



HAL
open science

Décompositions spatio-temporelles pour l'étude des textures dynamiques : contribution à l'indexation vidéo

Sloven Dubois

► **To cite this version:**

Sloven Dubois. Décompositions spatio-temporelles pour l'étude des textures dynamiques : contribution à l'indexation vidéo. Autre [cs.OH]. Université de La Rochelle, 2010. Français. NNT : 2010LAROS310 . tel-00605667

HAL Id: tel-00605667

<https://theses.hal.science/tel-00605667>

Submitted on 3 Jul 2011

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

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



UNIVERSITÉ DE LA ROCHELLE

*ÉCOLE DOCTORALE SCIENCES ET
INGÉNIERIE POUR L'INFORMATION*

Laboratoire Informatique, Image et Interaction

Laboratoire Mathématiques, Image et Applications



THÈSE présentée par :

Sloven DUBOIS

19 novembre 2010

pour obtenir le grade de : **Docteur de l'université de La Rochelle**

Mention : **Automatique, Image et Signal**

**Décompositions spatio-temporelles
pour l'étude des textures dynamiques**
Contribution à l'indexation vidéo

JURY :

Laure BLANC-FÉRAUD

Directrice de recherche au CNRS, Université de Sophia
Antipolis, Rapporteur

Christian GERMAIN

Professeur, ENITA de Bordeaux, Rapporteur

Denis PELLERIN

Professeur, Université de Grenoble, Examineur

Michel BERTHIER

Professeur, Université de La Rochelle, Examineur

Michel MÉNARD

Professeur, Université de La Rochelle, Directeur de thèse

Renaud PÉTERI

Maître de conférences, Université de La Rochelle, Co-
directeur de thèse

Note au lecteur

Le domaine d'application de ce manuscrit étant l'analyse de vidéos, de nombreux résultats se présentent sous la forme de séquences d'images. Pour que le lecteur puisse les visionner, celles-ci sont présentes sur mon site internet : <http://sloven.dubois.free.fr>. Ces figures annotées par le symbole ♠. Le nom de la vidéo correspond à la référence de la figure.

Remerciements

Je souhaite tout d'abord exprimer ma profonde gratitude à Michel Ménard, Professeur au sein de l'équipe Imédoc du laboratoire L3i de l'Université de La Rochelle, pour m'avoir témoigné autant de confiance au début de ces travaux. Je le remercie donc pour m'avoir permis de réaliser ce travail de thèse et pour en avoir assuré la direction avec enthousiasme, disponibilité et rigueur scientifique. Je tiens à souligner également l'aide précieuse et les conseils avisés qu'il m'a prodigué au travers de nos nombreuses discussions pendant ces dernières années et ce, malgré son emploi du temps bien rempli.

Je tiens à remercier Renaud Péteri, Maître de Conférences au sein de l'équipe Mathématiques et Image du laboratoire MIA de l'Université de La Rochelle, qui a co-dirigé cette thèse avec patience tout au long de ces trois années. Il a su m'apporter le soutien tant méthodologique que scientifique nécessaire à l'accomplissement de ce travail. Je le remercie également pour les nombreuses connaissances qu'il a su me transmettre sur les domaines de l'analyse multi-résolution et des textures dynamiques.

Je tiens à exprimer ma sincère reconnaissance à Laure Blanc-Féraud, Directrice de recherche au pôle SIS du laboratoire i3S de l'Université Sophia Antipolis et à Christian Germain, Professeur au sein de l'équipe Signal et Image du laboratoire IMS de l'Université de Bordeaux, pour avoir accepté la lourde tâche de rapporteur. Qu'ils soient grandement remerciés pour leur lecture approfondie du manuscrit, leurs conseils qui ont contribué à en améliorer le contenu ainsi qu'à leur participation au jury.

Mes remerciements sont également adressés à Denis Pellerin, Professeur au sein de l'équipe Architecture, Géométrie, Perception, Images, Gestes du laboratoire Gipsa-lab de l'Université de Grenoble et à Michel Berthier, Professeur au sein de l'équipe Mathématiques et Images du laboratoire MIA de l'Université de La Rochelle, pour avoir accepté de faire partie de ce jury en tant qu'examineurs.

Je remercie Michel Berthier, Directeur du laboratoire MIA, et Remy Mullot, Directeur du laboratoire L3i, pour leurs accueils au sein de leurs laboratoires ainsi que pour leurs conseils. Ces deux laboratoires sont des endroits vivants et agréables, dans lesquels j'ai passé trois intenses années, chargées de travail et de stress mais aussi de joies et d'instantanés mémorables. Mes remerciements vont également au personnel administratif que j'ai pu côtoyer tout au long de ces années pour leur disponibilité, leur gentillesse et leur accueil.

Mes remerciements vont aussi à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail. Je souhaite exprimer ma gratitude à mes différents collègues de bureau avec qui j'ai passé ces trois années. Plus particulièrement à Agathe Chollet, pour nos nombreuses discussions qui ont su animer les différentes heures passées ensemble, pour ses remarques toujours très directes et ô combien efficaces, à José Mennesson pour son soutien et sa gentillesse sans limite, à Charles Guyon pour son ingéniosité et ses discussions extravagantes et à Romain Quéré pour

sa jovialité et son engouement. Merci à vous, pour tous ces moments passés et à venir, à errer dans les cafés ou ailleurs. Enfin, je souhaite bon courage à tous les doctorants actuels et leur souhaite comme moi de sortir enfin de ce tunnel.

J'aimerais également remercier tous mes amis et mes proches pour m'avoir supporté durant toute cette période nécessaire à la réalisation d'une thèse.

Enfin, je remercie ma compagne, Sophie, de partager sa vie, son temps, ses pensées, son amour et ses conseils avec moi, mais aussi de m'avoir encouragé durant tous les moments difficiles et pour son soutien absolument sans faille dans ce projet professionnel. Je lui dédie cette thèse, ainsi qu'à notre fille qui va bientôt remplir notre vie de joie et de tendresse.

Table des matières

Remerciements	5
Introduction générale	11
1 Textures Dynamiques : un état de l'art	19
1.1 Qu'est ce qu'une texture dynamique ?	22
1.1.1 Un domaine de recherche récent et dynamique	22
1.1.2 Discussion sur les textures dynamiques	24
1.1.3 Bases de données existantes	29
1.2 Caractérisation de textures dynamiques	33
1.2.1 Approches fondées sur le mouvement	34
1.2.2 Approches s'appuyant sur un modèle mathématique	40
1.2.3 Techniques utilisant la géométrie spatio-temporelle	44
1.2.4 Approches fondées sur des transformées spatio-temporelles	48
2 Décompositions multi-échelles de séquences d'images	51
2.1 Décompositions multi-échelles usuelles	54
2.1.1 Analyse multi-résolutions	56
2.1.1.1 Bases d'ondelettes et analyse multi-résolutions	56
2.1.1.2 Bancs de filtres	58
2.1.2 Analyse de séquences d'images par transformées en ondelettes	59
2.1.2.1 Décomposition en ondelettes spatiales	60
2.1.2.2 Décomposition en ondelettes temporelles	61
2.1.2.3 Décomposition en ondelettes 2D+T	61
2.1.2.4 Décomposition en ondelettes par la méthode de J. R. Smith <i>et al.</i>	62
2.1.3 Ondelettes usuelles versus ondelettes géométriques	64
2.2 Transformée en curvelets	65
2.2.1 Curvelets 2D continues	65
2.2.2 Curvelets 2D discrètes	67
2.2.2.1 Description	67
2.2.2.2 Implémentations	69
2.2.2.3 Expérimentations	71
2.2.3 Transformée en curvelets tri-dimensionnelle	74
2.2.3.1 Description	75
2.2.3.2 Passage au 2D+T	77
2.2.3.3 Application aux séquences d'images	78
3 Étude des textures dynamiques	87
3.1 Un modèle de textures dynamiques	90
3.1.1 Définition du modèle	90
3.1.2 Évaluation du modèle de textures dynamiques proposé	93

3.1.3	Perspectives du modèle	94
3.2	Étude de l'onde porteuse	95
3.2.1	Caractéristiques de l'onde porteuse	95
3.2.2	Segmentation spatio-temporelle utilisant directement la transformée en curvelets 2D+T	98
3.2.2.1	Schéma général	98
3.2.2.2	Resultats et discussion	100
3.2.2.3	Perspectives	101
3.2.3	Segmentation spatio-temporelle à l'aide d'une structure en octree et de la transformée en curvelets 2D+T	101
3.2.3.1	Schema général	101
3.2.3.2	Discussion sur les critères d'homogénéité	104
3.2.3.3	Résultats	107
3.2.3.4	Perspectives concernant la structure en octree	112
4	Décomposition de textures dynamiques	115
4.1	Problème de décomposition d'images	118
4.1.1	Discussion	118
4.1.2	L'analyse en composantes morphologiques (MCA)	119
4.1.2.1	Cadre théorique	120
4.1.2.2	Implémentation	121
4.1.2.3	Stratégies de seuillage	123
4.2	Application aux textures dynamiques	126
4.2.1	Choix du dictionnaire	126
4.2.2	Nouvelles stratégies de seuillage	128
4.2.2.1	Théorie	128
4.2.2.2	Évaluation	130
4.2.3	Expérimentations	136
4.2.3.1	Paramètres utilisés	136
4.2.3.2	Résultats	137
5	Applications	143
5.1	Indexation de textures dynamiques	145
5.1.1	Construction des vecteurs caractéristiques	146
5.1.2	Méthode de validation	148
5.1.3	Bases de données utilisées	149
5.1.4	Résultats et discussion	149
5.2	Recherche par le contenu	153
5.3	Estimation du mouvement global	155

Table des matières	9
<hr/>	
Conclusion générale et perspectives	159
Annexes	161
A Littérature des textures dynamiques	163
B Évaluation quantitative du modèle de textures dynamiques	165
C Bases de données expérimentales	169
D Compléments résultats indexation	173
Publications	177
Liste des algorithmes	179
Liste des tableaux	180
Table des figures	182
Bibliographie	189

Introduction générale

Motivations

Les avancées technologiques récentes ont permis aux professionnels et aux particuliers d'acquérir et de stocker quasiment sans limites des documents numériques (textes mais aussi images et vidéos). Ainsi, nous assistons à la croissance des données multimédia qui, si elles ne sont pas bien archivées, deviennent difficilement utilisables. Il devient alors indispensable d'organiser ces masses de données, car toute information non archivée peut être considérée comme perdue.

L'indexation multimédia est, depuis déjà quelques dizaines d'années, un thème de recherche très actif dans la communauté du traitement d'images. Au delà de l'indexation de documents textuels, il s'agit aujourd'hui d'être capable d'indexer par le contenu des documents audio, vidéo ou plus généralement multimédia. L'indexation correspond à l'organisation des données, en fonction d'un ordre défini par un ou des attributs. Il s'agit d'associer aux données multimédia non plus des annotations textuelles, mais des signatures spécifiques décrivant le contenu de l'image ou de la vidéo, souvent liées aux caractéristiques visuelles comme la couleur, la texture, la forme.

Une des caractéristiques importantes pour décrire une image est la texture. La caractérisation de la texture statique (2D) a fait l'objet de très nombreux travaux de recherche, et fait partie des descripteurs de la norme MPEG-7 [Wu 01]

L'extension de ces caractéristiques visuelles à la dimension temporelle pose certaines difficultés. En effet, la notion de texture dans des séquences d'images soulève de nombreuses questions :

- Que sont les textures dans le cadre des vidéos ?
- Est ce que les textures 2D+T sont de nature différente, ou s'agit t'il d'une simple extension à la 3D de structures 2D ?
- Quelles sont les phénomènes induisant une texture 2D+T ? Cette dernière possède-t'elle des propriétés spécifiques et différentes de la texture 2D ?

On peut se demander d'ailleurs si dans le domaine de la vidéo, la notion de texture 2D+T reste un descripteur pertinent.

Il est très probable que oui, car l'information de mouvement représente un réel apport pour la reconnaissance ; dans le monde animal, la grenouille sait par exemple distinguer de part le mouvement un objet d'une proie potentielle, comme une mouche d'une feuille qui tombe. De plus, le mouvement permet d'extraire de nouveaux descripteurs pour l'indexation, comme des mesures liées à l'accélération, à la turbulence ou à la vorticité pour un fluide. Il est donc raisonnable de penser que l'extension de la texture au domaine temporel aura son apport pour l'indexation vidéo, sous réserve de pouvoir la caractériser efficacement.

Dans la littérature, l'extension temporelle de la notion de texture se rencontre sous le nom de texture dynamique (*dynamic texture* en anglais), plus rarement sous le nom de texture temporelle (*temporal texture*). Un drapeau dans le vent, un champ d'herbe ondulant, les vagues de la mer, la surface d'un lac, le mouvement de la forêt, la fumée, le feu, une colonie de fourmis, les ailes d'un moulin qui tournent, les fontaines, les cascades, ... sont autant d'exemples de textures dyna-

miques présentées dans la littérature. La figure 1 présente trois exemples de textures dynamiques. Les séquences d'images sont visualisées ici comme des cubes de données tri-dimensionnelles permettant ainsi d'observer l'axe temporel. Pour chacune d'elles, des répétitivités spatio-temporelles sont clairement visibles.

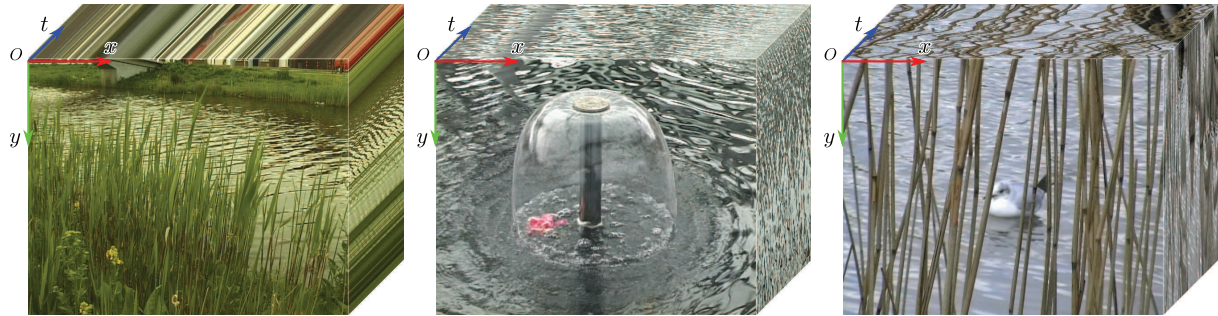


FIG. 1 – ♠ *Exemples de textures dynamiques. Les séquences d'images sont visualisées ici comme des cubes de données tri-dimensionnelles.*

Les textures dynamiques se rencontrent dans de nombreuses scènes naturelles ou artificielles, et donc, tout système d'indexation de vidéo devra pouvoir analyser et caractériser ces textures dynamiques.

L'augmentation de la puissance des ordinateurs ainsi que l'importance du flux de données vidéo a permis l'éclosion récente de nombreux travaux sur les textures dynamiques. Comme nous le verrons dans ce manuscrit, l'augmentation très importante du nombre de publications liées à ce sujet traduit l'intérêt de la communauté scientifique pour ce domaine. Les thèmes de recherche et les applications potentielles des textures dynamiques sont en effet nombreux :

- l'indexation vidéo : les scènes présentant des textures dynamiques sont très courantes et leur reconnaissance faciliterait l'étape d'indexation. L'extraction de caractéristiques pertinentes sur la dynamique des textures ouvre la voie à des requêtes plus sémantiques ("flux turbulent", "mer agitée", ...)
- la vidéo surveillance : un exemple d'application mettant en jeu des textures dynamiques concerne la surveillance des départs de feux dans des forêts : une analyse avancée des textures dynamiques de type feu ou fumée est nécessaire pour pouvoir signaler un début d'incendie sans fausses alertes.
- la segmentation spatio-temporelle : il s'agit d'une étape importante dans l'analyse vidéo, car elle peut, par la suite, être utilisée pour effectuer un résumé vidéo, ou pour détecter une perturbation au sein d'une texture dynamique. Dans le même ordre d'idée, elle peut être appliquée à la soustraction d'un fond dynamique (*background subtraction* en anglais), par exemple pour détecter un individu passant devant une forêt.
- le suivi : dans certaines applications, l'objet à suivre peut se présenter sous la forme d'une texture dynamique (par exemple le suivi d'un vortex dans un fluide ou le suivi de la flamme olympique).
- synthèse vidéo : il s'agit d'un domaine applicatif important des textures dynamiques, notamment par les jeux vidéos ou les films d'animation : rendu réaliste d'un feu, du pelage d'un animal, du mouvement des vagues à la surface de la mer, ...

Toutes ces applications nécessitent de pouvoir caractériser efficacement les textures dynamiques. La figure 2 illustre quelques résultats de la littérature.

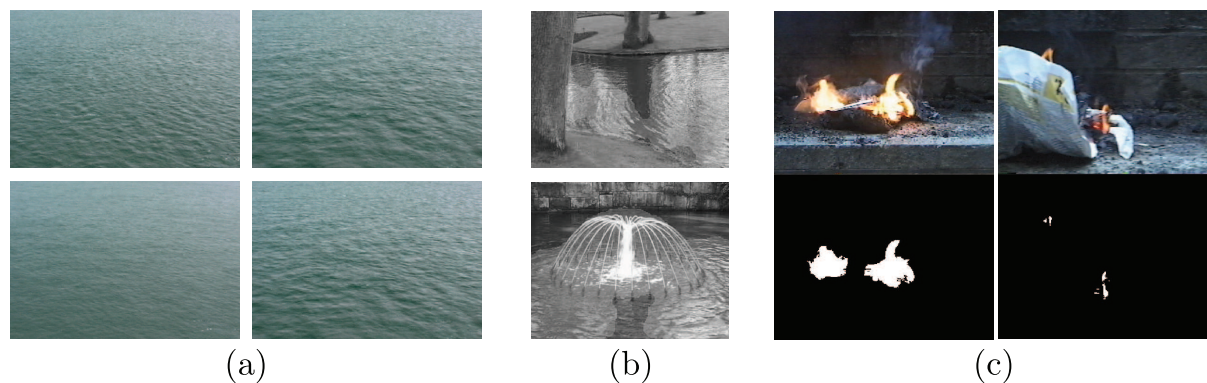


FIG. 2 – Quelques exemples applicatifs des textures dynamiques : (a) synthèse de la surface d'un lac selon plusieurs paramètres (d'après [Doretto 02]), (b) résultats de segmentation spatio-temporelle (d'après [Li 09b]) et (c) détection de changement dans une application de vidéo surveillance (d'après [Phillips 02]).

Contributions

Dans cette thèse, nous nous intéresserons à l'étude et la caractérisation des textures dynamiques dans des séquences d'images avec comme objectif l'indexation dans de grandes bases de vidéos. Une caractérisation pertinente conduira aussi à d'autres applications, comme la segmentation spatio-temporelle ou la suppression de fonds dynamiques.

Il y a eu très peu de thèses sur le sujet, et à notre connaissance, celle-ci est la première en français sur les textures dynamiques. Il est donc important de poser une définition sur ce que sont les textures dynamiques, et de faire un état de l'art du domaine. Malgré la notion intuitive que l'on a des textures dynamiques, l'élaboration d'une définition la plus précise et la plus complète possible pose certaines difficultés. En effet, dans la littérature, nous trouvons souvent des définitions succinctes et très générales qui ne permettent pas une description complète de l'ensemble des textures dynamiques.

La première étape est donc de délimiter le cadre des textures dynamiques. Pour cela, il est nécessaire de comprendre ce qui les génère afin de construire une taxonomie la plus complète et la plus représentative possible. Nous profiterons de cette expertise pour proposer une définition permettant de formaliser ce domaine complexe.

A partir de cette taxonomie, différentes classes de textures dynamiques sont identifiées. Nous nous intéresserons à celles qui représentent une majorité des phénomènes. Afin de mieux comprendre cette classe et de guider notre étude, un modèle formel est proposé. Celui-ci considère les textures dynamiques comme la superposition de plusieurs composantes possédant chacune des propriétés spécifiques : une composante "onde porteuse" et une composante "phénomènes locaux". Ce modèle sera évalué dans un premier temps dans la synthèse de séquences d'images.

L'étape suivante, qui constitue une des préoccupations principales de notre étude, est la construction d'outils d'analyse spatio-temporelle adaptés à la texture dynamique. Comme pour la texture statique dans le domaine de l'image, il est important de prendre en compte les propriétés multi-échelles des textures dynamiques pour pouvoir les caractériser efficacement. Ainsi nos approches seront fondées sur des transformées multi-échelles.

