du tissu sain est beaucoup plus élevée. De ce fait, l’utilisation d’une méthode de détection qui fait le moins d’erreurs dans la détection des tissus sains est meilleure dans le domaine industriel. D’ailleurs, la méthode de détection de défauts basée sur les contourlettes redondantes et le modèle MoGG réalise un taux de détection de tissu sain en terme de blocs de

$((17240/17250) = 99.94\%)$.

Nous présentons dans Figure 6.11 quelques exemples de la détection de défauts de texture dans les tissus de textile en utilisant les deux méthodes : 1) la détection basée sur l'ACI et 2) la détection basée sur la transformée RCT, la modélisation MoGG et le classificateur de Bayes avec la loi normale. Notons que la détection pour les deux méthodes est avec des chevauchement de blocs de taille 64×64 pixels.



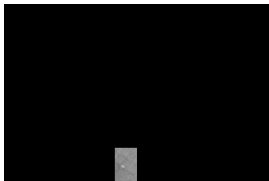
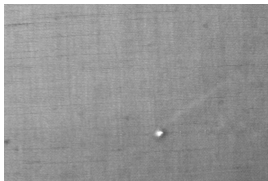
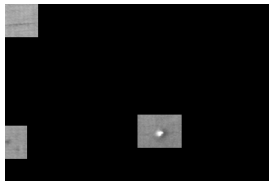
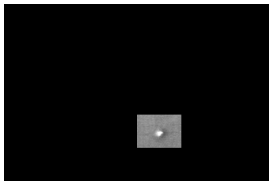



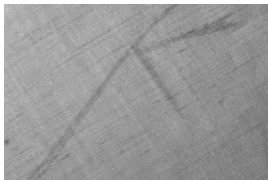


Nom de l'image	Image Originale	Détection basée sur l'ACI	Détection basée sur le MoGG, RCT et le classificateur de Bayes avec la loi normale
C1R1E1N8			
C1R1EABS			
C1R1EAEA			
C1R1EAEH			

Figure 6.11 – Comparaison de la détection de défauts de notre méthode avec la détection de défauts de la méthode de référence.

6.8 Analyse de la complexité algorithmique

L'algorithme de modélisation par les mélanges finis de Gaussiennes généralisées [4] utilise l'algorithme de expectation-maximisation [7] pour l'estimation des paramètres de K composantes d'un mélange en utilisant N données. L'ensemble des deux étapes expectation et maximisation a une complexité de $O(KN)$. L'initialisation des paramètres de K composantes du modèle est effectuée par l'algorithme de K-moyennes flou qui a à son tour une complexité de $O(KN)$. Alors, l'algorithme de modélisation a une complexité de $O(KN)$. La comparaison de modèles est effectuée par le calcul de similarité basé sur la formule de divergence de KL. Nous avons expliqué dans la section 5.2.3 que cette formule n'est pas applicable pour les mélanges finis de Gaussiennes généralisées et que la solution consiste à l'approximer par échantillonnage. La méthode d'échantillonnage utilisée est celle de Monte Carlo [52]. En supposant que le nombre d'échantillons est n , la complexité de l'algorithme de calcul de similarité est de $O(LKn)$ où L est le nombre de niveaux de la décomposition multirésolution. Ainsi, l'algorithme de détection de défauts de texture basé sur la modélisation MoGG et les transformée multirésolutions a une complexité de $O(BLKn)$ où B est le nombre de blocs d'une image.

Dans le même contexte, nous notons qu'en utilisant une station de travail avec la configuration citée dans la section 6.2, l'estimation des paramètres des modèles des sous-bandes d'un bloc d'image de taille 64×64 pixels coûte 0.9 seconde et environs 2 secondes pour un bloc d'image de tailles 128×128 pixels. Le calcul de la divergence de KL entre les modèles d'un bloc d'une image avec les modèles du bloc de référence, avec la considération de 10000 échantillons, coûte 1 seconde. Le temps total de traitement d'un bloc de taille 64×64 pixels est environ 2 secondes et pour un bloc de taille 128×128 pixels est 3 secondes. Le temps de traitement total d'une image avec chevauchement de blocs de taille 64×64 pixels est environs 10 minutes. Notons dans ce contexte que nous avons

opté pour le choix de traitement par chevauchement de blocs de taille 64×64 pixels pour avoir une meilleure localisation de défauts ainsi qu'un meilleur taux de détection.