A notre connaissance, une seule transformée de ce type a été proposée dans la littérature pour l'étude des textures dynamiques, celle de Smith *et al.* [Smith 02]. Elle est à l'heure actuelle la seule référence en terme de classification de textures dynamiques par analyse en ondelettes. Cette transformée s'appuie sur une extension presque directe de l'analyse multi-résolutions 1D. Pour compléter cette étude, nous proposerons trois autres extensions de cette approche qui diffèrent par la manière dont sont considérées les variables spatiales et temporelles dans l'analyse multi-résolutions.

Dans le domaine de l'image, il est connu que la transformée en ondelettes échoue dans la représentation/détection d'objets anisotropiques, comme les structures courbes, car elle n'est pas adaptée à la géométrie de l'image. Dans l'étude des textures dynamiques, et donc dans le domaine de la vidéo, nous sommes confrontés à cette même problématique. Pour répondre à cette difficulté, nous implémenterons une transformée en curvelets 2D+T.

La représentation par atomes de curvelets est creuse pour la représentation de structures de co-dimension 1 [Candès 04, Candès 05a], ce qui, dans le cadre des séquences d'images, correspond à des surfaces spatio-temporelles.

La transformée en curvelets 2D+T paraît donc être la transformée multi-échelles la plus pertinente pour représenter la composante "onde porteuse" de notre modèle de textures dynamiques. Pour mettre en avant cette propriété, un algorithme de segmentation spatio-temporelle à l'aide de la transformée en curvelets 2D+T et d'une structure en octree sera étudié.

Les textures dynamiques sont complexes, il est crucial de pouvoir les simplifier pour mieux comprendre les phénomènes sous-jacents. Le modèle de textures dynamiques que nous proposons permet de les séparer en deux composantes. Cela soulève cependant un problème majeur : comment séparer les séquences de textures dynamiques en deux composantes avec chacune leurs propres caractéristiques ?

Deux approches de décomposition sont envisageables pour répondre à ce problème :

- les modèles de décomposition de Meyer utilisant la variation totale [Meyer 01].
- l'analyse en composantes morphologiques s'appuyant sur la projection du signal sur un ensemble de bases d'un dictionnaire [Starck 04].

L'approche de l'analyse en composantes morphologiques a finalement été retenue pour plusieurs raisons :

- le principe même de l'analyse en composantes morphologiques, utilisant un dictionnaire comprenant différentes bases, est en meilleure adéquation avec le modèle que nous proposons. En effet, comme nous venons de l'évoquer, la transformée en curvelets 2D+T est adaptée à la représentation de la composante "onde porteuse" de notre modèle. Ce dernier étant susceptible d'évoluer (ajout d'autres composantes), il est envisageable d'intégrer dans l'analyse en composantes morphologiques de nouvelles bases adaptées.
- l'extension temporelle des modèles de décomposition de Meyer fait parallèlement l'objet

d'une thèse dans les laboratoires MIA et L3i de l'Université de La Rochelle.

Les apports de l'analyse en composantes morphologiques guidée par notre modèle de textures dynamiques seront illustrés par plusieurs applications : décomposition de textures dynamiques en une composante "onde porteuse" et une composante "phénomènes locaux" et estimation du mouvement global d'une texture dynamique.

L'ensemble des outils d'analyse spatio-temporelle mis en place sera utilisé pour l'indexation et pour la recherche par le contenu de textures dynamiques. Ces applications permettent de parfaire l'étude de Smith *et al.* [Smith 02], et ainsi contribuer à la caractérisation de textures dynamiques à l'aide de transformées multi-résolutions spatio-temporelles.

La figure 3 résume l'ensemble des contributions de ce manuscrit.

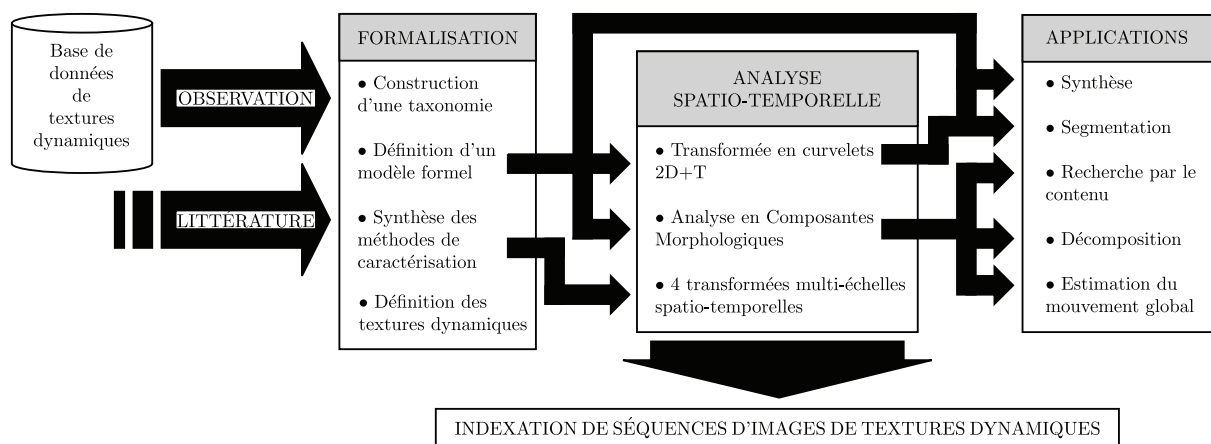


FIG. 3 – Ensemble des contributions de ce manuscrit.

Structure du manuscrit

Ce document se divise en cinq chapitres.

Dans le **chapitre 1**, après avoir discuté en détails de l'intérêt de caractériser une texture dynamique, nous évoquerons les définitions rencontrées dans la littérature. Nous proposerons ensuite une taxonomie permettant une classification la plus complète possible de l'ensemble des textures dynamiques. De ces réflexions, une définition de ce phénomène complexe sera donnée. Nous évoquerons aussi les bases de textures dynamiques rencontrées dans la littérature, et nous essayerons d'évaluer leur pertinence pour notre étude.

La deuxième partie de ce chapitre sera consacrée à une synthèse des méthodes permettant de caractériser les textures indépendamment des domaines applicatifs. Nous établirons et décrirons quatre types d'approches reposant sur le mouvement, les propriétés géométriques de l'espace spatio-temporel, des transformées spatio-temporelles et enfin des modèles mathématiques. Pour chacun, nous mettrons l'accent sur l'information prise en compte, ainsi que sur les descripteurs extraits. Enfin, nous justifierons et positionnerons notre approche fondée sur l'utilisation de décompositions spatio-temporelles.

Le **chapitre 2**, après avoir présenté une courte introduction de l'analyse multi-résolutions dans le cadre mono-dimensionnel, étudie quatre transformées en ondelettes, et présentera leurs avantages et inconvénients dans le cadre de notre étude. Notre objectif est de relever les points critiques afin de souligner la nécessité de rechercher des transformées ayant un caractère plus géométriques. Nous présenterons alors les différentes familles de transformées multi-échelles géométriques de la littérature.

Nous nous focaliserons ensuite sur une transformée multi-échelles géométrique non-adaptative : la transformée en curvelets. Après avoir évoqué sa théorie et son implémentation dans le cas 2D puis 3D, nous développerons plus longuement son adaptation à l'espace 2D+T. Nous mettrons plus particulièrement l'accent sur le développement d'outils d'aide à l'interprétation et à la visualisation de l'espace des coefficients de la transformée en curvelets 2D+T.

Fort de ces outils d'analyse 2D+T, les trois chapitres suivant se concentrent sur l'étude des textures dynamiques.

Dans le **chapitre 3**, nous proposons un modèle formel d'une classe de textures dynamiques. Celui-ci est inspiré de la synthèse d'images et de l'observation de la base de données DynTex [Péteri 10b]; les textures dynamiques sont ainsi perçues comme la superposition de plusieurs phénomènes physiques. Nous montrerons que les synthèses réalisées à l'aide de ce modèle témoignent de sa pertinence.

Une fois le modèle défini, nous présenterons l'étude d'une de ses composantes. Nous montrerons que l'utilisation de la transformée en curvelets 2D+T apporte une solution pour la caractérisation de celle-ci. Pour cela, deux méthodes de segmentation, au sens textures dynamiques, seront présentées et appliquées à des séquences d'images synthétiques et naturelles.

Le chapitre 3 propose une formalisation d'une classe particulière de textures dynamiques, mais l'estimation de ses différentes composantes s'avère difficile. Toutefois, les approches de décomposition d'images de la littérature semblent tout à fait pertinentes pour notre problème. Ainsi, le **chapitre 4**, après une introduction sur les méthodes de décomposition, se focalise sur l'analyse en composantes morphologiques. Nous présenterons le cadre théorique de cette approche, ainsi que son implémentation et les stratégies de seuillage existantes.

Nous nous focaliserons ensuite sur son application aux textures dynamiques. Après étude des bases de la littérature les plus représentatives des différentes composantes de notre modèle, nous proposerons un dictionnaire adapté à son étude. Puis nous présenterons et interpréterons des résultats de décomposition sur des séquences d'images de la base DynTex. Nous mettrons ensuite l'accent sur les contraintes de temps de calcul de l'algorithme de l'analyse en composantes morphologiques dans le cas des séquences d'images. Afin de lever ce verrou, nous proposerons de nouvelles stratégies de seuillage. Celles-ci seront validées et comparées à celles de la littérature au travers de plusieurs critères fondés sur les normes ℓ_0 et ℓ_2 .

Nous proposons de terminer cette étude par un **chapitre 5** présentant plusieurs applications, mettant en évidence l'intérêt de nos contributions.

Les différents outils d'analyse ayant permis une meilleure analyse des textures dynamiques, nous montrerons leurs apports dans le cadre de l'indexation de vidéos. Pour cela, nous étudierons plusieurs descripteurs utilisés couramment lors de l'utilisation de transformées multi-échelles.

Ceux-ci seront ensuite analysés sur trois bases de séquences d'images de complexité croissante. Une étude de recherche par le contenu sera également présentée : nous soumettrons des requêtes sous forme de séquences d'images. Pour finir, notre méthode sera utilisée pour extraire le mouvement global d'une texture dynamique.

Enfin, ce manuscrit se conclut en résumant nos contributions et en présentant quelques perspectives tant méthodologiques que d'un point de vue applicatif.

CHAPITRE 1

Textures Dynamiques : un état de l'art

Textures Dynamiques : un état de l'art

Sommaire

1.1	Qu'est ce qu'une texture dynamique ?	22
1.1.1	Un domaine de recherche récent et dynamique	22
1.1.2	Discussion sur les textures dynamiques	24
1.1.3	Bases de données existantes	29
1.2	Caractérisation de textures dynamiques	33
1.2.1	Approches fondées sur le mouvement	34
1.2.2	Approches s'appuyant sur un modèle mathématique	40
1.2.3	Techniques utilisant la géométrie spatio-temporelle	44
1.2.4	Approches fondées sur des transformées spatio-temporelles	48

Au regard de la littérature abondante et variée sur le sujet des textures 2D, nous constatons que les définitions de celles-ci sont bien établies et référencées. Il existe plusieurs définitions qui dépendent des objectifs que l'on se fixe. A chacune d'elle est ensuite associée une méthodologie : champ de Markov, analyse fréquentielle, analyse spatiale, ...

Concernant les séquences d'images, nous pouvons nous poser les questions suivantes : que sont les textures dans le cadre des vidéos ? Que devient une texture 2D lorsque l'on ajoute une dimension temporelle ? Quelles sont les particularités des textures 2D+T, où s'agit-il d'une simple extension à la 3D de structures 2D ? Qu'appelle t'on texture dynamique ?

Dans la première partie de ce chapitre, après avoir discuté de l'intérêt de caractériser une texture dynamique, nous discuterons des définitions rencontrées dans la littérature, puis nous proposerons une taxonomie permettant une description la plus complète possible de l'ensemble des textures dynamiques. Nous décrirons plusieurs bases de données de textures dynamiques rencontrées dans la littérature. Nous essayerons d'évaluer leur pertinence pour notre étude.

La deuxième partie de ce chapitre sera consacrée à un état de l'art des méthodes de caractérisation des textures dynamiques. Pour chacune des familles rencontrées, nous mettrons l'accent sur l'information prise en compte ainsi que les descripteurs extraits. Enfin, nous justifierons et positionnerons notre approche reposant sur l'utilisation de décompositions spatio-temporelles pour l'étude des textures dynamiques.

1.1 Qu'est ce qu'une texture dynamique ?

Un drapeau dans le vent, un champ d'herbe ondulant, les vagues de la mer, la surface d'un lac, le mouvement de la forêt, la fumée, le feu, ... sont autant d'exemples de textures dynamiques présentés dans la littérature. D'autres exemples sont illustrés sur la figure 1.1.

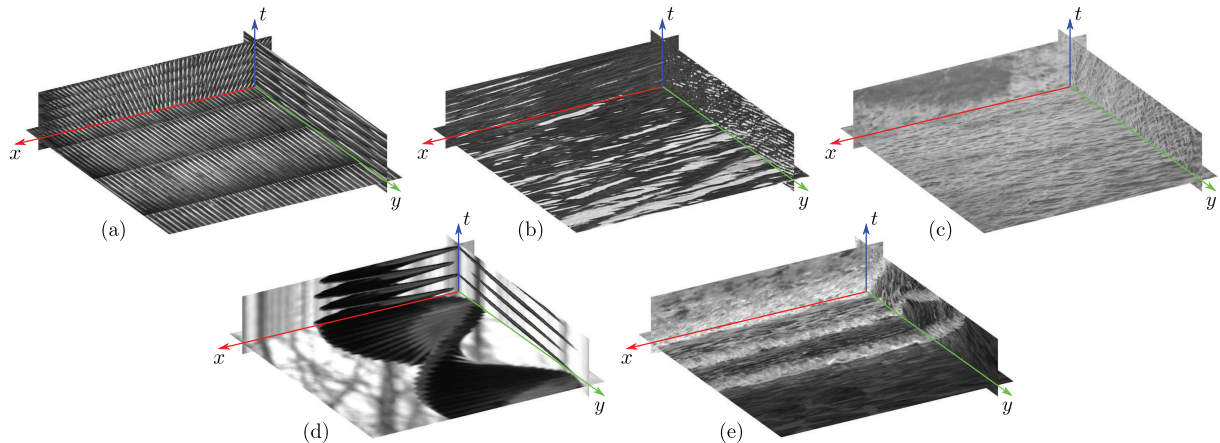


FIG. 1.1 – Exemples de textures dynamiques. On peut observer les séquences d'images d'un escalator (a), de la surface d'un lac (b) et (c), d'une hélice sans fin en rotation due au vent (d) et d'une mer calme (e). Ici chaque séquence d'images est vue comme un cube de données 3D dans lequel on réalise des coupes afin d'observer les différents comportements spatio-temporels.

Avant d'examiner les définitions existantes des textures dynamiques et de proposer un modèle formel, nous évaluons l'intérêt de la caractérisation des textures dynamiques dans le cadre de l'analyse vidéo.

1.1.1 Un domaine de recherche récent et dynamique

Les textures dynamiques représentent un sujet d'étude de plus en plus d'actualité. Comme le graphique de la figure 1.2 le suggère, le nombre des principales publications sur ce thème est en forte augmentation ces dernières années (par principales publications, nous entendons les revues ou les congrès de bonne notoriété, portant clairement sur les textures dynamiques et leur caractérisation, cf. Annexe A).

Les premières publications sur ce domaine apparaissent en 1992 avec les articles de Nelson et Polana [Nelson 92, Polana 92], dans lesquels les auteurs parlent de textures temporelles, et commencent à définir un premier cadre formel d'étude. Par la suite, jusqu'en 2001, quelques publications évoquent le sujet et proposent des méthodes pour les caractériser. C'est surtout à partir de l'année 2002 que le nombre de publications augmente.

La baisse relative sur les dernières années du nombre de publications est due au délai de latence du référencement.

L'émergence de ce contexte de recherche peut s'expliquer, d'une part, par la démocratisation du support vidéo et des performances des ordinateurs pouvant les traiter, et d'autre part, par un vaste champ applicatif :

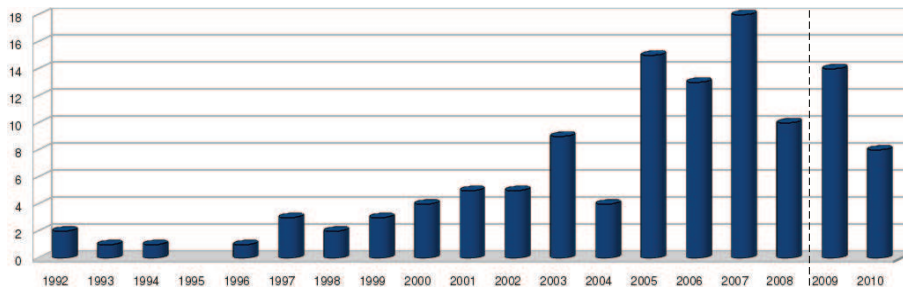


FIG. 1.2 – Principales publications sur le thème des textures dynamiques au cours des dernières années.

- indexation vidéo : dans des bases de séquences d’images, l’objectif est de formuler des requêtes de plus en plus élaborées, associant des caractéristiques de nature sémantique. Par exemple, rechercher des vidéos contenant une mer agitée (la notion “agitée” est une caractéristique temporelle), ou un feu, un lac calme, une forêt très peu agitée ...
- vidéo surveillance : dans certaines séquences, la texture dynamique est une caractéristique importante de la scène. Détecter un accident ou un comportement à risque dans un trafic routier, surveiller et caractériser le mouvement d’une foule, détecter des départs de feu de forêt ou de la fumée sont des exemples où une description robuste de la texture dynamique est nécessaire.
- segmentation spatio-temporelle de séquences d’images : dans le cadre de l’analyse vidéo, la segmentation spatio-temporelle de la séquence, relativement aux textures dynamiques, permettra d’enrichir notre compréhension de la scène et d’optimiser la segmentation en plages caractéristiques. Elle peut conduire à la détection de perturbation au sein d’une texture dynamique (présence d’un véhicule au milieu d’une forêt, d’un canard sur un plan d’eau), à aider à la construction de résumés vidéo (présence à un instant t d’une texture dynamique donnée), à compresser plus efficacement des vidéos en fonction de la nature texturée ou non de la séquence, ...
- soustraction de fonds dynamiques : dans différentes séquences, le fond de la scène peut présenter une texture dynamique. La soustraire permet d’améliorer l’efficacité des algorithmes.
- suivi : suivre des textures dynamiques dans une séquence d’images peut permettre par exemple, d’analyser l’évolution de phénomènes comme le déplacement d’un feu, l’écoulement d’un fluide, ...
- synthèse de vidéos : dans le cadre de la création de film d’animation, de jeux vidéos, d’inpainting vidéo, la synthèse de textures dynamiques est nécessaire pour satisfaire les contraintes de rendu réaliste.

La figure 1.3 illustre la répartition des publications sur les textures dynamiques classées selon les domaines d’application décrits précédemment.

Nous constatons que la synthèse et l’indexation de textures dynamiques sont les problématiques les plus actives. L’indexation est souvent utilisée car elle permet de rapidement tester la pertinence des descripteurs et est facile à mettre en œuvre. La difficulté est d’estimer un vec-

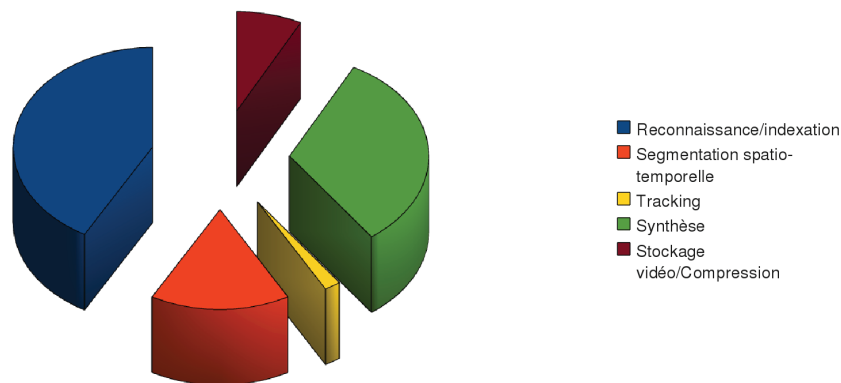


FIG. 1.3 – Répartition des principales publications sur le thème des textures dynamiques selon les principaux domaines applicatifs.

teur caractéristiques pertinent et représentatif des classes de textures dynamiques. La synthèse, quant à elle, doit satisfaire des contraintes de rendu réaliste (et éventuellement d'affichage en temps réel).

Dans le cadre de cette thèse, nous nous intéressons à mieux comprendre et caractériser ce qu'est une texture dynamique. L'objectif de ce chapitre est d'aboutir à sa formalisation. En préalable, nous examinons les définitions proposées dans la littérature.

1.1.2 Discussion sur les textures dynamiques

En parcourant la littérature, nous constatons que les phénomènes complexes que nous nommons textures dynamiques sont identifiées sous différents termes selon les auteurs : *temporal textures* [Nelson 92], *spatio-temporal textures* [Peh 99], *dynamic textures* [Saisan 01], *time-varying textures* [Bar-Joseph 01], *moving textures* [Wang 03].

Ces différentes dénominations sont bien sûr dues à la diversité des définitions sous-jacentes proposées par les auteurs. Dans la littérature, quatre familles de définitions peuvent être identifiées.

La première définition rencontrée apparaît en 1992 avec les articles de Nelson et Polana [Nelson 92, Polana 92]. Dans celui-ci, les auteurs catégorisent les événements d'une séquence d'images en trois grandes classes : les activités, les événements et les textures temporelles.

Les activités se définissent comme des événements périodiques dans le temps et localisés spatialement (une personne qui marche ou saute sur place, un oiseau qui vole, ...). Les événements se décrivent comme une singularité spatio-temporelle ponctuelle ne présentant pas de périodicité temporelle ou spatiale (l'ouverture d'une porte, la chute d'un objet, ...). Dès qu'un événement présente une périodicité temporelle, celui-ci devient une activité. Enfin, les textures temporelles répondent à la définition suivante : elles présentent une régularité statistique, mais possèdent un support indéterminée temporellement et spatialement.

Cette définition fût par la suite reformulée [Chetverikov 06] de la manière suivante :

Définition 1. *Les textures dynamiques sont des séquences d'images de scènes naturelles ayant des propriétés spatiales et temporelles stationnaires.*

Cette définition met l'accent, d'une part, sur la notion de répétitivité de la texture dynamique, à la fois spatiale et temporelle, et d'autre part, sur la notion de scènes naturelles. Ces dernières regroupent l'ensemble des processus issus de la nature et présentant certaines caractéristiques stationnaires.

En 2001, une nouvelle définition de la texture dynamique apparaît [Saisan 01, Soatto 01] :

Définition 2. *Les textures dynamiques correspondent à des scènes naturelles en mouvement dans une séquence d'images, et qui présentent des propriétés stationnaires dans le temps.*

Cette définition est similaire à la précédente, sauf qu'elle considère que l'essence même d'une texture dynamique doit posséder une caractéristique temporelle. Cette définition ne met pas l'accent sur les propriétés spatiales, il s'agit juste de scènes naturelles en mouvement.

La définition suivante, qui apparaît dans la littérature en 2002 [Zhong 02], lie les deux précédentes définitions puisqu'elle n'impose pas que la texture dynamique soit uniquement temporelle ou simultanément spatiale et temporelle :

Définition 3. *Les textures dynamiques correspondent à des scènes de séquences d'images qui présentent certaines structures spatiales ou temporelles périodiques.*

Cette définition est plus générale que les deux précédentes car elle ne se restreint plus aux scènes naturelles. En effet, elle met en avant les structures spatiales ou temporelles que l'on peut rencontrer aussi bien dans les scènes naturelles (processus issus de la nature : forêt, lac, mer, ...), les scènes artificielles (fabriquées par l'être humain : escalator, moulin, ...) et les scènes synthétiques (créées par l'ordinateur : jeux vidéos, ...). Elle prend également en compte les scènes où des textures statiques (purement spatiales) peuvent induire une texture dynamique par le simple fait d'une acquisition effectuée en mouvement (travelling d'une caméra par exemple).

La définition la plus récente que nous avons identifiée dans la littérature s'appuie quant à elle, sur la nature des phénomènes physiques générateurs de la texture dynamique. Celle-ci est rencontrée pour la première fois en 2006 [Woolfe 06] :

Définition 4. *Les textures dynamiques correspondent à des scènes naturelles de séquences d'images en mouvement de nature stochastique.*

Cette définition met l'accent sur les mouvements de la texture. Selon cette approche, une texture dynamique est une scène naturelle non statique, dans laquelle un mouvement caractéristique est présent (un mouvement d'ondulation par exemple) mais dont la localisation spatio-temporelle, dans la séquence d'images, est stochastique.

Cette définition permet d'identifier comme textures dynamiques les déplacements de fluides (vagues sur la mer, ...) et celles présentant des caractéristiques proches (surface d'un lac, les drapeaux, les champs d'herbe ondulant au vent, ...). Toutefois, les textures dynamiques comme

celle induite par le mouvement d'une roue d'un moulin, d'un trafic routier, ... ne correspondent pas à cette définition, puisque le mouvement est ici localisé dans le temps et l'espace. Cependant cette définition a le mérite de classer un certain nombre de textures dynamiques que certains auteurs appellent textures dynamiques fortes [Péteri 10b].

Dans la littérature quatre définitions de la texture dynamique ont donc été identifiées.

En observant le graphique de la figure 1.4 qui illustre la répartition des principales publications en fonction des différentes définitions que nous venons d'évoquer, nous constatons que :

- les 3/4 des publications ne précisent pas de définition. Le domaine de l'image se heurte à des difficultés importantes pour définir de manière formelle ce qu'est une texture spatiale, l'ajout de la dimension temporelle complique encore la tâche. Comme Yves Meyer l'a décrit dans le cadre d'un workshop [Meyer 07], les textures sont "un subtil équilibre entre répétition et innovation". Leur formalisation reste cependant complexe.
- de nombreuses publications sont en accord avec la première définition proposée par Nelson et Polana en 1992. Nous proposons donc dans un premier temps de nous appuyer sur cette première définition pour construire notre propre formalisation.
- les références, en nombre également important, à la définition 2 montrent que cette dernière semble également pertinente.

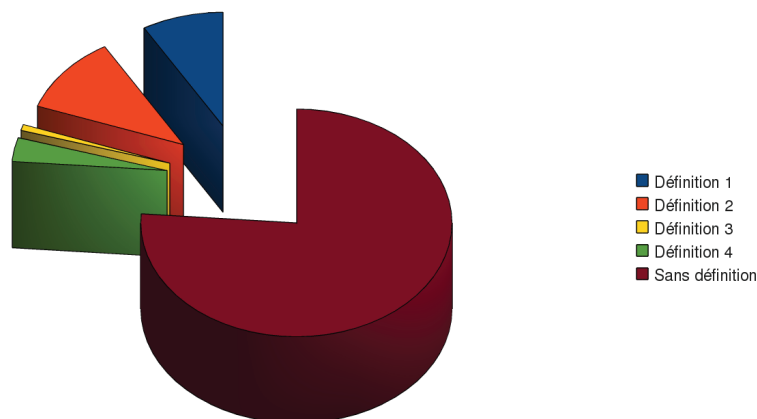


FIG. 1.4 – Répartition des principales publications selon leur définition de la texture dynamique.

La figure 1.5 présente une taxonomie des textures dynamiques. Sur celle-ci, nous observons que :

- une séquence d'images peut contenir des composantes textures statique et/ou dynamique. Par exemple, le pont, les cailloux et le lierre sont des éléments texturés statiques alors que l'eau et l'arbre sont en mouvement ce qui induit une texture dynamique. Dans notre étude, c'est la composante texture dynamique qui nous intéresse.
- la composante texture dynamique contient au moins une texture dynamique. Dans cet exemple, la composante texture vidéo est constituée de deux textures dynamiques distinctes : l'arbre et le ruisseau. Il serait préférable de parler d'arbres en mouvement et

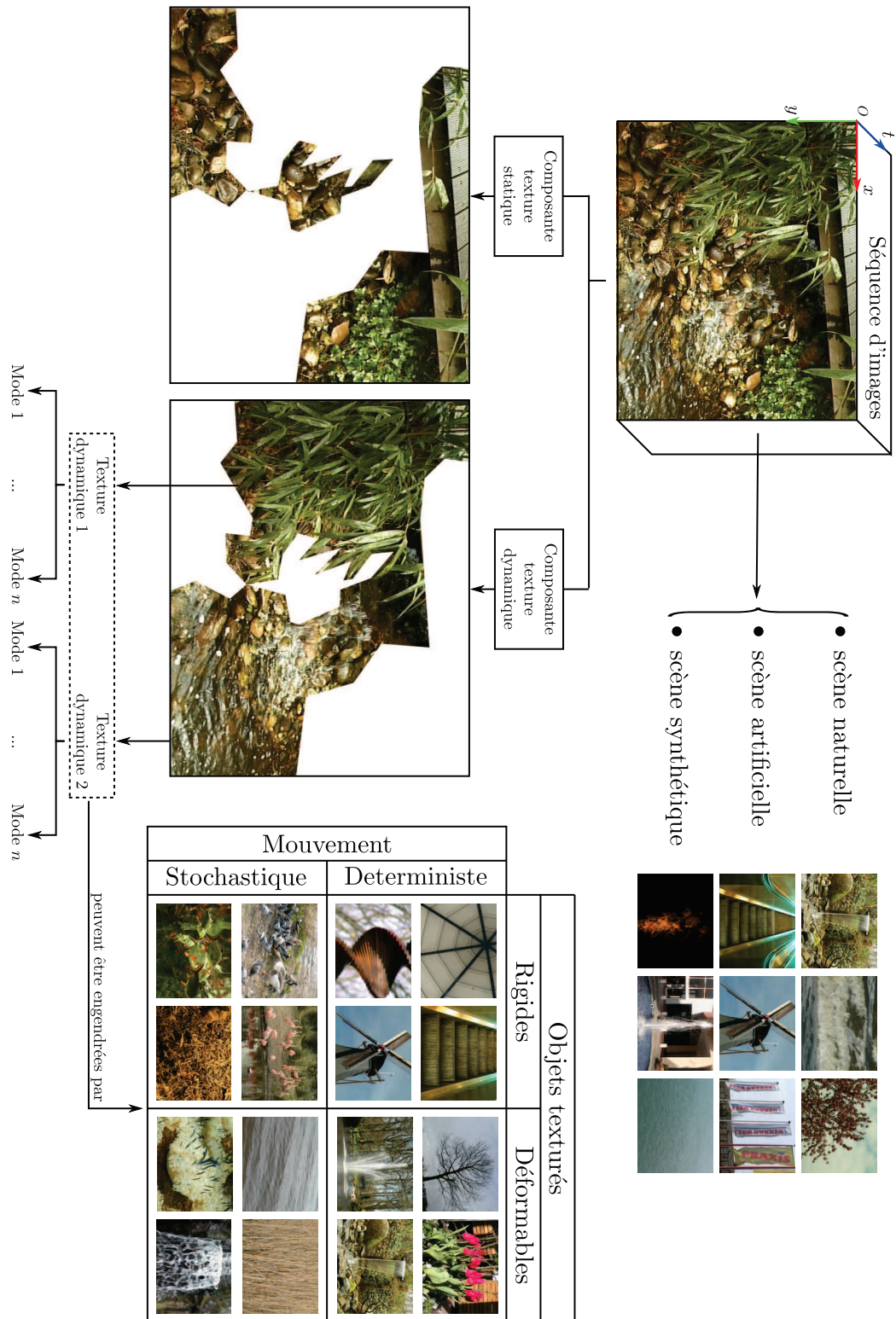


FIG. 1.5 – Taxonomie des textures dynamiques.

d'écoulement d'un ruisseau, mais pour alléger l'écriture, il sera fait référence uniquement aux éléments sources de la texture dynamique. Ces différentes textures dynamiques peuvent bien sûr interférer, ce qui n'est pas le cas ici. Dans le cadre d'une séquence naturelle d'un feu, les oscillations des flammes, les voluptés de fumée en avant plan de l'oscillation d'un champ d'herbe interfèrent pour former une composante texture dynamique complexe.

- quand on observe de nombreuses scènes, nous constatons que les textures dynamiques ne sont pas toutes de même nature : naturelles (issues de la nature), artificielles (créées par l'être humain) et synthétiques (produites par un ordinateur). Une texture dynamique se caractérise par trois facteurs :
 - une structure texturée (rigide ou déformable).
 - un mouvement induit par une force que subit la structure texturée ou par le déplacement de la caméra : ce mouvement peut être d'apparence déterministe ou stochastique. Celui-ci peut être de plusieurs natures : force interne (moteur d'un escalator ou d'une voiture), force externe (le vent dans les ailes d'un moulin) ou induite par la caméra (translation de la caméra).
 - des changements de condition d'acquisition (éclairage, illuminant,...). Ces modifications induisent une modification apparente de la texture et donc l'apparition d'une texture dynamique.

Les ailes d'un moulin présentant un mouvement de rotation, les marches d'un escalator défilant, le trafic routier, ... sont des exemples de textures dynamiques issues d'une structure texturée rigide possédant un mouvement déterministe. Par contre, un banc de poisson, une colonie de fourmis, sont également des structures texturées rigides mais qui ont un mouvement stochastique.

Les textures dynamiques constituées de structures texturées déformables et présentant des mouvements stochastiques sont par exemple une cascade d'eau avec de multiples remous, une anémone ballottée par le courant, ... Par contre, un arbre ou des fleurs oscillants sous un vent constant, une cascade sans remous, ... sont des exemples de textures dynamiques issues de structures déformables et présentant un mouvement déterministe. Toutefois, selon l'échelle d'observation, certains de ces phénomènes peuvent être vus comme des structures rigides animées par un mouvement déterministe.

- l'observation de certaines textures dynamiques permet de constater qu'elles peuvent être la composition de plusieurs phénomènes. Prenons l'exemple de la vidéo de la surface de la mer (vidéo (e) de la figure 1.1), on observe l'écume (mouvement haute fréquence) portée par les vagues (mouvement basse fréquence). Ces composantes appartiennent à la même texture dynamique et peuvent être définis comme étant des modes [Péteri 10b]. Ces modes peuvent se superposer.

Ainsi les séquences d'images de la figure 1.1 possèdent toutes une composante texture vidéo comportant une seule texture dynamique. Les textures dynamiques des vidéos (a) et (e) sont constituées de deux modes alors que les autres sont constitués d'un seul. Par contre, dans le cadre de la séquence d'images de la figure 1.5, nous avons deux textures dynamiques d'un mode chacune dans la composante texture vidéo.

L'ensemble de ces réflexions nous conduit à la définition suivante :

Définition 5. *Une séquence d'images naturelle, artificielle ou synthétique peut contenir une composante texture statique et/ou une composante texture dynamique. Cette dernière est composée d'au moins une texture dynamique.*

Et la texture dynamique répond à la définition suivante :

Définition 6. *Une texture dynamique est une structure texturée qui peut être rigide ou déformable. Cette structure possède un mouvement induit par une force qu'elle subit ou par le déplacement de la caméra. Ce mouvement, qu'il soit issu d'une force interne, d'une force externe ou induit par la caméra, peut être déterministe ou stochastique.*

Les textures dynamiques sont constituées de modes, pouvant se superposer, caractérisés par des phénomènes spatiaux et temporels répétitifs.

La définition posée n'est peut être pas complète mais possède l'avantage de décrire plus profondément la notion de textures dynamiques. Avant d'étudier les approches permettant de caractériser les textures dynamiques, nous parlerons des bases de données existantes de textures dynamiques.

1.1.3 Bases de données existantes

Trois d'entre elles sont principalement utilisées : MIT, UCLA et DynTex. Nous présentons ci-après chacune de ces bases et leurs avantages.

Base de données du MIT La première base de données apparaît en 1996 avec l'article de Szummer et Picard [Szummer 96b]. Il s'agit d'une compilation de 25 séquences d'images en niveaux de gris et d'une séquence d'images couleur. L'ensemble de ces vidéos est disponible librement (<http://alumni.media.mit.edu/~szummer/icip-96/>) et visible sur la figure 1.6.



FIG. 1.6 – Ensemble des textures dynamiques de la base de données MIT [Szummer 96b].

Toutefois, cette base de données, bien que pionnière, possède plusieurs défauts majeurs :

- les vidéos possèdent une faible résolution tant du point de vue spatiale (au maximum 170×115 pixels) que du point de vue temporelle (70 à 75 frames). De plus, elles ne possèdent pas toutes la même taille spatialement et temporellement.
- l'ensemble des textures dynamiques peuvent être regroupé en 10 classes. Toutefois, il y a peu d'occurrence pour chaque classe de textures dynamiques.
- certaines séquences d'images sont corrompues par des mouvements de caméra non désirés ou des problèmes d'illuminant.
- la texture dynamique occupe toute la durée de la séquence d'images. Il n'y a pas de mélange de plusieurs textures dynamiques, de rupture de texture (par exemple par l'apparition d'objets ou par des occlusions), ...

Cette base possède le grand avantage d'avoir été la première à être mise en place offrant à la communauté un même cadre expérimental.

Base de données de UCLA Une deuxième base de données de textures dynamiques a vu le jour en 2001 et est due à Saisan et Doretto [Saisan 01]. Cette base, présentée sur la figure 1.7, permet de combler une partie des lacunes rencontrées avec la base de données MIT.

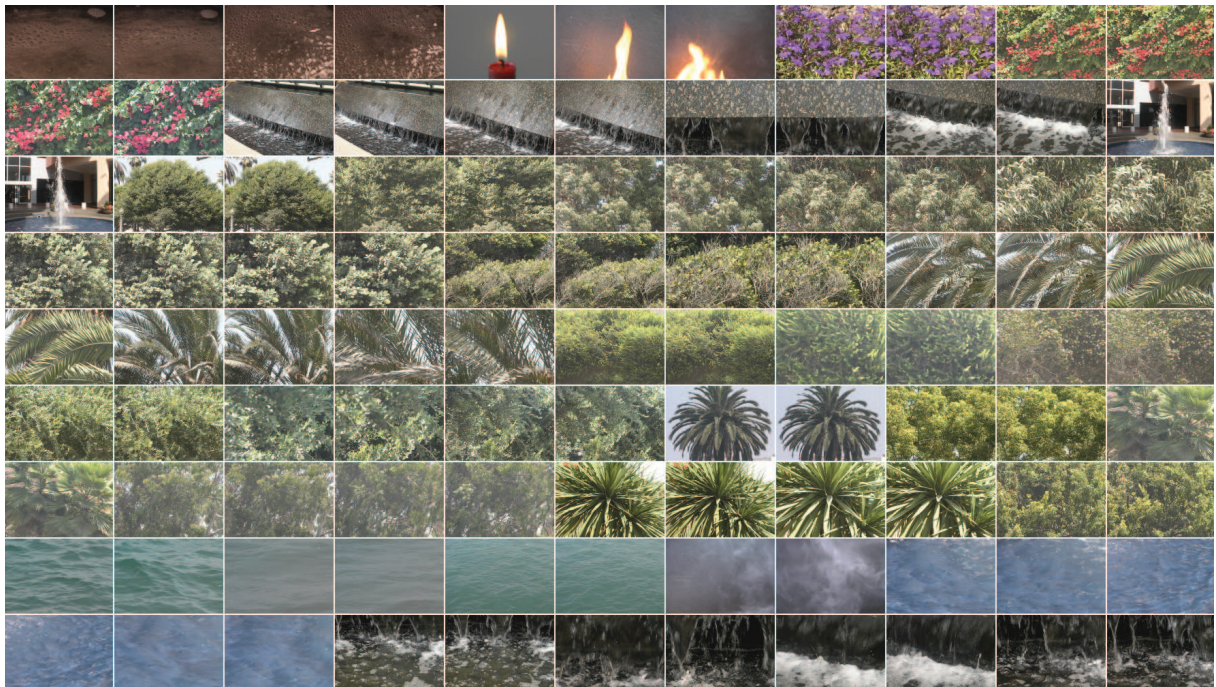


FIG. 1.7 – Ensemble des textures dynamiques de la base de données UCLA [Saisan 01].

- les vidéos sont de faibles résolutions (160×110 pixels avec 150 frames) mais sont en couleur, et toutes de la même taille.
- les auteurs proposent 10 classes présentant plusieurs individus par classe. Nous pouvons d'ailleurs observer des textures dynamiques à différentes échelles.

- les séquences d'images sont acquises dans de bonnes conditions (pas de mouvements de caméra non désirés, pas de problème d'illuminant, ...).
- comme pour la base de données MIT, la composante texture dynamique ne contient qu'une texture. Les séquences ne sont donc pas perturbées par des interférences de dynamiques, ce qui est un avantage pour l'étape d'apprentissage.

Cette base de données est également mise à disposition librement, il suffit de contacter les auteurs pour l'obtenir.

Base de données DynTex Par la suite, une autre base de données, DynTex [Péteri 10b], a été créée afin d'offrir un ensemble plus complet de séquences d'images présentant des textures dynamiques. Cette base, présentée en détail dans [Péteri 10b] et intégralement visible sur la figure 1.8 possède les caractéristiques suivantes :

- les vidéos possèdent une bonne résolution aussi bien spatiale que temporelle (720×576 pixels avec au moins 250 frames) et sont en couleur.
- la base possède une large variété de textures dynamiques (679 séquences d'images) appartenant à de nombreuses classes avec beaucoup d'occurrences. Toutefois, à l'heure actuelle seulement la moitié de la base de données a été étiquetée.
- les vidéos sont disponibles en format brut (entrelacé) ou compressé (desentrelacé)
- les conditions d'acquisition ont été parfaitement contrôlées (certaines textures dynamiques sont filmées avec des effets de caméra pré-établis)
- les séquences d'images proposent de nombreux cas de figure : perturbations des textures dynamiques par l'apparition d'objets ou d'occlusions, interférence entre plusieurs textures dynamiques, ...