6.9 Sommaire du chapitre

Dans ce chapitre, nous avons présenté la méthode de détection de défauts de texture basée sur la transformée en contourlettes redondantes et la modélisation par les mélanges finis de Gaussiennes généralisées. Nous avons présenté également les études empiriques des différentes méthodes de classification des divergences de KL ainsi que différents classificateurs de Bayes. Les études empiriques ont montré que la meilleure classification est effectuée avec le classificateur de Bayes basé sur le modèle de la loi normale. Nous avons montré que le traitement par chevauchement de blocs donne une meilleure localisation des défauts. Nous avons également montré l'influence de la taille des blocs sur les résultats de détection. D'un autre côté, nous avons comparé notre méthode avec la méthode de référence qui est basée sur l'ACI [58]. Nous avons conclu que la méthode de détection de défauts basée sur l'analyse en composantes indépendante est une méthode efficace. Cependant, cette méthode est très sensible et classe un bloc comme défectueux pour le moindre défaut et commet beaucoup d'erreur lors de la détection des tissus sains alors qu'il faut considérer que le milieu industriel est un milieu bruyant et l'acquisition de l'image peut être légèrement altérée par ces bruits. Ceci induit cette méthode en erreur de détection de blocs exempts de défauts. Notre méthode de détection basée sur les contourlettes redondantes, le modèle MoGG et le classificateur de Bayes résout ce problème et est meilleure dans la détection des blocs exempts de défauts.

Chapitre 7

Conclusion générale

L'élaboration d'un outil automatisé de détection de défauts est devenue une nécessité pour l'industrie de textile. Dans ce mémoire, nous avons présenté dans l'état de l'art les différents types d'inspection de textile et nous avons montré que l'inspection automatique est possible. La détection de défauts basée sur l'ACI est une méthode efficace et rapide pour la détection et la localisation des défauts de textiles. Cependant, cette méthode est très sensible et génère beaucoup de fausses alertes en détectant des tissus exempts de défauts comme étant défectueux.

Les contributions apportées dans ce mémoire se présentent comme suit :

1. La ré-implémentation de la méthode de détection de défauts de texture dans les tissus de textiles basée sur l'ACI dans ses deux versions [56, 57]. Nous avons proposé par la même occasion des améliorations qui peuvent être apportées à cette méthode pour combler ses limites.
2. Proposition d'une nouvelle méthode de détection de défauts de texture dans les tissus de textiles basée sur les transformées multirésolutions et la modélisation par les

mélanges finis de Gaussiennes généralisées. Notre méthode extrait les paramètres des modèles des sous-bandes de la transformée multirésolution d'un bloc d'image pour en produire sa signature compacte, ensuite compare ces modèles avec ceux d'un bloc de référence classé comme exempt de défauts. Cette comparaison est exprimée par la divergence de KL. Nous avons appliqué différentes méthodes de classification sur les distances de KL afin de déterminer la méthode la plus appropriée. D'un autre côté, nous avons utilisé 3 différentes transformées multirésolutions pour déterminer celle la plus appropriée à notre problématique. L'étude empirique détaillée dans le chapitre 6 a montré que la transformée en contourlettes redondantes (RCT) est la transformée multirésolution qui donne le meilleur rendement, grâce à la redondance d'informations dans les sous-bandes et les filtres pseudo-Gaussiens dans le système pyramidal Laplacien qui sont mieux appropriés à la discrimination de texture. Nous avons montré aussi que le classificateur de Bayes avec le modèle de la loi normale est le plus approprié pour classifier efficacement les divergences de KL en permettant un nombre minimum d'erreurs et un nombre minimum de blocs de références. Nous avons expérimenté différentes tailles de blocs ainsi que le traitement avec et sans chevauchement de blocs. La taille des blocs 64×64 pixels est la taille minimale que nous pouvons utiliser dans notre algorithme. Pour avoir plus de précision et une meilleure localisation, le traitement par chevauchement de bloc permet de contourner la limite de la taille de blocs pour avoir une localisation de défaut à une taille de 32×32 pixels.