Cette base de données est accessible librement à l'adresse <http://projects.cwi.nl/dyntex/>, et nécessite seulement une inscription en ligne.

Le graphique de la figure 1.9, illustre la proportion des principales publications vis-à-vis de la base utilisée :

- la base de données MIT a été beaucoup utilisée. Elle fut en effet la première construite et a donc fait l'objet de nombreuses expérimentations. Elle sert souvent de cadre de référence.
- bien que récente, la base de données DynTex commence à être utilisée régulièrement et de part sa richesse on peut penser qu'elle deviendra une base de référence.
- la base de données UCLA est peu utilisée. Elle offrait pourtant en 2001 un bien meilleur cadre expérimental que la base de données MIT.
- de nombreux auteurs utilisent encore des bases de données non disponibles ou trop spécifiques ne permettant pas de reproduire leur cadre expérimental.

Comme évoqué dans la section précédente, le cadre applicatif des textures dynamiques est très riche. Dans la plupart des applications, la caractérisation des textures dynamiques par l'extraction de caractéristiques est une étape fondamentale.

Les approches courantes de caractérisation de textures 2D impliquent l'extraction de descripteurs comme la couleur, la forme, la régularité, la périodicité, ...

Extraire ce type de vecteur caractéristique sur chaque frame de la séquence revient à utiliser uniquement l'information spatiale et à ignorer la dynamique propre à la texture temporelle. Cette approche est donc déconseillée pour réaliser une caractérisation pertinente. Il faut donc



FIG. 1.8 – Ensemble des textures dynamiques de la base de données *DynTex* [Péteri 10b].

soit étendre les techniques de caractérisation de la texture 2D à la texture dynamique, soit en trouver de nouvelles.

Dans la section suivante, nous effectuons une synthèse des approches de la littérature cherchant à caractériser les textures dynamiques par des descripteurs.

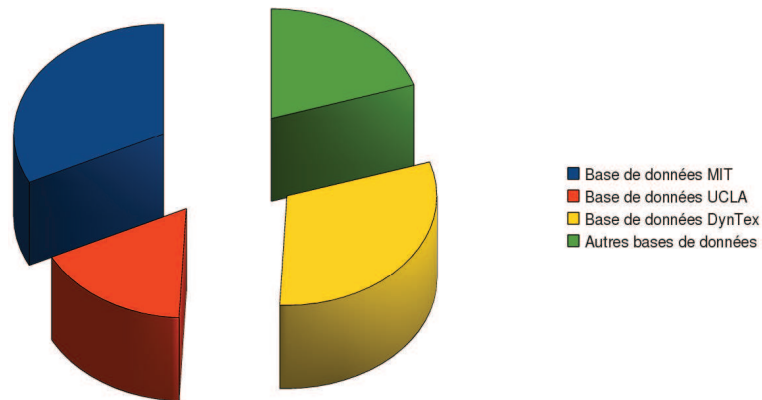


FIG. 1.9 – Répartition des principales publications vis-à-vis de la base de données utilisée dans les expérimentations.

1.2 Caractérisation de textures dynamiques

A partir de la littérature, nous proposons une taxonomie des approches cherchant à extraire des caractéristiques des textures dynamiques selon quatre groupes. Ces approches reposent sur :

- le mouvement ;
- les propriétés géométriques de l'espace spatio-temporel ;
- des transformées spatio-temporelles ;
- des modèles mathématiques.

Le graphique de la figure 1.10 montre les proportions de ces différentes approches rencontrées dans la littérature.

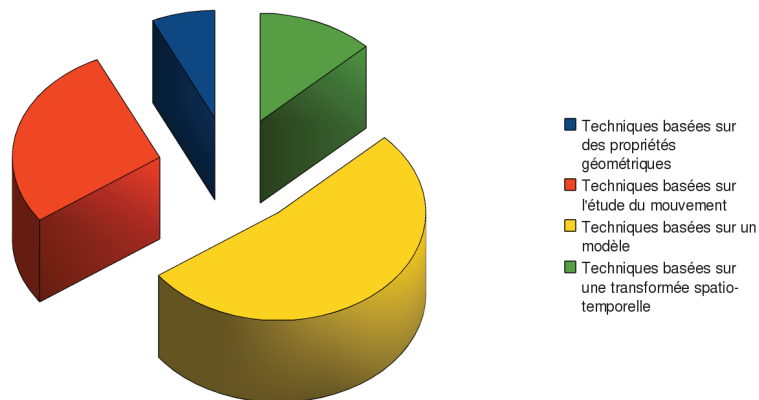


FIG. 1.10 – Répartition des principales publications selon les différentes approches de caractérisation.

Les approches reposant sur les modèles mathématiques sont les plus rencontrées. Elles possèdent l'avantage de prendre en compte uniquement le comportement dominant (associé au modèle) des textures, ce qui simplifie l'interprétation des scènes complexes. Les paramètres du modèle forment souvent le vecteur caractéristique.

Les techniques fondées sur l'étude du mouvement sont également fréquemment utilisées. Une texture est dynamique à partir du moment où un mouvement est présent, ainsi l'étude de celui-ci semble s'avérer être une voie pertinente.

Les approches fondées sur la géométrie ou des transformées spatio-temporelles ont été utilisées beaucoup plus rarement.

Pour chacune de ces approches, nous étudions dans la section suivante leurs avantages et leurs limites. Nous insisterons tout particulièrement sur les caractéristiques extraites des textures dynamiques.

Nous avons choisi de les présenter dans l'ordre de leur apparition dans la littérature.

1.2.1 Approches fondées sur le mouvement

La première étude qui traite des textures dynamiques a été réalisée par Nelson et Polana dans [Nelson 92, Polana 92, Polana 97]. Dans leur approche, intitulée *Spatial Feature based Texture Recognition* (SFTR), les auteurs utilisent une mesure partielle du mouvement à estimer, le flot normal, pour les raisons suivantes :

- il est très rapide à calculer et il peut être directement estimé sans utiliser de schéma itératif [Horn 81].
- il contient les informations de texture spatiales et temporelles.

Le champs de mouvement obtenus à l'aide du flot normal est ensuite analysé par un ensemble de descripteurs présentés dans la table 1.1.

Descripteur	Obtention
Non-uniformité du champ de direction	Calculée à l'aide d'un histogramme des directions en cumulant les différences entre cet histogramme et une distribution uniforme
Coefficient inverse de la variance	Ratio entre la moyenne et l'écart-type des amplitudes du flot normal
Divergence positive	Caractérise le flux de mouvement qui sort des "sources"
Divergence négative	Caractérise le flux de mouvement qui rentre dans les "puits"
Courbure positive	Courbure dans le sens positif
Courbure négative	Courbure dans le sens négatif
Homogénéité spatiale	Différences statistiques directionnelles dans une direction donnée (horizontale, verticale, diagonale positive et négative)

TAB. 1.1 – Ensemble des descripteurs utilisés par l'approche de [Polana 92].

Dans leur approche, illustrée par la figure 1.11, Nelson et Polana [Nelson 92, Polana 92] s'attachent à rendre leurs descripteurs invariants en échelle, en rotation et en translation. Les vecteurs caractéristiques ainsi calculés sont ensuite utilisés pour la classification de séquences d'images.

Cependant cette approche possède quelques limitations (exposées en détail dans [Peh 02])

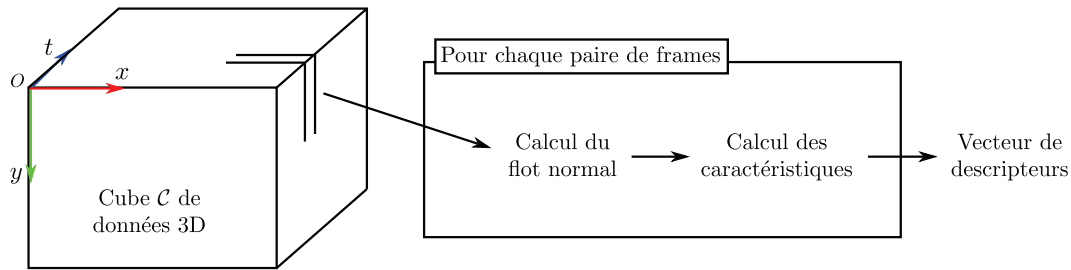


FIG. 1.11 – Illustration de la méthode de [Polana 92].

principalement dus à son étude mettant trop fortement l'accent sur les informations spatiales :

- par construction, le flot normal est fortement corrélé avec la structure spatiale de la texture dynamique ; ainsi des textures dynamiques où l'information temporelle est dominante seront mal discriminées (par les informations spatiales).
- le flot optique, et par conséquent le flot normal prennent comme hypothèse la conservation de la luminance, ce qui, de part leur éventuelle déformation, peut ne plus être valable dans le cadre de certains types de textures dynamiques.

Toutefois, comme nous verrons dans les prochains paragraphes, l'influence de ces premiers travaux est encore présente dans des articles récents.

Fablet et Bouthemy ont publié un grand nombre d'études [Bouthemy 98, Fablet 99, Fablet 00, Fablet 01b, Fablet 01a, Fablet 02, Fablet 03] sur la reconnaissance de textures dynamiques à l'aide du flot normal. Par leur approche, intitulée *Spatio-Temporal Clique Neighborhood* (STCN), ils cherchent, par rapport aux travaux précédents, à mieux prendre en compte l'aspect temporel, et ainsi permettre de corriger certains défauts rencontrés par l'approche de Nelson et Polana.

Dans [Bouthemy 98, Fablet 99], ils introduisent en premier le concept de matrice de co-occurrence temporelle qui mesure la probabilité de co-occurrence de deux flots normaux séparés d'un intervalle de temps pour une zone spatiale donnée de la texture dynamique. Plus précisément, dans [Bouthemy 98], trois intervalles de temps différents sont fixés (1, 4 et 8 frames), puis les descripteurs standards de la matrice de co-occurrence (moyenne, variance, second moment angulaire et contraste) sont utilisés pour discriminer des séquences d'images dont certaines contiennent des textures dynamiques.

Les auteurs développent une technique beaucoup plus sophistiquée [Fablet 01b, Fablet 01a, Fablet 02, Fablet 03] qui prend en compte plus directement les informations spatiales dans le calcul de la matrice de co-occurrence temporelle. Pour cela, les auteurs capturent les statistiques de co-occurrence en s'appuyant sur un modèle temporel multi-échelles de Gibbs.

Huit classes, dont cinq représentatives de textures dynamiques sont utilisées dans les expérimentations [Fablet 01b, Fablet 01a, Fablet 03]. Suite à une étape d'apprentissage (comprenant 80 séquences d'images), une estimation selon le maximum de vraisemblance est obtenue pour chaque classe. 376 séquences d'images sont ensuite classées à l'aide de la règle du maximum de vraisemblance et permet l'obtention d'un taux de reconnaissance de 87.25%.

Concernant cette approche, les cas suivants semblent toutefois poser problème :

- pour certaines périodicités spatiales ou temporelles des textures dynamiques (l'étude étant faite pour des intervalles de temps de 1, 4 ou 8 frames)
- pour une texture dynamique non présente sur la totalité l'ensemble de la vidéo.
- si plusieurs textures dynamiques sont présentes simultanément.

Peh et Cheong [Peh 99, Peh 02], en s'appuyant sur les travaux de Nelson et Polona et ceux de Fablet et Bouthemy, ont développé l'approche intitulée *Synergizing Spatial and Temporal Features* (SSTF) alliant les aspects temporel et spatial des textures dynamiques.

Pour cela, les auteurs commencent par effectuer une analyse spatio-temporelle des objets en mouvement en estimant l'amplitude et la direction du flot normal. Ces résultats sont superposés sur la fonction d'intensité de la séquence d'images. Séparément, celles-ci sont ensuite superposées frame par frame pour construire ce que les auteurs appellent des cartes d'amplitude (EMP) et de direction (EDP). Les cartes ainsi générées sont étudiées par des outils classiques d'analyse de textures (cf. figure 1.12).

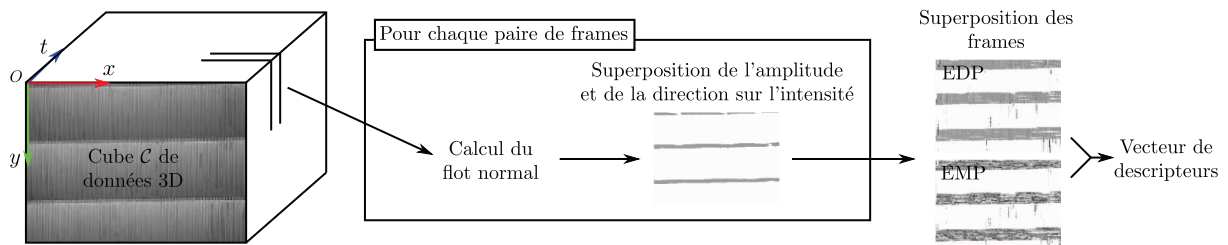


FIG. 1.12 – Illustration de la méthode de [Peh 99].

Les descripteurs extraits sont des éléments de la matrice de co-occurrence comme dans les études de Fablet et Bouthemy, et des éléments de nature fréquentielle comme l'énergie calculée selon un angle donné (45° et 135°). Ils sont résumés dans la table 1.2.

Descripteur	Obtention
Inertie	Inertie de la matrice de co-occurrence du graphique EDP
Moyenne	Moyenne de la carte EDP
Corrélation	Corrélation de la matrice de co-occurrence du graphique EMP
Contraste	Contraste de la matrice de co-occurrence du graphique EMP
Angle 45°	Énergie calculée selon l'angle 45° dans l'analyse de Fourier du graphique EDP
Angle 135°	Énergie calculée selon l'angle 135° dans l'analyse de Fourier du graphique EMP

TAB. 1.2 – Ensemble des descripteurs utilisés par l'approche *Synergizing Spatial and Temporal Features* (SSTF).

Les descripteurs de Peh et Cheong [Peh 99, Peh 02] sont rapides à calculer et invariants en rotation et en translation.

Les expérimentations (détaillées très précisément dans [Peh 02]) sont réalisées sur 470 séquences d'images réparties selon 10 classes. La méthode de classification employée est l'algorithme

des k -moyennes non-supervisées. Les auteurs obtiennent avec les 6 descripteurs de la table 1.2 un taux de réussite de 81.49%. Leur approche est meilleure que celle de Nelson et Polana [Nelson 92] qui obtient un score de 65.75% et de celle de Fablet et Bouthemy [Bouthemy 98] qui obtient seulement 48.94%. L'ensemble des expérimentations est réalisé de manière non supervisée, c'est l'extraction des paramètres qui est ici étudiée.

Le principal défaut de cette approche est que certaines caractéristiques de la texture dynamique peuvent être perdues. En effet, la superposition des frames pour l'obtention des cartes d'amplitude (EMP) et de direction (EDP) sur un temps long peut faire perdre des informations spatiales et temporelles locales qui seront masquées par les grandes tendances temporelles. Pour atténuer ce problème, les auteurs ont mis en place un paramètre permettant de changer la taille de la fenêtre d'étude. Toutefois, cette taille est difficilement paramétrable.

Dans des études plus récentes [Péteri 05, Péteri 04], Péteri et Chetverikov proposent une autre approche, illustrée par la figure 1.13, pour combiner l'information temporelle et spatiale des textures dynamiques.

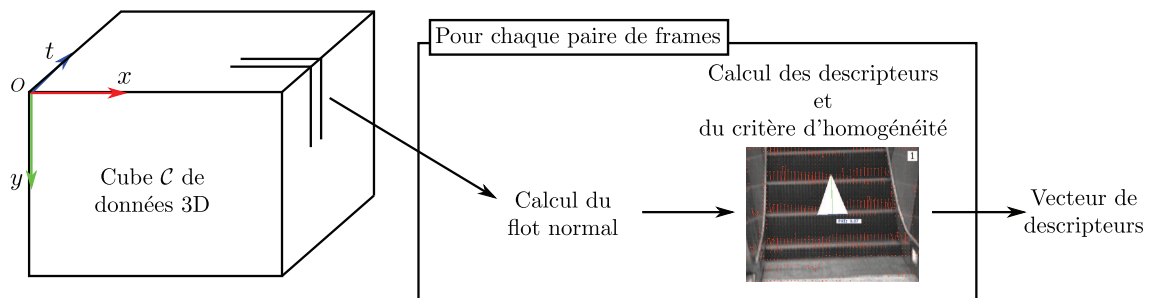


FIG. 1.13 – Illustration de la méthode proposée par Péteri et Chetverikov [Péteri 05, Péteri 04].

Les auteurs proposent ainsi un vecteur de six descripteurs (présentés dans le tableau 1.3) : trois sont similaires à ceux de Nelson et Polana [Nelson 92] (la divergence, la courbure et la *peakiness*) et trois sont nouveaux. Le premier permet de juger de l'homogénéité en orientation du flot normal (cf. [Péteri 04] pour plus de détails), les deux autres permettent d'évaluer la régularité de la texture (cf. [Chetverikov 00, Chetverikov 06]).

Descripteur	Obtention
Divergence	Moyenne sur le flot normal de la divergence
Courbure	Moyenne sur le flot normal de la courbure
<i>Peakiness</i>	Moyenne des amplitudes divisée par l'écart-type des amplitudes du flot normal
Homogénéité d'orientation	Norme de la somme des vecteurs du flot normal divisée par la somme des normes des vecteurs du flot normal
Moyenne de MRA	Moyenne des maxima de périodicité
Variance de MRA	Variance des maxima de périodicité

TAB. 1.3 – Ensemble des descripteurs utilisés par l'approche de Peteri et Chetverikov [Péteri 05, Péteri 04].

Les auteurs ont pris soin de sélectionner des descripteurs invariants en rotation et en translation. À l'aide des vecteurs caractéristiques présentés et de la méthode du *leave-one-out*, un taux de reconnaissance de 93.8%, sur un ensemble de 80 séquences d'images réparties en 10 classes de la base de données MIT, a été obtenu. Les six descripteurs sont nécessaires pour obtenir le meilleur taux de reconnaissance.

Dans une étude très récente [Péteri 10a], Péteri utilise trois de ces descripteurs (la divergence, la courbure et l'homogénéité directionnelle) dans le but de réaliser le suivi de textures dynamiques. L'auteur propose d'utiliser comme critère de vraisemblance le mouvement intrinsèque de la texture dynamique à suivre. La méthode proposée permet ainsi de suivre des textures dynamiques avec succès là où un filtre à particule basé sur la distribution de couleurs échoue.

Plutôt que de combiner l'aspect spatial et temporel, Rahman et Murshed [Rahman 04a, Rahman 04b, Rahman 05, Rahman 07b] proposent d'établir des descripteurs caractérisant les deux domaines séparément à travers leur approche *Optimal Time-Space Ratio technique* (OTSR).

Les auteurs partent du constat qu'une texture dynamique est composée d'une dimension temporelle et de deux dimensions spatiales. Ainsi, ils proposent de calculer une matrice de co-occurrence par dimension. Pour cela, ils considèrent la séquence d'images comme un cube de données 3D. La première, appelée matrice de co-occurrence temporelle est calculée classiquement selon l'axe temporel. Par contre, les deux autres, appelées matrices de co-occurrences spatiales, sont calculées, respectivement, selon l'axe x et l'axe y (cf. figure 1.14).

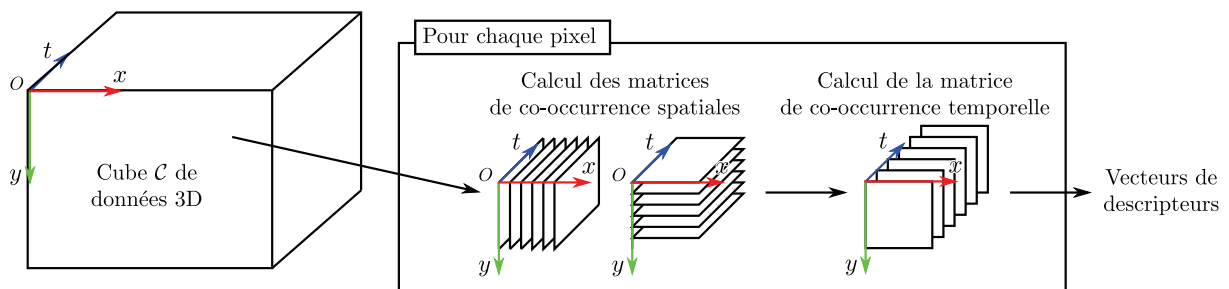


FIG. 1.14 – Illustration de la méthode de [Rahman 04a].

Dans [Rahman 07b], de multiples expérimentations sont réalisées. Les auteurs comparent leur approche (OTSR) à celles de la littérature (SFTR, SSTF et STCN) sur deux bases de données différentes (MIT et DynTex).

Sur la base de données MIT, dans les mêmes conditions expérimentales que [Péteri 04] (80 séquences d'images réparties en 10 classes) les auteurs obtiennent à l'aide de l'algorithme des k plus proches voisins, un taux de reconnaissance de 97.62% alors que l'approche SFTR arrive à obtenir 99.21%. La différence entre les deux méthodes est minimale (environ 1.5%), par contre, l'approche OTSR est 9.8 fois plus rapide que l'approche SFTR.

Sur la base de données DynTex, la même constatation est effectuée : en se plaçant dans les mêmes conditions que [Fazekas 05] (26 classes de textures dynamiques) et toujours en utilisant l'algorithme des k plus proches voisins, ils obtiennent un taux de reconnaissance de 93% alors que l'approche SFTR donne 95.5%. La différence est encore minimale.

Rahman et Murshed, dans [Rahman 07a], mettent en évidence un autre apport de la caractérisation marginale. Les auteurs montrent que les matrices de co-occurrences d’une séquence d’images contenant deux textures dynamiques présentes chacune à un certain pourcentage, sont les mêmes matrices que si l’on calcule séparément celles-ci sur les deux textures dynamiques puis qu’on les somme en respectant une pondération égale à leurs pourcentages. Les auteurs illustrent leur méthode par une expérimentation dans laquelle ils reconnaissent différentes séquences d’images contenant deux textures dynamiques présentes en différents proportions.

La dernière approche, illustrée par la figure 1.15, qui s’appuie sur l’étude de mouvement est une étude originale de Lu et Xie [Lu 05]. Les auteurs utilisent le flot optique total plutôt que le flot normal pour caractériser une texture dynamique.

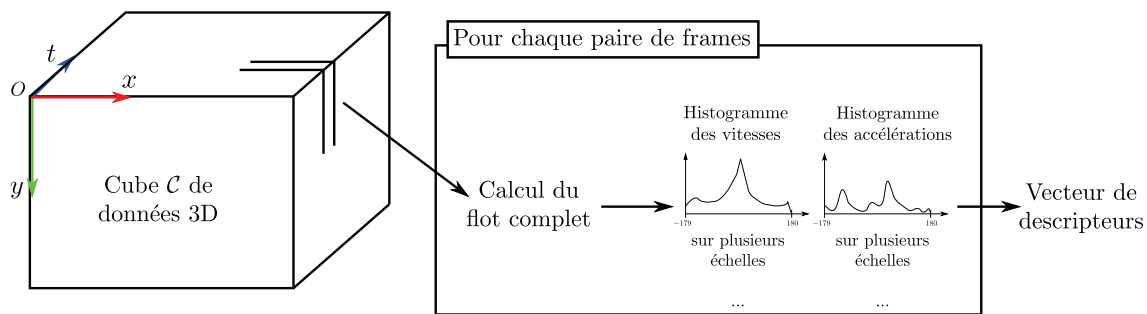


FIG. 1.15 – Illustration de l’approche de Lu et Xie [Lu 05].

Par rapport au flot normal, le flot optique total permet l’étude de l’accélération. Pour obtenir le flot optique total, les auteurs utilisent une technique s’appuyant sur un tenseur de structure 3D et une minimisation d’une fonction d’énergie.

A partir des champs de vitesse et d’accélération, des histogrammes angulaires multi-résolutions (calculés via un noyau gaussien 3D) sont utilisés. Une mesure de distance entre les histogrammes de deux textures dynamiques permet de les différencier. La performance de l’algorithme est illustrée sur 7 séquences d’images (deux de fontaines, trois d’eau, une d’arbre et une de feu) en étudiant visuellement les distances entre les différents histogrammes.

Cette approche est invariante en rotation et prends en compte, par l’utilisation de plusieurs noyaux gaussien, l’échelle.

Suite à cette étude, Fazekas et Chetverikov [Fazekas 05] ont proposé une comparaison entre le flot normal et le flot optique total pour la reconnaissance de textures dynamiques. Dans celle-ci, les auteurs effectuent la comparaison des deux approches à l’aide de la méthode *leave-one-out* sur deux jeux de séquences d’images : la base de données MIT et DynTex.

Dans le cas de la base de données MIT (10 classes de textures dynamiques), l’approche utilisant le flot normal est meilleure que celle utilisant le flot optique total puisque celle-ci obtient un taux de reconnaissance de 95.4% contre 93.8% pour l’autre. Par contre, dans le cas de la base de données DynTex (26 classes de textures dynamiques), les résultats sont inversés. En effet, l’approche avec le flot optique total obtient un taux de reconnaissance de 98.1% contre 93.4% pour l’approche utilisant le flot normal.

Ainsi lorsque le nombre de classes de textures dynamiques augmente le flot optique total s'avère plus discriminant. Pour confirmer cela, Fazekas et Chetverikov [Fazekas 07b, Fazekas 07a] ont comparé, dans le même cadre expérimental (26 classes de textures dynamiques issues de la base de données DynTex), quatre approches utilisant le flot optique total contre une utilisant le flot normal. Les quatre premières apportent un meilleur taux de reconnaissance.

Dans toutes ces études reposant sur l'estimation du mouvement, le flot normal ou le flot optique total est utilisé avec comme hypothèse sous-jacente, la conservation de la luminance. Cette hypothèse s'avère pertinente dans le cas du suivi d'un objet, mais ne l'est pas toujours dans le cas d'un fluide ou d'un gaz. C'est pourquoi, dans deux études très récentes [Amiaz 07, Fazekas 09], Amiaz et Fazekas examinent d'autres hypothèses : conservation du gradient, de la couleur ou une conservation de masse (contrairement au suivi d'objets, les auteurs autorisent la luminance à se diffuser parmi les pixels voisins). Après un premier travail où ils montrent que la conservation de masse et de la couleur sont les deux hypothèses les mieux vérifiées pour les textures dynamiques, les auteurs mettent en place une méthode de segmentation spatio-temporelle quasi temps-réel.

Après cette synthèse des travaux de caractérisation des textures dynamiques par des approches fondées sur le mouvement, nous nous intéressons dans la section suivante aux approches s'appuyant sur la définition d'un modèle mathématique adapté aux textures dynamiques.

1.2.2 Approches s'appuyant sur un modèle mathématique

Les textures dynamiques relèvent de phénomènes complexes. Ainsi, de nombreux auteurs ont cherché à les modéliser pour, d'une part, les utiliser en synthèse, et d'autre part, pour l'utilisation des paramètres du modèle comme descripteurs. Il faut préciser qu'un modèle particulièrement efficace dans le cadre de la synthèse ou de la compression n'a pas de raison d'être un bon modèle pour la classification. Dans les approches par modèle, nous distinguons cinq grandes familles :

- les modèles basés sur des lois physiques, on peut citer par exemple les travaux de Kung et Richards [Kung 88]. Nous ne rentrerons pas plus en détails sur ce type de modèle car ils sont souvent établis pour une seule texture dynamique et ne présentent donc que peu d'intérêt dans le contexte de notre travail, puisqu'ils ne permettent pas la comparaison entre différentes classes de textures dynamiques ;
- les modèles auto-régressifs spatio-temporels ;
- les modèles auto-régressifs à moyenne mobile et les approches dérivées ;
- les modèles utilisant la réponse impulsionnelle des variables d'états cachés ;
- les autres modèles.

Modéliser une séquence d'images de textures dynamiques en utilisant un modèle auto-régressif spatio-temporel (STAR) a été proposé en premier par Szummer et Picard [Szummer 96a, Szummer 96b]. Il s'agit d'une extension 3D des modèles auto-régressifs (AR) utilisés pour la reconnaissance et la synthèse de textures 2D. La luminance de chaque voxel est estimée à l'aide d'une combinaison linéaire de la luminance des voxels choisis dans un voisinage spatio-temporel, à laquelle se superpose un bruit. Les paramètres du modèle auto-régressif spatio-temporel (STAR) sont déterminés en minimisant l'erreur quadratique moyenne de prédiction.

Les modèles auto-régressifs spatio-temporels ont permis de modéliser efficacement les textures dynamiques comme la fumée, les rivières, ... présentant des mouvements de translation. Toutefois, ils échouent dans la capture des mouvements rotationnels, d'accélération, et requièrent un temps de calcul long du au nombre important de paramètres du modèle.

Afin de pallier aux difficultés rencontrées par le modèle de Szummer et Picard, Doretto *et al.* ont proposé une série d'études utilisant un autre modèle [Soatto 01, Doretto 03a, Doretto 05a], le *Linear Dynamic System* (LDS), illustré par la figure 1.16. Celui-ci opère sur un modèle de type auto-régressif à moyenne mobile (ARMA). L'apprentissage du modèle est réalisé selon deux méthodes : celle du maximum de vraisemblance et celle de la minimisation de l'erreur de la variance (pour plus de détails, le lecteur pourra se référer à [Doretto 05a]).

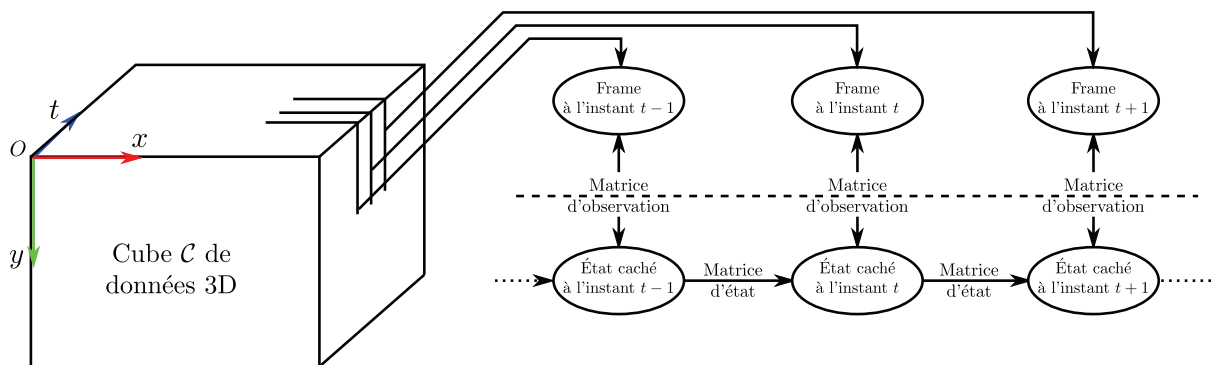


FIG. 1.16 – Illustration de la méthode utilisant un modèle auto-régressif à moyenne mobile (ARMA).

Par la suite, ce modèle est utilisé dans de nombreuses applications :

- dans [Doretto 03a, Doretto 04], en utilisant des séquences d'images de la base MIT, les auteurs proposent à partir de l'apprentissage de 100 frames d'une séquence d'images, de synthétiser une séquence de 300 frames.
- les auteurs de [Doretto 03b] proposent une méthode segmentant une texture dynamique de manière spatio-temporelle à l'aide des *level sets* et du LDS.
- une fois les paramètres du modèle obtenus, les auteurs de [Doretto 03c, Doretto 03d] proposent de les modifier afin de changer et de contrôler les propriétés spatio-temporelles des textures dynamiques.
- dans [Saisan 01], dans le but de classer des textures dynamiques, trois métriques sur l'espace des modèles sont définies. Le taux de reconnaissance maximum obtenu par les auteurs est de 89.5% pour 40 textures dynamiques appartenant à 10 classes de la base de données UCLA.

Le large panorama applicatif et la qualité des résultats obtenus montrent la pertinence des approches fondées sur un modèle mathématique. Toutefois, le grand nombre de paramètres du modèle nécessite un temps de calcul important.

L'autre point clé de la méthode est la définition de la métrique intervenant dans la comparaison des paramètres de deux modèles. Afin d'étudier son influence, d'autres études ont été conduites :

- Chan et Vasconcelos introduisent deux métriques : la *Kulbach-Leibler divergence* [Chan 05b] et la distance de Martin [Chan 07]. Pour la base UCLA, le taux de reconnaissance maximum obtenu est de 96% sur 50 vidéos.
- Vishwanathan et Vidal utilisent les noyaux de Binet-Cauchy avec les *Support Vector Machine* pour le classement de textures dynamiques de la base MIT [Vishwanathan 07].
- Woolfe et Fitzgibbon, dans [Woolfe 06], utilisent les distances de Chernoff et les distances entre coefficients du *cepstrum* et obtiennent un taux de reconnaissance de 60% sur 200 vidéos de la base UCLA.

Dans les études précédentes, les séquences d'images ne contiennent qu'une seule texture dynamique. Si plusieurs textures dynamiques sont présentes, l'approche LDS les formalisera dans un seul et même modèle. Pour pallier à ce problème, différentes approches ont été proposées :

- Doretto [Doretto 05b, Doretto 06] propose de coupler son modèle aux *Active Appearance Models* (AAM). Ceux-ci permettent de prendre en compte les structures spatio-temporelles des séquences de textures dynamiques.
- Chan et Vasconcelos, dans [Chan 05a, Chan 06, Chan 08, Chan 09] proposent de voir une composante texture dynamique comme la composition de plusieurs textures dynamiques interagissant entre elles. Ainsi, ils proposent un modèle, illustré par la figure 1.17, constitué de plusieurs chaînes d'états cachés qui représentent les différentes textures dynamiques présentes dans une séquence d'images. L'apprentissage des différents paramètres du modèle est réalisé à l'aide d'un algorithme EM.

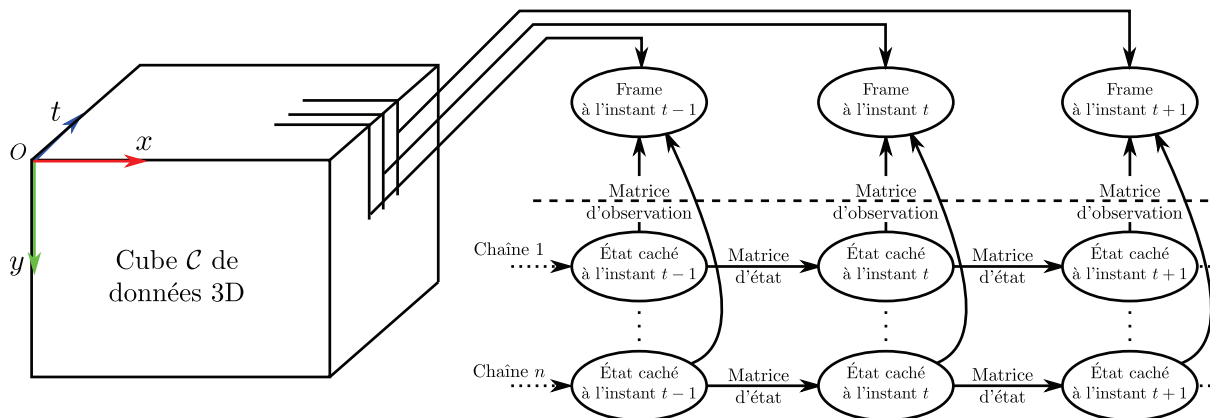


FIG. 1.17 – Illustration de la méthode de [Chan 05a].

Leur approche est ensuite utilisée dans de nombreuses applications : segmentation de plusieurs textures dynamiques [Chan 05a, Chan 06, Chan 08, Chan 09], classification d'une base de données de trafic routier [Chan 05a, Chan 08].

- Cooper et Liu [Cooper 06] proposent la même méthode que Chan et Vasconcelos excepté le fait que l'apprentissage du modèle s'effectue à l'aide de l'analyse en composantes principales généralisée (GPCA).

- Ravichandran et Chaudhry [Ravichandran 09], découpent une séquence d’images en plusieurs sous-cubes. Sur chacun de ces patches, un modèle ARMA est mis en place afin de créer ce que les auteurs appellent un *bag of system* (BoS). Cette approche est guidée par celle des *bag of features*. Cependant, la métrique euclidienne n’est plus adaptée aux BoS ce qui posent certaines difficultés que les auteurs résolvent.

Une autre des limitations de l’approche LDS est que le modèle s’appuie sur les variations de l’intensité des pixels et ne prend donc pas assez en considération le mouvement global. Afin de corriger ce défaut, plusieurs approches ont été proposées :

- Ghanem et Ahuja [Ghanem 07a, Ghanem 07b, Ghanem 08] se focalisent sur les variations de la phase de la transformée de Fourier. Leur modèle est testé dans des applications de compression et de synthèse, mais également de classification. Un taux de reconnaissance de 95.2% (alors que la méthode LDS n’obtient que 46.3%) a été obtenu sur 121 textures dynamiques distribuées sur 14 classes. Cependant, ces expérimentations ont été conduites sur une base construite à partir de celle du MIT et de DynTex.
- Zhou et Dong [Zhou 09] proposent de modéliser une texture dynamique à l’aide d’un *constrained Gaussian process latent variable model* (CGPLVM).

Par la suite, une étude de Yuan et Wen [Yuan 04] montre que l’approche LDS fonctionne très bien dans le cas où le comportement de celui-ci est oscillatoire. Dans le cas contraire, les auteurs ajoutent un asservissement dans le modèle permettant de corriger ce problème. Leur approche est illustrée par différents résultats de synthèse pertinents.

Le LDS et le modèle ARMA ont été, et restent, très populaires. Ceci est corroboré par le nombre de publications sur ce sujet. Toutefois, d’autres approches par modèle ont été étudiées. Fujita et Nayar [Fujita 03] modifient l’approche de [Soatto 01, Saisan 01] en utilisant la réponse impulsionnelle des variables d’état. Du point de vue de l’identification, la réponse impulsionnelle capture les propriétés intrinsèques de la dynamique. Leur méthode, illustrée par la figure 1.18, se divise en deux étapes :

- une étape d’apprentissage où la séquence d’images est divisée en sous-cubes permettant ainsi une analyse locale. Dans chacun de ces sous-cubes, un modèle auto-régressif à moyenne mobile (ARMA) est utilisé afin d’obtenir les différents paramètres. Le changement des réponses impulsionnelles des variables d’état permet de prédire l’évolution de la texture dynamique.
- puis la deuxième étape utilise cette prédiction afin de réaliser une estimation de la réponse impulsionnelle des variables d’état servant à mettre à jour les paramètres.

Cette méthode, suite à une période d’apprentissage de 100 frames, est testée avec succès pour la synthèse de 300 frames.

Dans [Wang 03], une approche originale par modèle est proposée. Celle-ci modélise une texture dynamique par une composition de plusieurs couches dont chacune d’elle appartient à une base de fonctions donnée. Dans leur approche, les auteurs utilisent la base de Fourier et de Gabor afin de synthétiser différentes textures dynamiques.

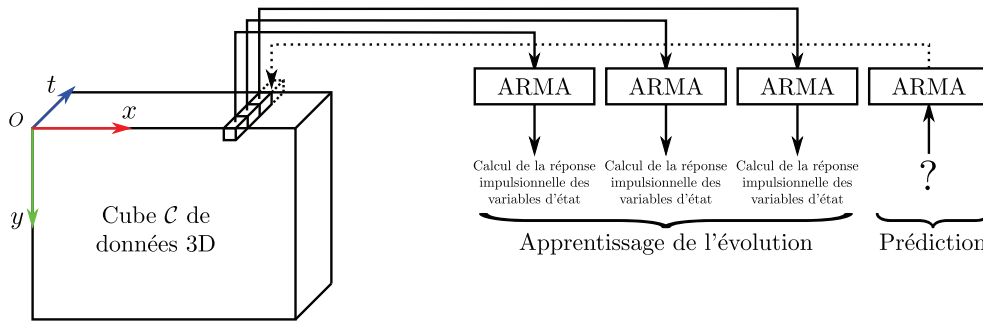


FIG. 1.18 – Illustration de la méthode utilisant la réponse impulsionnelle des variables d'état [Fujita 03].

1.2.3 Techniques utilisant la géométrie spatio-temporelle

Nous présentons maintenant plusieurs approches s'appuyant sur la géométrie spatio-temporelle des textures dynamiques.

La première étude est celle de Otsuka et Horikoshi [Otsuka 98]. Les auteurs considèrent qu'une texture dynamique peut être représentée par les structures spatio-temporelles qu'elle engendre dans l'espace spatio-temporelle. Ces structures forment ce que les auteurs appellent des trajectoires spatio-temporelles (des surfaces/motifs dans des cubes de données 3D) qui peuvent être suivies. Ainsi les auteurs représentent les trajectoires spatio-temporelles par des plans tangents à celles-ci en utilisant la transformée de Hough 3D. L'intersection des différents plans est représentative de la direction du mouvement (Cf. figure 1.19).