3. Comparaison des résultats de la nouvelle méthode proposée avec les résultats des méthodes de l'état de l'art. Nous avons montré que notre méthode dépasse la méthode de référence surtout en termes de détection de tissu exempt de défauts. Le modèle statistique MoGG prouve son efficacité non seulement dans la discrimination de texture [4] et la segmentation d'image [4], mais aussi dans le domaine de détection de défauts de texture. Ceci est dû essentiellement à la capacité de ce

modèle de contrôler la forme de n'importe quel type de distributions de données (platikurtique, leptokurtique, unimodale, multimodale).

En conclusion, la modélisation statistique par les mélanges finis de Gaussiennes généralisées des sous-bandes de la transformée en contourlettes redondantes permet une détection de défauts atteignant 99.94%. Cependant, cette exactitude a un prix qui est le temps de calcul. En effet, le traitement d'une image peut prendre jusqu'à 10 minutes en considérant le traitement par chevauchement de blocs de taille 64×64 pixels. Néanmoins, ceci ne constitue pas une contrainte pour l'utilisation de cette méthode de détection dans l'industrie de textile, du fait que l'ensemble des algorithmes peut être embarqué sur une puce électronique dédiée à l'industrie et qui a un temps de calcul beaucoup plus rapide que les simulateurs. Il est possible aussi d'implémenter l'ensemble des algorithmes sur des processus parallèles. De plus, le processus d'inspection de qualité de la trame de tissu est un processus post-production et ne nécessite pas une application temps réel.

Notre méthode proposée dans le présent mémoire a marqué un succès pour les tissus à texture uniforme. Les travaux futurs qui peuvent être des voies de recherche se présentent comme suit :

1. La généralisation de cette méthode pour résoudre la problématique de la détection de défauts pour les textures avec motifs, du fait que la transformée en contourlettes redondantes combinée avec la modélisation par les mélanges finis de Gaussiennes généralisées [6] a marqué un grand succès dans la discrimination de texture (homogène et avec motif).
2. Extension du modèle de mélanges finis de Gaussiennes généralisées pour permettre de traiter les images en couleur. En effet, nous avons traité dans notre travail des images en niveaux de gris. Cette extension nous permettra de traiter les images en couleur et détecter ainsi certains défauts qui ne sont pas visibles dans les images

en niveaux de gris.

3. Réduire le temps de calcul de la divergence de KL. Nous avons expliqué dans la section 6.8 que le calcul de la divergence de KL est coûteux en termes de temps.

Bibliographie

- [1] F. Ade. Application of Principal Component Analysis to The Inspection of Industrial Goods. *Proc. SPIE*, Vol. 397, pp. 216-223, 1983.
- [2] F. Ade, N. Lins et M. Unser. Comparison of Various Filter Sets for Defect Detection in Textiles. *IEEE Int'l Conf. on Pattern Recognition*, pp. 428-431, 1984.
- [3] B. Aiazzi, L. Alparone, et S. Baronti. Estimation Based on Entropy Matching For Generalized Gaussian PDF Modelling. *IEEE Signal Processing Letters*, Vol. 6, No. 6, pp. 138-140, 1999.
- [4] M. S. Allili. Wavelet-Based Texture Retrieval Using a Mixture of Generalized Gaussian Distributions. *IEEE Int'l Conf. on Pattern Recognition*, pp. 3143-3146, 2010.
- [5] M. S. Allili. Wavelet Modeling Using Finite Mixtures of Generalized Gaussian Distributions : Application to Texture Discrimination and Retrieval. *IEEE Trans. on Image Processing*, Vol. 21, No. 4, pp. 1452-1464, 2012.
- [6] M. S. Allili et N. Baaziz, Contourlet-Based Texture Retrieval Using a Mixture of Generalized Gaussian Distributions. *Int'l Conf. on Computer Analysis of Images and Patterns*, Vol. 2, pp. 446-454, 2011.
- [7] M. S. Allili, N. Bouguila et D. Ziou. Finite General Gaussian Mixture Modeling

- and Application to Image and Video Foreground Segmentation. *Journal of Electronic Imaging* Vol. 17, No. 1, pp. 013005-1 - 013005-13, 2008.
- [8] D. A. Al Nadi, A. M. Mansour. Independent Component Analysis (ICA) for Texture Classification. *IEEE Int'l Multi-Conf. on Systems, Signals and Devices*, pp. 1-5, 2008.
- [9] N. Baaziz. Adaptive Watermarking Schemes based on a Redundant Contourlet Transform. *IEEE Int'l Conf. on Image Processing*, pp. 221-224 , 2005.
- [10] N. Baaziz, O. Abahmane et R. Missaoui. Texture Feature Extraction in the Spatial-Frequency Domain for Content-Based Image Retrieval. *CoRR Information Retrieval and Multimedia*, arXiv :1012.5208, 2010.
- [11] M. Bennamoun et A. Bodnarova. Automatic Visual Inspection and Flaw Detection in Textile Materials : Past, present and future. *IEEE Int'l Conf. on Systems, Man, and Cybernetics*, Vol. 5, pp. 4340-4343, 1998.
- [12] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [13] A. Bodnarova, M. Bennamoun et K. K. Kubik. Defect Detection in Textile Materials Based on Aspects of HVS. *IEEE Int'l. Conf. on Systems, Man, and Cybernetics*, Vol. 5, pp. 4423-4428, 1998.
- [14] A. Boulmerka et M.S. Allili. Thresholding-Based Segmentation Revisited Using Mixtures of Generalized Gaussian Distributions. *IEEE Int'l Conf. on Pattern Recognition*, pp. 2894-2897, 2012.
- [15] C. A. Bouman et B. Liu. Multiple Resolution Segmentation of Textured Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 2, pp. 99-113, 1991.
- [16] G.-E. P. Box et G. C. Tiao. *Bayesian Inference in Statistical Analysis*. Wiley, 1992.

- [17] M. Bradshaw. The Application of Machine Vision to the Automated Inspection of Knitted Fabrics. *Mechatronics*, Vol. 5, No. 2/3, pp. 233-243, 1995.
- [18] P. J. Burt et E. H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Trans. on Communications*, Vol. 31, No. 4, pp. 532-540, 1983.
- [19] C. H. Chen, L. F. Pau et P. S. P. Wang. *The Handbook of Pattern Recognition and Computer Vision*. 2ème édition, World Scientific, 1999.
- [20] F. S. Cohen et Z. Fan. Rotation and Scale Invariant Texture Classification. *IEEE Conf. on Robotics and Automation*, Vol. 3, pp. 1394-1399, 1988.
- [21] F. S. Cohen, Z. Fan, et S. Attali. Automated Inspection of Textile Fabrics Using Textural Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 13, pp. 803-808, 1991.
- [22] A. Conci et C. B. Proença. A Fractal Image Analysis System for Fabric Inspection Based on Box-Counting Method. *Computer Networks and ISDN Systems*, Vol. 30, No. 20, pp. 1887-1895, 1998.
- [23] A. Conci et C. B. Proença. A Computer Vision Approach for Textile Inspection. *Textile Research Journal*, Vol. 70, No. 4, pp. 347-350, 2000.
- [24] R. W. Connors et C. A. Harlow. A Theoretical Comparison of Texture Algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 2, No. 3, pp. 204-222, 1980.
- [25] R. Cross et A. K. Jain. Markov Random Field Texture Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 5, No. 1, pp. 25-39, 1983.