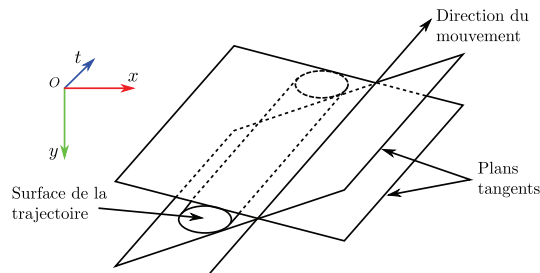


FIG. 1.19 – Plans tangents à une trajectoire définissant la direction spatio-temporelle du mouvement par l'approche de Otsuka et Horikoshi [Otsuka 98].

De ces plans, différentes caractéristiques peuvent être extraites et sont présentées dans le tableau 1.4.

L'approche proposée par Otsuka et Horikoshi [Otsuka 98] est très robuste face au bruit et aux occlusions. Toutefois celle-ci souffre d'un temps de calcul non négligeable. De plus, les descripteurs obtenus sont étroitement liés aux plans tangents. Ceux-ci sont construits à partir des surfaces des trajectoires qui sont difficiles à estimer.

Descripteur	Obtention
Direction des contours	Mesure la précision des directions des plans tangents dans le domaine spatial
Dispersion du contour	Intervalle entre les lignes tangentes du contour dans le plan image
Uniformité du champ	Mesure de la diversité des différentes directions du mouvement
Ratio mouvement rapide	Mesure la quantité de mouvement, d'apparition et de disparition d'objets
Longueur de la trajectoire	Caractérise le scintillement des objets

TAB. 1.4 – Ensemble des descripteurs utilisés par l'approche de Otsuka et Horikoshi [Otsuka 98].

Afin de corriger ce défaut, Zhong et Scarlaroff [Zhong 02] proposent une nouvelle approche. Un filtre gradient 3D est d'abord appliqué sur la séquence d'images afin d'extraire les contours 3D. A partir du vecteur gradient, un seuil est appliqué pour obtenir les surfaces des trajectoires. Cette méthode améliore la précision des surfaces extraites car elle incorpore l'information temporelle contrairement à la méthode de Otsuka et Horikoshi [Otsuka 98].

Les surfaces de trajectoires ainsi obtenus sont discrétisées à l'aide d'un ensemble de sous-cubes régulier. Pour chacun de ces sous-cubes, différents descripteurs sont extraits : la direction du plan tangent, l'amplitude du contour et la courbure. Le vecteur de caractéristiques de la texture dynamique est obtenu en effectuant la moyenne de ces descripteurs pour tous les sous-cubes.

Cette approche est ensuite testée sur 9 classes de textures dynamiques de la base MIT. Après une étape d'apprentissage effectuée à l'aide d'un modèle de mélange gaussien, les auteurs obtiennent un taux de reconnaissance de 98.6%, et de 89.9% lorsque la taille des cubes d'étude est plus faible.

D'autres approches proposent d'effectuer une étude plus globale de la géométrie spatio-temporelle. Les textures dynamiques présentant des répétitions du point de vue temporel, Schödl et Szeliski [Schodl 00], proposent d'étudier et de reconnaître la périodicité du phénomène étudié. Pour cela, ils calculent pour chaque frame de la séquence d'images, sa distance en norme ℓ_2 avec toutes les autres frames. A partir de cette matrice de distance, ils obtiennent une matrice de probabilités permettant de savoir à un instant donné, la probabilité de passer à une autre frame (cf. figure 1.20).

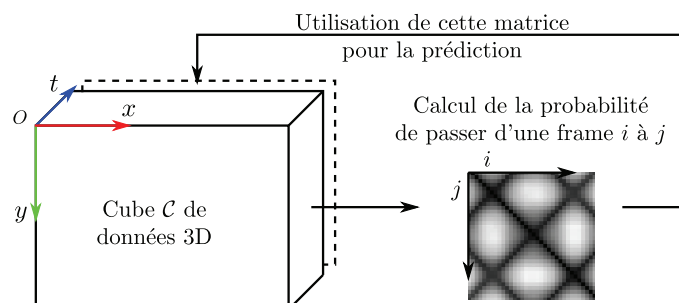


FIG. 1.20 – Illustration de la méthode de [Schodl 00].

Les auteurs mettent ensuite en place deux méthodes pour synthétiser des séquences d'images à partir d'un échantillon. La première méthode, très simple, consiste, à la dernière frame connue, à prendre le moindre risque en prenant le maximum de vraisemblance. On synthétise ainsi frame par frame. La deuxième méthode va quand à elle étudier la séquence d'images afin de trouver la suite la plus probable pour l'échantillon donné.

Dans [Schodl 00] de nombreuses applications sont évoquées (synthèse d'une séquence de feu, de drapeau, d'une cascade, d'un champ d'herbes, ...).

D'autres études, comme celles de [Zhao 06, Zhao 07a, Zhao 07b], proposent une approche étudiant localement la texture dynamique.

Pour cela, ils proposent d'étendre, à la texture dynamique, l'opérateur *Local Binary Pattern* (LBP) [Ojala 96, Ojala 02] utilisé couramment en analyse de textures 2D. Cette approche, intitulée *Volume Local Binary Patterns* (VLBP) et illustrée par la figure 1.21, se déroule, pour chaque voxel de la séquence d'images, en trois étapes :

- définition d'un parcours du voisinage de ce voxel (différents parcours sont étudiés dans [Zhao 06]). Pour la définition de ces parcours, on définit R le rayon du cercle sur une image, P le nombre de points pris sur ce cercle et L la distance entre deux frames.
- binarisation du voisinage par rapport à la valeur de ce voxel (si la valeur du voxel voisin est supérieure (respectivement inférieure) à la valeur du voxel que l'on observe, celui-ci est mis à 1 (respectivement 0)).
- en parcourant le voisinage, on obtient un vecteur constitué de 0 et de 1 représentatif de la texture dynamique en ce point.

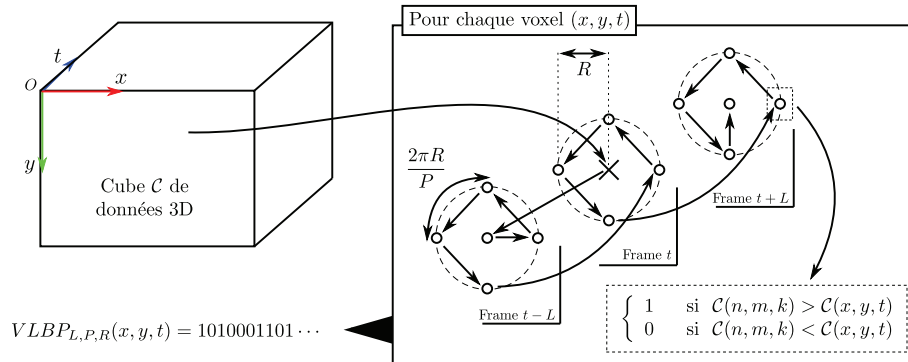


FIG. 1.21 – Illustration de l'opérateur *Volume Local Binary Patterns* (VLBP) de [Zhao 07b].

Des histogrammes des distributions de ce vecteur sont alors construits et servent de vecteurs caractéristiques. Ceux-ci sont, par nature, invariants au changement de luminosité. De plus, Zhao et Pietikainen ont pris soin de les rendre invariants en rotation [Zhao 07b]. Les auteurs définissent ensuite une mesure pour comparer les histogrammes entre eux.

Dans [Zhao 07b], des expérimentations de cette méthode sont proposées. Sur la base de données MIT (112 séquences d'images de textures dynamiques appartenant à 14 classes), en utilisant la méthode du *leave-one-out*, un taux de reconnaissance de 100% est obtenu. Ce taux de reconnaissance passe à 92%, en utilisant la base DynTex (26 classes différentes de textures dyna-

miques). Toutefois, ces résultats dépendent grandement du choix des paramètres de l'opérateur VLBP (R , L et P) qui ne sont pas des plus simples à régler, car sensibles à la classe de textures dynamiques.

De très bons taux de reconnaissance sont également obtenus avec cette méthode sur une base de données de visages [Zhao 07a].

L'opérateur *Local Binary Pattern* (LBP) est également utilisé dans [Ma 09] afin de reconnaître des activités sur des séquences de vidéo surveillance. Pour cela, Ma et Cisar calculent frame par frame les codes de chaque pixel (comme l'illustre la figure 1.21). Ensuite, les codes de la frame t sont connectés avec ceux de la frame $t + 1$ afin de construire un flux. Celui-ci décrit les mouvements intrinsèques de la texture dynamique.

Dans une étude récente, Gao et Xu [Gao 08] proposent d'étendre à la vidéo les histogrammes *Statistical Landscape Features* (SLF), eux aussi très utilisés en analyse de textures 2D. Pour cela, les auteurs proposent de définir un vecteur caractéristiques composé d'un descripteur de la structure spatiale de la texture dynamique concaténé à un descripteur représentant le mouvement interne de la texture dynamique.

L'apparence spatiale est caractérisée par la moyenne des histogrammes SLF obtenus sur chaque frame de la séquence d'images. L'information temporelle est, quant à elle, observée à l'aide de l'histogramme LMP qui correspond au cumul d'un indice de déplacement. Celui-ci est calculé pour chaque pixel de la séquence d'images : la valeur du pixel d'une frame est recherchée dans la frame suivante dans un voisinage d'une petite taille (généralement 5×5 pixels) en calculant la valeur absolue de la différence. La plus petite distance ainsi trouvée est codifiée par un indice représentant la position de ce pixel dans le voisinage. L'ensemble de ces indices est ensuite cumulé afin de créer un histogramme LMP.

L'association des deux histogrammes donne naissance à un vecteur caractéristique nommé par les auteurs *Extended Statistical Landscape Features* (ESLF) (cf. figure 1.22). Celui-ci est invariant en translation et robuste au changement de luminosité.

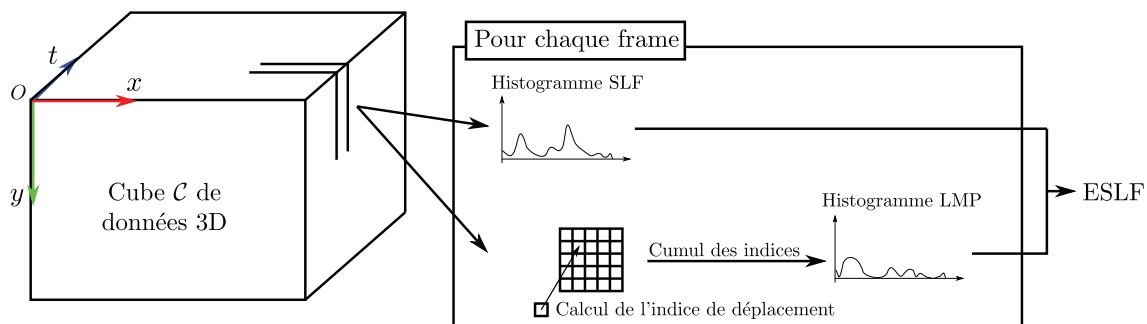


FIG. 1.22 – Illustration de la méthode de [Gao 08].

Afin de valider leur méthode, les auteurs utilisent 26 classes de textures dynamiques issues de DynTex, pour se placer dans le même cadre expérimental que [Zhao 07b]. Un taux de reconnaissance de 97.33% est obtenu contre 92% pour la méthode de référence.

Une approche similaire à la précédente a été récemment présentée par Derpanis et Wildes [Derpanis 10]. Dans celle-ci, les auteurs proposent une autre solution pour étudier les différentes directions d'une texture dynamique. Dans ce but, 27 masques d'étude sont définis selon la direction du mouvement et la vitesse de celui-ci. Pour chaque masque, l'énergie de la séquence d'images convoluée à un filtre dérivatif gaussien 3D du troisième ordre est calculée afin de caractériser les structures orientées selon cet angle. Les 27 directions permettent ensuite de calculer un histogramme d'énergie.

La reconnaissance des textures dynamiques est alors réalisée en effectuant un calcul de distance entre leurs histogrammes d'énergie respectifs. Les auteurs mettent ensuite en place une expérimentation sur la base UCLA conduisant à un taux de reconnaissance de 92.3%.

Dernièrement, une approche originale de Li et Hall [Li 08] représente une texture dynamique à l'aide d'un arbre structurel que l'on suit au cours du temps. Pour cela, les auteurs étudient la topologie d'une texture dynamique en calculant un arbre de *Maximally Stable Extremal Region* (MSER) pour chaque frame de la vidéo. Ensuite à l'aide d'une méthode de suivi, les auteurs relient les arbres des différentes frames entre eux et déterminent des caractéristiques. Pour prédire la suite du mouvement de la texture dynamique, les auteurs mettent en place le suivi d'un certain nombre de points d'intérêt de la texture dynamique et les comparant à leur prédiction. Les résultats obtenus sont très satisfaisants.

Cette méthode fonctionne très bien pour la caractérisation de textures dynamiques présentant une structure spatiale marquée telle que les arbres, les feuillages, les vols d'oiseaux, mais présente des difficultés pour les textures dynamiques comme l'eau, la fumée, les drapeaux.

Une étude récente [Kobayashi 09a] propose la reconnaissance de textures dynamiques à l'aide d'une méthode statistique. Pour cela, Kobayashi et Higuchi propose d'utiliser la méthode *Higher-Order Auto-Correlation* (CHLAC) étendue au temps dans [Kobayashi 09b]. Celle-ci consiste pour un cube de données, à effectuer une corrélation entre deux sous-cubes de celui-ci. Un jeu de 279 masques permet de sélectionner les différentes combinaisons possibles. Les auteurs effectuent ensuite une régression multiple afin d'estimer les caractéristiques de la texture dynamique.

Les expérimentations sont effectuées sur un ensemble de 18 séquences d'images d'ultra-son. Les temps de calcul de l'algorithme ne sont pas évoqués. Ceux-ci doivent être non négligeables, au vue du nombre d'auto-corrélation (279) à effectuer pour chaque sous-cube de la séquence d'images.

1.2.4 Approches fondées sur des transformées spatio-temporelles

La dernière approche concerne la caractérisation de textures dynamiques par des transformées spatio-temporelles. Deux études peuvent être distinguées : la première utilise la transformée en ondelettes et la seconde utilise la transformée de Fourier.

Dans la littérature, une seule étude utilise la transformée en ondelettes pour l'indexation de textures dynamiques [Smith 02]. Smith et Lin présentent une méthode multi-résolutions utilisant une transformée en ondelettes de Haar présentant la particularité de sous-échantillonner, à

chaque changement de résolution, deux fois le domaine spatial et deux fois le domaine temporel.

Les auteurs construisent ensuite leurs signatures en calculant l'énergie des sous-bandes de détails obtenue à chaque résolution de la transformée. Ils étudient également différentes combinaisons de l'information temporelle. Cependant, leur méthode est testée sur une petite base de données (45 vidéos appartenant à 6 classes) non disponible.

Nous discuterons de cette méthode plus en détails dans les chapitres suivants.

Dans plusieurs études récentes, la transformée de Fourier est utilisée pour segmenter des textures dynamiques. Li et Chen [Li 09b, Li 09a] calculent une transformée de Fourier 3D directement sur la séquence d'images. Les auteurs s'appuient sur le fait que la phase de la transformée de Fourier apporte souvent plus d'information que l'amplitude. Ainsi ils ne reconstruisent que la partie correspondante à la phase. Cette reconstruction est ensuite lissée à l'aide d'un filtre moyennneur, puis est seuillée par sa moyenne. La zone ainsi obtenue représente la texture dynamique.

Cette approche, sur des séquences d'images de DynTex, donne des résultats visuels très pertinents. Les résultats obtenus ne sont pas comparés avec ceux de la littérature [Doretto 03b] et cette méthode demande à être évaluée plus profondément. En effet, les séquences d'exemples choisies sont, à chaque fois, un fond statique et une texture dynamique bien marquée temporellement (la phase semble donc tout à fait pertinente dans ce cas). Il faudrait tester cette méthode sur des séquences d'images où plusieurs textures dynamiques sont présentes, où des mouvements de caméra importants perturbent celles-ci, ...

Conclusion du chapitre Dans la première partie de ce chapitre, après avoir étudié les différentes définitions données dans la littérature de la composante texture dynamique, nous proposons de la décrire comme la composition de plusieurs textures dynamiques caractérisées par différents modes.

Par la suite, nous effectuons une synthèse des méthodes caractérisant les textures indépendamment des domaines applicatifs. Nous avons remarqué qu'il existait quatre grands types d'approches qui reposent sur :

- le mouvement ;
- les propriétés géométriques de l'espace spatio-temporel ;
- des transformées spatio-temporelles ;
- des modèles mathématiques.

Nous pouvons constater que :

- de nombreux tests ont été réalisés sur une base de données de 26 classes de textures dynamiques issues de DynTex. Ces 26 classes sont construites à partir de 26 séquences d'images originales. Les individus représentatifs de ces classes sont obtenus en divisant la séquence d'images originale en 8 sous-séquences. A notre sens, cette approche présente un biais, car si on considère qu'une texture dynamique donnée est homogène pour l'ensemble de la séquence, les 8 séquences d'images possèdent alors des caractéristiques très proches. Finalement, les descripteurs extraits caractérisent la séquence d'images originale plutôt qu'une classe de textures dynamiques.
- le nombre de textures dynamiques utilisées dans les tests de classifications n'est pas très grand. En effet, soit la base de données ne contient pas assez d'éléments (c'est le cas des

bases MIT et UCLA), soit les auteurs restreignent leur choix (cas de la base DynTex). Ceci peut s'expliquer par le fait que l'étiquetage de DynTex n'est pas encore terminé, et n'est donc pas disponible. Dans nos expérimentations, nous prendrons soin de construire des classes comportant une grande variété de phénomènes.

- les méthodes s'appuyant sur des transformées spatio-temporelles ne sont pas très nombreuses. Ceci est liée probablement aux difficultés d'interprétation. Pourtant cette approche paraît pertinente puisque, d'un part, elle a déjà fait ses preuves dans le cadre de l'analyse de textures 2D, et que d'autre part, ces méthodes permettent de prendre en compte la notion d'échelle très importante en analyse de textures.

Dans le prochain chapitre, nous abordons le cadre de l'analyse multi-résolutions et étudierons plusieurs approches pouvant servir à caractériser les textures dynamiques.

CHAPITRE 2

Décompositions multi-échelles de séquences d'images

Décompositions multi-échelles de séquences d'images

Sommaire

2.1	Décompositions multi-échelles usuelles	54
2.1.1	Analyse multi-résolutions	56
2.1.1.1	Bases d'ondelettes et analyse multi-résolutions	56
2.1.1.2	Bancs de filtres	58
2.1.2	Analyse de séquences d'images par transformées en ondelettes	59
2.1.2.1	Décomposition en ondelettes spatiales	60
2.1.2.2	Décomposition en ondelettes temporelles	61
2.1.2.3	Décomposition en ondelettes 2D+T	61
2.1.2.4	Décomposition en ondelettes par la méthode de J. R. Smith <i>et al.</i>	62
2.1.3	Ondelettes usuelles versus ondelettes géométriques	64
2.2	Transformée en curvelets	65
2.2.1	Curvelets 2D continues	65
2.2.2	Curvelets 2D discrètes	67
2.2.2.1	Description	67
2.2.2.2	Implémentations	69
2.2.2.3	Expérimentations	71
2.2.3	Transformée en curvelets tri-dimensionnelle	74
2.2.3.1	Description	75
2.2.3.2	Passage au 2D+T	77
2.2.3.3	Application aux séquences d'images	78

Le principal objectif de cette thèse porte sur l'analyse et la caractérisation de textures dynamiques à partir d'un ensemble de mesures ou d'observations.

Le chapitre précédent a montré que les textures dynamiques présentaient des répétitivités à la fois spatiales et temporelles apparaissant à différentes échelles. Comme nous essaierons de le montrer au travers de ce chapitre, les transformées multi-échelles représentent une stratégie pertinente et naturelle pour l'extraction d'information et l'analyse des textures dynamiques.

Ce chapitre, après une courte introduction sur l'analyse multi-résolutions dans le cadre mono-dimensionnel, étudie différentes transformées en ondelettes, et présente leurs avantages et inconvénients pour l'indexation. L'objectif est de relever les points critiques de celles-ci afin de souligner la nécessité de rechercher des transformées guidées par la géométrie du signal 2D ou 3D. Nous présentons alors les différentes familles de transformées multi-échelles géométriques de la littérature.

Nous nous focalisons ensuite sur les propriétés de la transformée en curvelets, étudiées dans le cadre bi puis tri-dimensionnel. Puis nous développons plus longuement son adaptation à l'espace 2D+T nécessaire à l'analyse des séquences d'images. Nous mettons plus particulièrement l'accent sur le développement d'outils d'aide à l'interprétation des informations extraites.

2.1 Décompositions multi-échelles usuelles

La transformée en ondelettes a été introduite initialement pour des données sismiques puis dans le domaine du traitement des données multimédia (sons, images, vidéos, ...) [Mallat 89, Mallat 98, Meyer 90] afin de combler la lacune entre deux représentations d'un signal $f \in \mathcal{L}^2(\mathbb{R})$

– dans la base continue des distributions de Dirac :

$$f(t) = \int_{\mathbb{R}} f(u)\delta(t-u)du \quad (2.1)$$

– dans la base de Fourier :

$$f(t) = \int_{\mathbb{R}} \widehat{f}(\omega)e^{i\omega t}d\omega \quad (2.2)$$

La représentation (2.1) fournit une information précise en temps : la valeur de $f(t)$ représente l'intensité du signal f au point t . Par contre, elle ne met aucunement en évidence l'information fréquentielle. Réciproquement, la représentation (2.2) est purement fréquentielle. Évidemment, chacune de ces représentations contient les informations de l'autre.

On peut illustrer ces deux points de vue différents dans le plan temps-fréquence où le temps varie horizontalement et la fréquence verticalement. Pour une représentation donnée, ce plan est pavé par des rectangles, appelé boîtes d'Heisenberg [Benedetto 00] dont les dimensions σ_t^2 et σ_ω^2 et les centres (μ_t, μ_ω) dépendent de la base choisie. Si ψ est un élément de cette base, les dimensions d'une boîte d'Heisenberg est :

$$\begin{aligned} \sigma_t^2 &= \frac{1}{\|f\|^2} \int_{-\infty}^{+\infty} (t - \mu_t)^2 |\psi(t)|^2 dt && \text{(dispersion temporelle)} \\ \sigma_\omega^2 &= \frac{1}{2\pi\|f\|^2} \int_{-\infty}^{+\infty} (\omega - \mu_\omega)^2 |\widehat{\psi}(\omega)|^2 d\omega && \text{(dispersion fréquentielle)} \end{aligned} \quad (2.3)$$

Le principe d'incertitude d'Heisenberg stipule que, quelle que soit la représentation, l'aire de chaque boîte est au moins égale à $1/4$:

$$\sigma_t^2 \sigma_\omega^2 \geq \frac{1}{4} \quad (2.4)$$

mais suivant la transformation, les boîtes auront des tailles et positions différentes. Pour les deux représentations (2.1) et (2.2), le pavage temps-fréquence est illustré par la figure 2.1.

Concernant la représentation dans la base des distributions de Dirac, l'analyse est précise sur l'axe temporel, mais l'incertitude sur l'axe fréquentiel est grande. Réciproquement, dans le cas de la représentation dans la base de Fourier, la précision fréquentielle est maximum tandis que le signal n'est pas localisé spatialement.

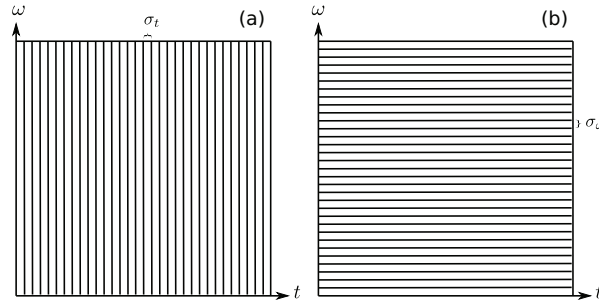


FIG. 2.1 – Pavages temps-fréquence sur la base des distributions de Dirac (a) et sur la base de Fourier (b).

Lorsqu'on utilise l'analyse de Fourier par fenêtre glissante [Gabor 46], les dimensions des boites d'Heisenberg dépendent de la taille de la fenêtre d'analyse. La figure 2.2 illustre deux exemples pavages temps-fréquence en fonction de la taille de la fenêtre.

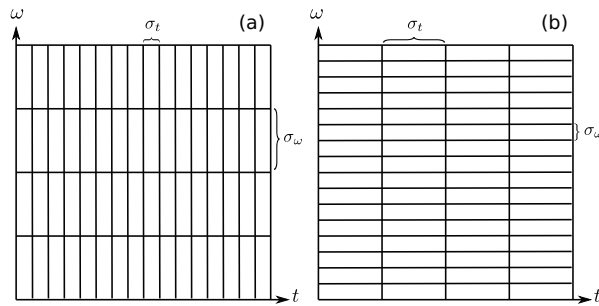


FIG. 2.2 – Pavages temps-fréquence de deux transformées de Fourier à fenêtre glissante (fenêtre étroite (a) et fenêtre plus large (b)) illustrant le principe d'incertitude d'Heisenberg.

Avec une fenêtre d'analyse de taille réduite (cf. figure 2.2.(a)), la précision temporelle reste importante, mais possède une incertitude forte du point de vue fréquentiel. Pour déterminer de manière plus précise la fréquence d'un signal, il faut alors l'analyser sur une durée plus longue (cf. figure 2.2.(b)). La théorie des ondelettes a été introduite pour effectuer un pavage temps-fréquence adaptatif, utilisant pour l'analyse des basses fréquences des grandes fenêtres et pour l'analyse des hautes fréquences des fenêtres avec un support plus petit. Alors que la base de décomposition de Fourier est constituée d'exponentielles complexes, localisée en fréquence mais non en temps. Une ondelette ψ est une fonction oscillante de moyenne nulle :

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (2.5)$$

et on obtient une version dilatée par un paramètre d'échelle s , et translatée de u :

$$\psi_{s,u}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) dt \quad (2.6)$$

permettant d'analyser différentes fréquences à différentes échelles.

Pour les ondelettes, la localisation en temps est d'autant plus précise que la localisation en fréquence est grande. On obtient ainsi le pavage de la figure 2.3, dont la résolution temps-fréquence dépend de l'échelle.

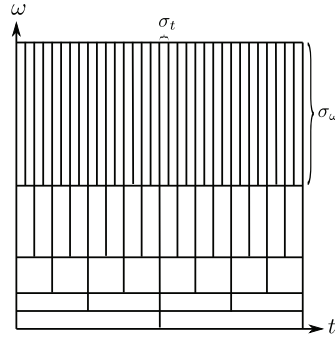


FIG. 2.3 – Pavage temps-fréquence obtenu lors d'une transformée en ondelettes. La résolution temps-fréquence dépend de l'échelle d'analyse.

2.1.1 Analyse multi-résolutions

Dans un soucis didactique, nous présentons les définitions importantes de l'analyse multi-résolutions dans le cas mono-dimensionnel en insistant sur l'implémentation pratique. Puis, nous présentons son extension dans le cadre de l'analyse de séquences d'images.

2.1.1.1 Bases d'ondelettes et analyse multi-résolutions

Soit la fonction $f \in \mathcal{L}^2(\mathbb{R})$, une suite de sous-espaces vectoriels fermés $\{\mathbf{V}_j\}_{j \in \mathbb{Z}}$ de $\mathcal{L}^2(\mathbb{R})$ est une analyse multi-résolutions de f si :

- une approximation au niveau de résolution j contient toute l'information nécessaire au calcul d'une approximation à un niveau de résolution $j + 1$.

$$\forall j \in \mathbb{Z}, \quad \mathbf{V}_{j+1} \subset \mathbf{V}_j \quad (2.7)$$

Comme le sous-espace vectoriel \mathbf{V}_{j+1} est inclus dans le sous espace vectoriel \mathbf{V}_j , l'approximation au niveau de résolution $j + 1$ contiendra moins de détails que celle au niveau de résolution j .

- \mathbf{V}_j est invariant à toute translation de longueur proportionnelle à l'échelle 2^j .

$$\forall (j, k) \in \mathbb{Z}^2, \quad f(t) \in \mathbf{V}_j \Leftrightarrow f(t - 2^j k) \in \mathbf{V}_j \quad (2.8)$$

- les fonctions de \mathbf{V}_j sont obtenues par dilatation d'un facteur 2 (analyse dyadique) des fonctions de \mathbf{V}_{j+1} .

$$\forall j \in \mathbb{Z}, f(t) \in \mathbf{V}_j \Leftrightarrow f\left(\frac{t}{2}\right) \in \mathbf{V}_{j+1} \quad (2.9)$$

-

$$\lim_{j \rightarrow +\infty} \mathbf{V}_j = \bigcap_{j=-\infty}^{+\infty} \mathbf{V}_j = \{0\} \quad (2.10)$$

Quand l'échelle 2^{-j} tend vers 0 lorsque $j \rightarrow +\infty$, (2.10) implique que l'on perd tous les détails de la fonction f .

•

$$\lim_{j \rightarrow -\infty} \mathbf{V}_j = \overline{\bigcup_{j=-\infty}^{+\infty} \mathbf{V}_j} = \mathcal{L}^2(\mathbb{R}) \quad (2.11)$$

avec \overline{V} l'adhérence de l'ensemble V .

Quand l'échelle 2^{-j} tend vers $+\infty$ lorsque $j \rightarrow -\infty$, (2.11) entraîne la convergence de l'approximation vers le signal d'origine f .

On suppose l'existence d'une fonction $\phi \in \mathcal{L}^2(\mathbb{R})$, appelée fonction d'échelle, telle que ses translatées $\{\phi(t-k)\}_{k \in \mathbb{Z}}$ forment une base orthonormale de \mathbf{V}_0 . L'utilisation des relations (2.9) et (2.8) permet de prouver que les fonctions $\{\phi_k^j\}_{k \in \mathbb{Z}}$ définies par :

$$\forall k \in \mathbb{Z}, \quad \phi_k^j(t) = \frac{1}{\sqrt{2^j}} \phi\left(\frac{t}{2^j} - k\right) \quad (2.12)$$

forment une base orthonormale de \mathbf{V}_j [Mallat 89, Mallat 98].

La relation d'emboîtement (2.7) implique que les projections de f sur \mathbf{V}_j sont de plus en plus grossières, au fur et à mesure que j croît. La différence entre deux approximations successives représente donc l'information perdue. On peut définir un sous-espace vectoriel \mathbf{W}_j représentant l'information perdue lors du passage à la résolution supérieure :

$$\forall j \in \mathbb{Z}, \quad \mathbf{V}_j = \mathbf{V}_{j+1} \oplus \mathbf{W}_{j+1} \quad (2.13)$$

On peut alors montrer l'existence d'une fonction ψ appelée ondelette mère telle que ses translatées $\{\psi(t-k)\}_{k \in \mathbb{Z}}$ forment une base orthonormale de \mathbf{W}_0 . Comme précédemment, l'utilisation de la relation (2.8) et (2.9) permet de montrer que les fonctions $\{\psi_k^j\}_{j,k \in \mathbb{Z}}$ définies par :

$$\forall (j, k) \in \mathbb{Z}^2, \quad \psi_k^j(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t}{2^j} - k\right) \quad (2.14)$$

forment une base orthonormale de \mathbf{W}_j [Mallat 89, Mallat 98].

f étant une fonction \mathcal{C}^0 , on peut fixer j de manière à ce que $f \in \mathbf{V}_0$. La transformée en ondelettes de la fonction f sur n niveaux est alors définie comme la projection de cette fonction sur les sous-espaces vectoriels \mathbf{V}_n et $\{\mathbf{W}_j\}_{1 \leq j \leq n}$ puisque :

$$\mathbf{V}_0 = \mathbf{V}_n \oplus \left[\bigoplus_{j=1}^n \mathbf{W}_j \right] \quad (2.15)$$

que l'on peut exprimer à l'aide des relations (2.12) et (2.14) comme :

$$f(t) = \sum_k a^0[k] \phi_k(t) + \sum_{j=0}^n \sum_k d^j[k] \psi_k^j(t) \quad (2.16)$$

où $a^j[k] = \langle f, \phi_k^j \rangle$ sont les coefficients d'approximation à l'échelle j et $d^j[k] = \langle f, \psi_k^j \rangle$ sont les coefficients d'ondelettes ou coefficients de détails à l'échelle j . $\langle \cdot, \cdot \rangle$ est le produit scalaire usuel de $\mathcal{L}^2(\mathbb{R})$.

Calculer explicitement les coefficients $a^j[k]$ et $d^j[k]$ de la transformée en ondelettes est possible. Cependant, l'intégration sur \mathbb{R} rend ces calculs très lourds en temps. L'approche par banc de filtres permet de lever ce verrou [Mallat 89] et de calculer de manière itérative ces coefficients.

2.1.1.2 Bancs de filtres

La propriété (2.7) impose que $\mathbf{V}_{j+1} \subset \mathbf{V}_j$, ce qui conduit à : $\frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right) \in \mathbf{V}_1 \subset \mathbf{V}_0$. De plus, comme $\{\phi(t-k)\}_{k \in \mathbb{Z}}$ est une base orthonormale de \mathbf{V}_0 , on peut faire la décomposition :

$$\frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right) = \sum_{k=-\infty}^{+\infty} h[k]\phi(t-k) \quad \text{avec} \quad h[k] = \left\langle \frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right), \phi(t-k) \right\rangle \quad (2.17)$$

Cette relation, appelée équation d'échelle, exprime la dilatation par 2 de ϕ en fonction de ses translatées. On peut facilement interpréter la suite $\{h[k]\}_{k \in \mathbb{Z}}$ comme un filtre discret. De la même manière, nous avons la décomposition :

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{k=-\infty}^{+\infty} g[k]\phi(t-k) \quad \text{avec} \quad g[k] = \left\langle \frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right), \phi(t-k) \right\rangle \quad (2.18)$$

Les relations 2.17 et 2.18 sont aussi appelées équations à deux échelles. La condition d'orthogonalité [Mallat 98] permet d'assurer l'existence de ψ et lie les deux suites $\{h[k]\}_{k \in \mathbb{Z}}$ et $\{g[k]\}_{k \in \mathbb{Z}}$ par :

$$g[k] = (-1)^{1-k}h[1-k] \quad (2.19)$$

Le banc de filtres se construit alors à l'aide des relations (2.17) et (2.18) :

$$a^{j+1}[k] = a^j \otimes \bar{h}[2k] \quad (2.20)$$

$$d^{j+1}[k] = a^j \otimes \bar{g}[2k] \quad (2.21)$$

où \otimes est le produit de convolution discret circulaire, et $\bar{g}[m] = g[-m]$.

Ce banc de filtres se représente classiquement comme sur la figure 2.4. $\boxed{\text{F}}$ représente la convolution du signal par le filtre F et $\downarrow 2$ la décimation d'un facteur 2 du signal.

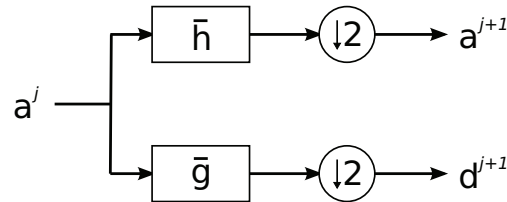
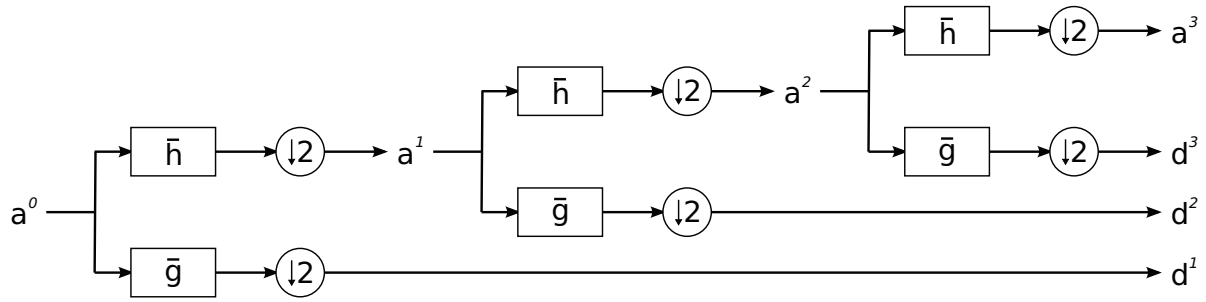


FIG. 2.4 – Schéma du banc de filtres de la transformée en ondelettes permettant de passer du niveau de résolution j à $j+1$.

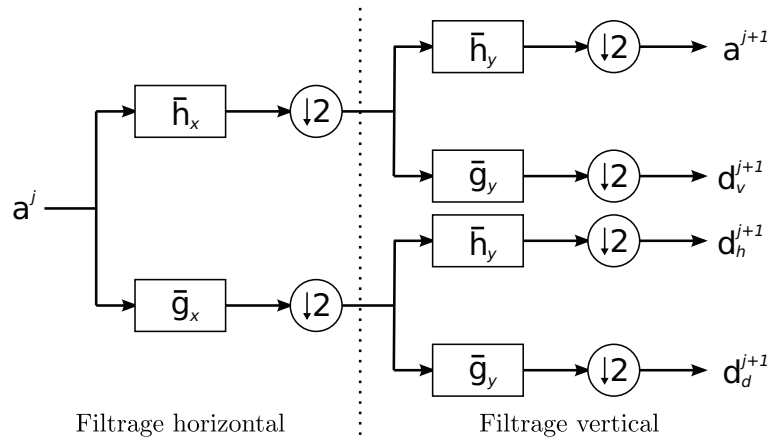
Ce banc de filtres implémente la transformée en ondelettes rapide d'un signal. Il suffit d'initialiser a^0 avec les valeurs des échantillons du signal d'origine et de mettre en cascade le banc de filtres jusqu'au niveau de résolution j_{max} maximum comme le montre la figure 2.5.

Après avoir présenté l'implémentation pratique de la transformée en ondelettes rapide pour un signal mono-dimensionnel, les extensions classiques aux séquences d'images sont exposées. L'apport de ces premières transformées tridimensionnelles sur les textures dynamiques sera ensuite analysé.

FIG. 2.5 – Schéma du banc de filtres de la transformée en ondelettes rapide pour $j_{max} = 3$.

2.1.2 Analyse de séquences d'images par transformées en ondelettes

La transformée en ondelettes de signaux multi-dimensionnels peut être réalisée de façon séparable par transformations successives sur chacune des dimensions [Mallat 98]. Ainsi la décomposition en ondelettes d'une image s'effectuera sur les lignes puis sur les colonnes comme illustré par la figure 2.6.

FIG. 2.6 – Banc de filtres de la décomposition en ondelettes d'une image. Avec a^{j+1} l'approximation de niveau de résolution $j + 1$, d_v^{j+1} les détails verticaux, d_h^{j+1} les détails horizontaux et d_d^{j+1} les détails diagonaux.

Par la suite, le banc de filtres sera représenté dans les figures par $\boxed{\text{WT}}$ et sera défini sous la forme suivante :

$$\begin{aligned}
 a^{j+1}[n, m] &= a^j \otimes \overline{h_x} \overline{h_y}[2n, 2m] \\
 d_v^{j+1}[n, m] &= a^j \otimes \overline{h_x} \overline{g_y}[2n, 2m] \\
 d_h^{j+1}[n, m] &= a^j \otimes \overline{g_x} \overline{h_y}[2n, 2m] \\
 d_d^{j+1}[n, m] &= a^j \otimes \overline{g_x} \overline{g_y}[2n, 2m]
 \end{aligned} \tag{2.22}$$

avec $hg[n_1, n_2] = h[n_1]g[n_2]$.

Ainsi, pour une image, nous obtenons une sous-bande d'approximation et trois sous-bandes de détails, avec des informations différentes. Ainsi, dans $d_v^{j+1}[n, m]$, on ne retrouvera que les détails verticaux de l'image à l'échelle j considérée.

Dans la suite de cette section, sont présentées quatre transformées en ondelettes pour l'analyse de séquences d'images qui sont appliquées à l'étude des textures dynamiques. Ces méthodes sont une extension presque directe du cas 1D, mais diffèrent par la manière dont sont considérées les variables spatiales et temporelles dans l'analyse multi-résolutions. Une de ces méthodes est issue de la littérature, et est à l'heure actuelle, la référence en terme de classification de textures dynamiques par analyse en ondelettes [Smith 02]. Les trois autres sont des extensions de cette approche et représentent une contribution à l'utilisation des ondelettes pour l'analyse des textures dynamiques.

2.1.2.1 Décomposition en ondelettes spatiales

La première approche consiste à utiliser une transformée en ondelettes 2D directement sur la séquence d'images, en l'appliquant image par image. Dans ce cas, l'information temporelle n'est pas prise en compte, et seule l'information spatiale sera présente dans les sous-bandes de détails.