- [26] G. S. Desoli, S. Fioravanti, R. Fioravanti et D. Corso. A System for Automated Visual Inspection of Ceramic Tiles. *IEEE Int'l Conf. on Industrial Electronics, Control, and Instrumentation*, Vol. 3, pp. 1871-1876, 1993.
- [27] M. N. Do et M. Vetterli. Contourlets : A Directional Multiresolution Image Representation. *IEEE Int'l Conf. on Image Processing*, pp. 357-360, 2002.
- [28] M. N. Do et M. Vetterli. Wavelet-Based Texture Retrieval Using Generalized Gaussian Density and Kullback-Leibler Distance. *IEEE Trans. on Image Processing*, Vol. 11, No. 2, pp. 146-158, 2002.
- [29] M. N. Do et M. Vetterli. The Contourlet Transform : an Efficient Directional Multiresolution Image Representation. *IEEE Trans. on Image Processing*, Vol. 14, No. 12, pp. 2091-2106, 2005.
- [30] J. L. Dorrity et G. Vachtsevanos. On-line Defect Detection for Weaving Systems. *IEEE Technical Conf. on Textile, Fiber and Film Industry*, pp. 1-6, 1996.
- [31] G. Fan et X. G. Xia. Wavelet-Based Texture Analysis and Synthesis Using Hidden Markov Models. *IEEE Trans. on Circuits and Systems-Part I : Fundamental Theory and Applications*, Vol. 50, No. 1, pp 106-120, 2003.
- [32] J. E. Fowler. The Redundant Discrete Wavelet Transform and Additive Noise. *IEEE Signal Processing Letters* Vol. 12, No. 9, pp 629-632, 2005.
- [33] J. Goldberger, S. Gordon et H. Greenspan. An Efficient Image Similarity Based on Approximations of KL-Divergence Between Two Gaussian Mixtures. *IEEE Int'l Conf. on Computer Vision*, pp.487-493, 2003.
- [34] R.C. Gonzalez et R.E. Woods. *Digital Image Processing*. 3ème édition, Prentice Hall, 2008.

- [35] R. W. Hamming. Error Detecting and Error Correcting Codes. *29th Bell System Technical Journal*, No. 2, pp. 147-160, 1950.
- [36] J. R. Hershey et P. A. Olsen. Approximating the Kullback-Leibler Divergence Between Gaussian Mixture Models. *IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, pp. 317-320, 2007.
- [37] P. O. Hoyer et A. Hyvärinen. Independent Component Analysis Applied to Feature Extraction from Colour and Stereo Images. *Network : Computation in Neural Systems* Vol. 11, pp. 191-210. 2000.
- [38] J. Huart et J.-G. Postaire. Integration of Computer Vision on to Weavers for Quality Control in the Textile Industry. *Proc. SPIE*, Vol. 2183, pp. 155-163, 1994.
- [39] R. Jadav, K. Thakkar et V. Patel. Fabric Defects and Point Grading System. <http://www.scribd.com/doc/52291149/FABRIC-DEFECTS-AND-POINT-GRADING-SYSTEM>. *Dernière consultation : Février 2012*.
- [40] W. J. Jasper et H. Potapalli. Image Analysis of Mispicks in Woven Fabrics. *Textile Research Journal*, Vol. 65, No. 11, pp. 683-692, 1995.
- [41] A. Kumar. Automated defect detection in textured materials. *Ph.D. dissertation, Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, May 2001*.
- [42] A. Kumar. Computer-Vision-Based Fabric Defect Detection : A Survey. *IEEE Trans. on Industrial Electronics*, Vol. 55 , No. 1, pp. 348-363 , 2008.
- [43] J. S. Lane. Textile Fabric Inspection System. *U.S. Patent 5*, pp. 774-177, 1998.

- [44] S. A. Mallat. Theory For Multiresolution Signal Decomposition : The Wavelet Representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 674-693, 1989.
- [45] B. N. Nickolay et H. Schmalfuâ. Automatic Fabric Inspection, Utopia or Reality? *Melliand Textilberichte*, Vol. 73, pp. 33-37, 1993.
- [46] L. Norton-Wayne, M. Bradshaw et A. J. Jewell. Machine Vision Inspection of Web Textile Fabric. *British Machine Vision Conf.*, pp. 217-226, 1992.
- [47] L. Norton-Wayne, M. Bradshaw et C. Sandby. Machine Vision for The Automated Inspection of Web Materials. *Proc. SPIE* Vol. 1989, pp. 2-13, 1993.
- [48] T. Ojala, M. Pietikäinen et T. Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence* Vol. 24, No. 7, pp 971-987, 2002.
- [49] J. J. Oliver, R. A. Baxter et C. S. Wallace. Unsupervised Learning Using MML. *Int'l. Conf. Mach. Learn.* pp. 364-372, 1996.
- [50] S. Piotin. Vers un Outil C.A.O. Pour la Maille. *THESE de Doctorat de l'Université de Reims Champagne-Ardenne (URCA)*, Ecole Doctorale Sciences, Technologies et Santé, France 2009.
- [51] D. D. -Y. Po et M. N. Do. Directional Multiscale Modeling of Images using the Contourlet Transform. *IEEE Trans. on Image Processing*, Vol. 15, No. 6, pp. 1610-1620, 2006.
- [52] C. P. Robert et G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.
- [53] A. Rosenfeld et A. Kak. *Digital Picture Processing*. 2ème édition, Computer Science and Applied Mathematics, 1982.

- [54] P. S. Sanjekar et P. S. Dhabe. Fingerprint Verification using Haar Wavelet. *IEEE Int'l Conf. on Computer Engineering and Technology*, Vol. 3, pp 361-365, 2010.
- [55] H. Sari-Sarraf et J. S. Goddard. Vision Systems for On-Loom Fabric Inspection. *IEEE Trans. on Industry Applications*, Vol. 35, No. 6, pp. 1252-1259, 1999.
- [56] A. Serdaroglu, A. Ertuzun et A. Ercil. Defect Detection in Textile Fabric Images using Wavelet Transforms and Independent Component Analysis. *Pattern Recognition and Image Analysis*, Vol. 16, No. 1, pp. 61-64, 2006.
- [57] O. G. Sezer, A. Ertuzun et A. Ercil. Independent Component Analysis for Texture Defect Detection. *Pattern Recognition and Image Analysis*, Vol. 14, No. 2, pp. 303-307, 2004.
- [58] O. G. Sezer, A. Ertuzun, et A. Ercil. Using Perceptual Relation of Regularity and Anisotropy in the Texture with Independent Component Model for Defect Detection. *Pattern Recognition*, Vol. 40, No. 1, pp. 121-133, 2007.
- [59] L. H. Siew, R. M. Hodgson et E. J. Wood. Texture Measures for Carpet Wear Assessment. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 1, pp. 92-105, 1988.
- [60] K. Srinivasan, P. H. Dastor, P. Radhakrishnaihan et S. Jayaraman. FDAS : A Knowledge-Based Frame Detection Work for Analysis of Defects in Woven Textile Structures. *Journal of the Textile Institute*, Vol. 83, No. 3, pp. 431-447, 1992.
- [61] M. Unser et F. Ade. Feature Extraction and Decision Procedure for Automated Inspection of Textured Materials. *Pattern Recognition Letters*, Vol. 2, No. 3, pp. 181-191, 1984.

- [62] S. Valette et R. Prost. Wavelet-Based Multiresolution Analysis of Irregular Surface Meshes. *IEEE Trans. Visualisation and Computer Graphics*, Vol. 10, No. 2, pp. 113-122, 2004.
- [63] M. Vetterli et J. Kovacevic. *Wavelets and Subband Coding*. Prentice Hall, 1995.
- [64] C. S. Wallace. Statistical and Inductive Inference by Minimum Message Length. *Information Science and Statistics*. Springer Verlag, 2005.
- [65] C. S. Wallace et P. R. Freeman. Estimation and Inference by Compact Coding. *J. Royal Stat. Society (Ser. B)*, Vol. 49, No. 3, pp. 240-265, 1987.
- [66] Y. F. Zhang et R. R. Bresee. Fabric Defect Detection and Classification using Image Analysis. *Textile Research Journal*, Vol. 65, No. 1, pp. 1-9, 1995.
- [67] R. Zhang, X. P. Zhang et L. Guan. Wavelet-Based Texture Retrieval using Independent Component Analysis. *IEEE Int'l Conf. on Image Processing*, Vol 6, pp VI-341 - VI-344, 2007.
- [68] Bureau National de Statistiques de la Chine. <http://www.stats.gov.cn/>. *Dernière consultation : Juillet 2012*.
- [69] CARTE ROUTIÈRE TECHNOLOGIQUE DE L'INDUSTRIE CANADIENNE DES TEXTILES. <http://www.ic.gc.ca/eic/site/textiles-textiles.nsf/fra/accueil>. *Dernière consultation : Juillet 2012*.
- [70] Vision Texture, <http://vismod.media.mit.edu/vismod/>. *Dernière consultation : Février 2013*.
- [71] Workgroup on Texture Analysis of DFG. Tilda Textile Texture Database. <http://lmb.informatik.uni-freiburg.de/resources/datasets/tilda.en.html>. *Dernière consultation : Février 2013*.