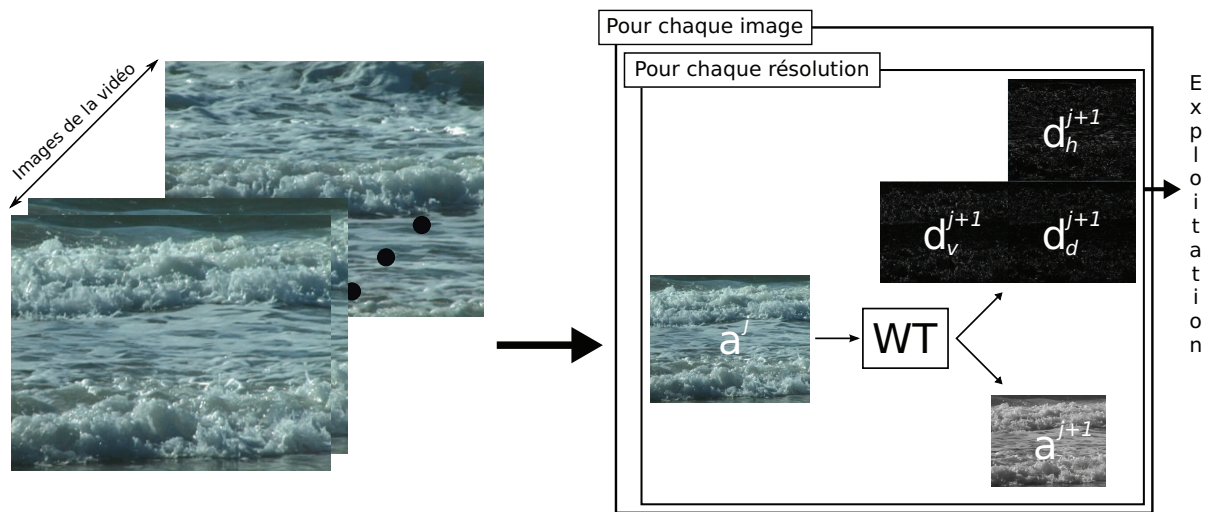


FIG. 2.7 – Transformée en ondelettes 2D appliquée sur une séquence vidéo image par image.

Cette première méthode, illustrée par la figure 2.7, calcule pour chaque image de la vidéo et pour chaque résolution de la décomposition en ondelettes, la sous-bande d'approximation et les trois sous-bandes de détails. Ces différentes sous-bandes permettront par la suite la construction d'un vecteur caractéristique (cf section 5.1).

Dans un objectif de caractérisation, cette transformée privilégiera les informations spatiales.

2.1.2.2 Décomposition en ondelettes temporelles

La première méthode mentionnée repose sur le principe d'une analyse multi-échelles image par image et est donc purement spatiale. La seconde approche est également une approche naïve dans le sens, où seule l'information temporelle est prise en compte. Autrement dit l'analyse multi-résolutions s'effectue uniquement dans la direction temporelle.

Cette méthode, illustrée par la figure 2.8, extrait pour chaque pixel (x, y) de la séquence d'images le profil temporel, puis applique le banc de filtres suivant :

$$\begin{aligned} a^{j+1}[k] &= a^j \otimes \overline{h_t}[2k] \\ d_t^{j+1}[k] &= a^j \otimes \overline{g_t}[2k] \end{aligned} \quad (2.23)$$

L'information obtenue dans les sous-bandes de détails est alors utilisée afin de construire un vecteur caractéristique (cf. section 5.1).

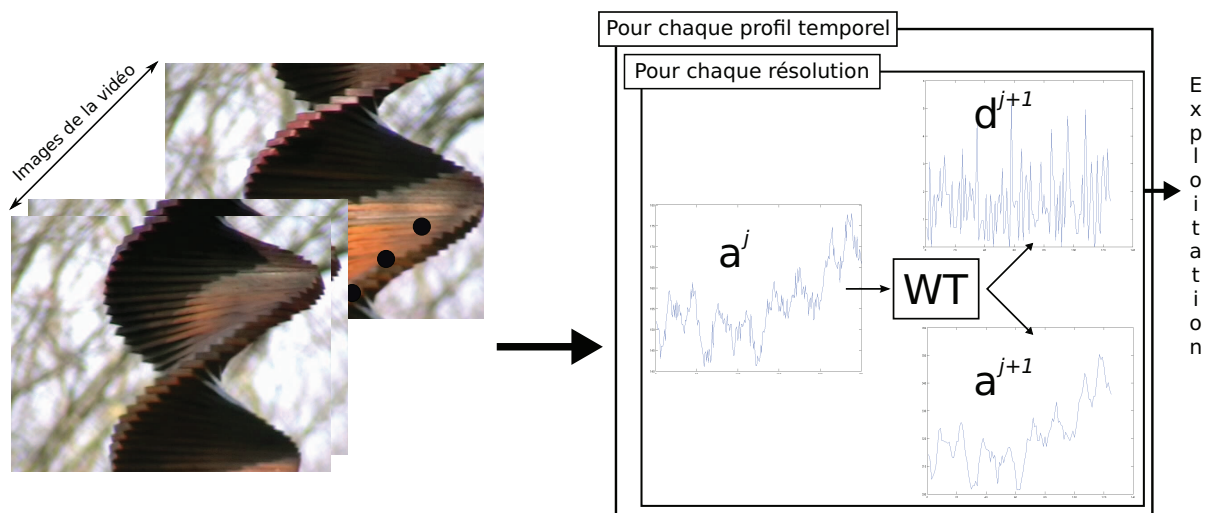


FIG. 2.8 – Transformée en ondelettes 1D appliquée sur les profils temporels d'une séquence d'images.

Contrairement à la transformée précédente, celle-ci va permettre une caractérisation uniquement temporelle de l'information contenue dans les séquences d'images.

2.1.2.3 Décomposition en ondelettes 2D+T

Les deux premières méthodes, que nous avons abordées, analysent uniquement un type d'information, l'information spatiale ou l'information temporelle, et pour cette raison nous les qualifions de naïves. La troisième méthode, quant à elle, entremêle les deux informations afin d'obtenir des sous-bandes regroupant l'information spatiale et temporelle.

Cette extension au domaine temporel de la transformée en ondelettes s'apparente au passage du 1D au 2D. Un filtre de convolution, à trois dimensions, séparable en trois noyaux (lignes, colonnes, profils temporels) est utilisé. Ainsi pour chaque niveau de la décomposition, trois filtres en cascade sont utilisés pour l'obtention de la résolution supérieure.

Ce qui se formalise de la manière suivante :

$$\begin{aligned}
 a^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{h_y} \overline{h_t}[2n, 2m, 2p] \\
 d_t^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{h_y} \overline{g_t}[2n, 2m, 2p] \\
 d_h^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{h_y} \overline{h_t}[2n, 2m, 2p] \\
 d_{ht}^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{h_y} \overline{g_t}[2n, 2m, 2p] \\
 d_v^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{g_y} \overline{h_t}[2n, 2m, 2p] \\
 d_{vt}^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{g_y} \overline{g_t}[2n, 2m, 2p] \\
 d_d^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{g_y} \overline{h_t}[2n, 2m, 2p] \\
 d_{dt}^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{g_y} \overline{g_t}[2n, 2m, 2p]
 \end{aligned} \tag{2.24}$$

La figure 2.9 illustre le fonctionnement de la troisième méthode. Nous pouvons observer que pour une séquence d'images, nous obtenons une sous-bande d'approximation et sept sous-bandes de détails. Comme pour les deux autres méthodes, l'information obtenue dans les différentes sous-bandes sera utilisée dans la construction de vecteurs caractéristiques (cf. section 5.1).

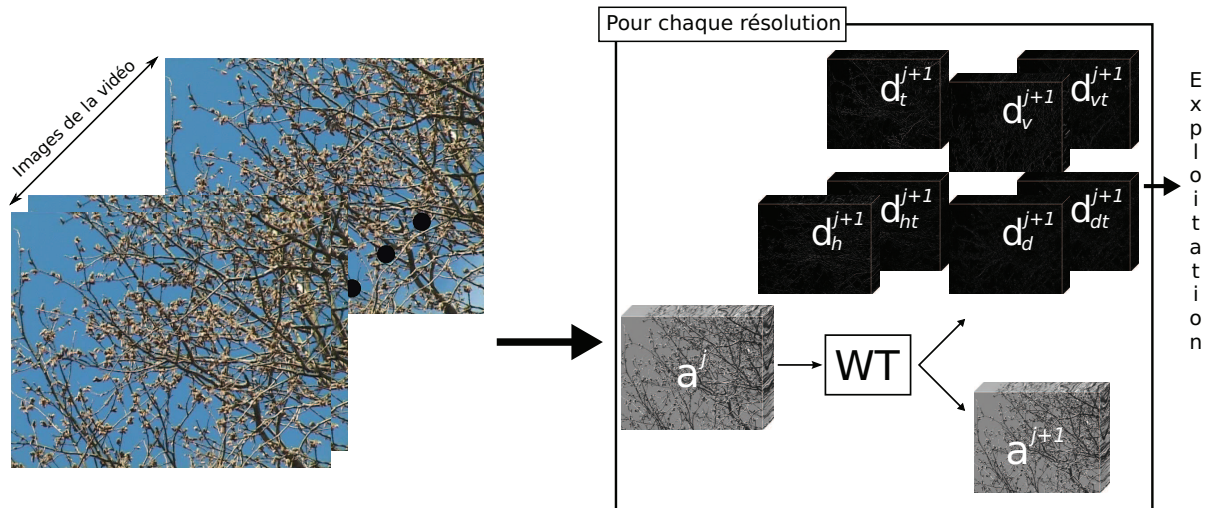


FIG. 2.9 – Décomposition en ondelettes 2D+T appliquée sur une séquence d'images.

Dans un objectif d'indexation, cette transformée permet, contrairement aux deux précédentes, de prendre en compte avec les mêmes pondérations, l'information spatiale et temporelle.

2.1.2.4 Décomposition en ondelettes par la méthode de J. R. Smith *et al.*

La dernière méthode de décomposition est celle de J. R. Smith *et al.* [Smith 02]. Cette transformée est similaire à la décomposition en ondelettes 2D+T présentée précédemment, sauf que le filtre temporel est appliqué une deuxième fois à chaque étape de l'analyse multi-résolutions. La vidéo est ainsi décimée autant de fois spatialement que temporellement.

Ainsi le passage d'une résolution j à $j + 1$, illustré par la figure 2.10, génère une sous-bande

d'approximation et quinze sous-bandes de détails. Celles-ci sont obtenues à l'aide du banc de filtres suivant :

$$\begin{aligned}
 a^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{h_y} \overline{h_t} \overline{h_t}[2n, 2m, 4p] \\
 d_t^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{h_y} \overline{h_t} \overline{g_t}[2n, 2m, 4p] \\
 d_h^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{h_y} \overline{h_t} \overline{h_t}[2n, 2m, 4p] \\
 d_{ht}^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{h_y} \overline{g_t} \overline{h_t}[2n, 2m, 4p] \\
 d_v^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{g_y} \overline{h_t} \overline{h_t}[2n, 2m, 4p] \\
 d_{vt}^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{g_y} \overline{g_t} \overline{h_t}[2n, 2m, 4p] \\
 d_d^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{g_y} \overline{h_t} \overline{h_t}[2n, 2m, 4p] \\
 d_{dt}^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{g_y} \overline{h_t} \overline{g_t}[2n, 2m, 4p] \\
 d_t^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{h_y} \overline{g_t} \overline{h_t}[2n, 2m, 4p] \\
 d_{tt}^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{h_y} \overline{g_t} \overline{g_t}[2n, 2m, 4p] \\
 d_{ht}^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{h_y} \overline{g_t} \overline{h_t}[2n, 2m, 4p] \\
 d_{htt}^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{h_y} \overline{g_t} \overline{g_t}[2n, 2m, 4p] \\
 d_{vt}^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{g_y} \overline{g_t} \overline{h_t}[2n, 2m, 4p] \\
 d_{vtt}^{j+1}[n, m, p] &= a^j \otimes \overline{h_x} \overline{g_y} \overline{g_t} \overline{g_t}[2n, 2m, 4p] \\
 d_{dt}^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{g_y} \overline{g_t} \overline{h_t}[2n, 2m, 4p] \\
 d_{dtt}^{j+1}[n, m, p] &= a^j \otimes \overline{g_x} \overline{g_y} \overline{g_t} \overline{g_t}[2n, 2m, 4p]
 \end{aligned} \tag{2.25}$$

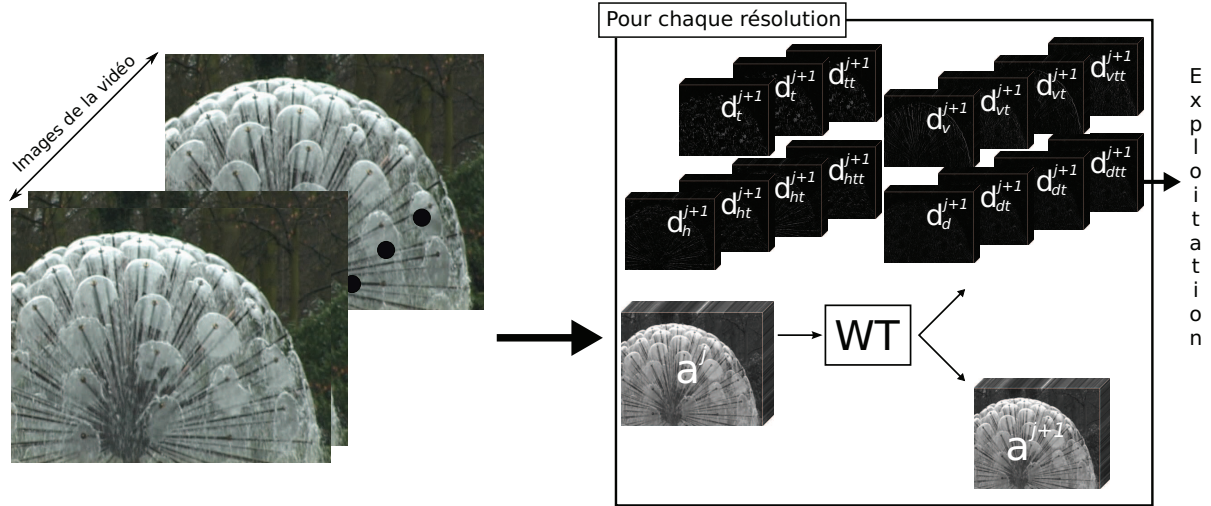


FIG. 2.10 – Transformée en ondelettes de J. R. Smith et al. [Smith 02] appliquée sur une séquence d'images.

L'ensemble des sous-bandes obtenue à l'aide de cette méthode sera, par la suite, utilisé dans la construction d'un vecteur de signatures (cf. Chapitre 5.1).

Les informations caractérisées par cette transformée sont à la fois spatiale et temporelle. Toutefois, une importance plus grande est donnée au temps.

2.1.3 Ondelettes usuelles versus ondelettes géométriques

Il est indéniable que la transformée en ondelettes a eu un impact majeur dans de nombreuses applications du traitement du signal et de l'image. Pourtant celle-ci échoue dans la représentation/détection d'objets composés d'éléments anisotropiques, comme des lignes ou des courbes.

Pour cette raison, ces dernières années ont vu la naissance de plusieurs transformées multi-échelles géométriques. Citons :

- la transformée en ridgelets [Candès 98, Donoho 98] ; elle compte parmi les premières transformées en ondelettes géométriques. Sa construction s'appuie sur la transformée de Radon. L'analyse par ridgelets revient à projeter le signal à l'aide de la transformée de Radon, puis à calculer une transformée en ondelettes sur chaque projection. Cette transformée géométrique permet de prendre en compte les ruptures linéaires d'une image.
- la transformée en curvelets [Donoho 99b, Candès 00b, Starck 00, Candès 04, Candès 05b] ; elle prend en compte les singularités de co-dimensions 1, autrement dit, des courbes dans le cadre d'une image réelle, des surfaces dans le cadre de volume 3D. Il existe deux générations de transformée en curvelets. La deuxième génération n'utilise pas la transformée en ridgelets.
- la transformée en contourlet [Do 05] ; elle est une variante de la transformée en curvelets dite de première génération, en utilisant à la place de la transformée de ridgelets, un banc de filtres directionnels proposé par Bamberger *et al.* [Bamberger 92].
- La transformée en beamlets [Donoho 01] ; utilisée sous forme pyramidale et hiérarchique, l'analyse par beamlets permet d'obtenir des résultats pertinents dans le cas d'une analyse multi-résolutions d'images contenant des filaments ou des courbes dans un contexte bruité.
- la transformée en wedgelet [Donoho 99a] ; la décomposition multi-échelles en wedgelets divise une image en bloc dyadiques à différentes échelles. Les wedgelets sont des fonctions constantes par morceaux avec des discontinuités linéaires, elles sont donc adaptées aux images constituées de régions constantes avec des contours réguliers. Une extension 3D de cette transformée existe [Chandrasekaran 04].
- la transformée en platelets [Willett 03] ; elle généralise la transformée en wedgelets et produit des approximations linéaires par morceaux d'une image. Les platelets sont donc très bien adaptées pour des images qui sont composées de régions linéaires séparées par des contours réguliers.

Les curvelets sont fondées sur des fonctions analysantes indexées par des paramètres d'échelle, de position et de direction ce qui permet une analyse fine d'un signal 2D ou 3D. Elles permettent une représentation plus creuse que les ondelettes pour les singularités de co-dimensions 1 de type contour, ce qui correspond pour les séquences d'images à des fronts d'ondes, motifs spatio-temporels souvent rencontrés pour les textures dynamiques comme nous le verrons plus loin.

De plus, l'implémentation actuelle de cette transformée est facilitée par l'apparition de nouveaux algorithmes rapides. Ainsi, celles-ci nous ont paru très pertinentes pour la compréhension des textures dynamiques.

2.2 Transformée en curvelets

Dans la littérature, deux générations de transformées en curvelets existent. Dans cette étude, nous utilisons la transformée en curvelets dite de deuxième génération [Candès 04, Candès 05b].

Celle-ci présente en effet des avantages par rapport à celle dite de première génération [Donoho 99b, Candès 00b, Starck 00] :

- une structure de coefficients plus simple.
- la relation d'échelle parabolique pour une sous-bande de détails respecte la relation

$$\text{hauteur}^2 \approx \text{largeur}.$$

Dans le domaine spatial, une curvelet à l'échelle j est un atome orienté qui a pour support un rectangle de taille $\text{largeur} \times \text{hauteur}$.

- le partitionnement spatial n'engendre plus d'effets de blocs.

Cette section commence par rappeler le cadre de la transformée en curvelets, en 2D dans un premier temps, puis en 3D. Seront abordées la construction des atomes de curvelets, leurs propriétés ainsi que leur implémentation.

La transformée en curvelets 3D étant définie pour des données tri-dimensionnelles, une approche est proposée pour l'appliquer au contexte des séquences d'images. Une transformée en curvelets 2D+T sera présentée dans ce but.

La dernière partie de cette section sera consacrée à la définition d'outils d'aide à l'analyse dans le domaine de cette transformée. Ceux-ci sont rendus nécessaires par le fait que l'espace des coefficients est à 5 dimensions et l'interprétation dans cet espace est donc difficile. Elle est pourtant indispensable pour une meilleure compréhension des résultats de la transformée en curvelets sur les séquences d'images.

2.2.1 Curvelets 2D continus

Comme pour la transformée en ondelettes, la transformée en curvelets 2D consiste en la projection d'une fonction sur une base de fonctions $L^2(\mathbb{R}^2)$. Ainsi, on obtient une collection de coefficients $c(j, \ell, \mathbf{k})$ définie par le produit scalaire de $L^2(\mathbb{R}^2)$ entre une fonction $f \in L^2(\mathbb{R}^2)$ et les fonctions de curvelets d'analyse $\varphi_{j,\ell,\mathbf{k}}$, appelées aussi atomes :

$$c(j, \ell, \mathbf{k}) := \langle f, \varphi_{j,\ell,\mathbf{k}} \rangle = \int_{\mathbb{R}^2} f(\mathbf{x}) \overline{\varphi_{j,\ell,\mathbf{k}}(\mathbf{x})} d\mathbf{x} \quad (2.26)$$

où $\varphi_{j,\ell,\mathbf{k}}$ est l'atome de curvelet à l'échelle 2^{-j} , à l'orientation θ_ℓ et à la position $\mathbf{k} = (k_1, k_2)$. Ces atomes sont construits par composition d'une translation $\mathbf{x}_{\mathbf{k}}^{(j,\ell)}$ et d'une rotation R_{θ_ℓ} d'angle θ_ℓ (en radian) d'un atome φ_j :

$$\varphi_{j,\ell,\mathbf{k}} = \varphi_j \left(R_{\theta_\ell} \left(\mathbf{x} - \mathbf{x}_{\mathbf{k}}^{(j,\ell)} \right) \right) \quad (2.27)$$

La rotation θ_ℓ se définit par :

$$R_{\theta_\ell} = \begin{pmatrix} \cos \theta_\ell & \sin \theta_\ell \\ -\sin \theta_\ell & \cos \theta_\ell \end{pmatrix} \quad \text{et} \quad R_{\theta_\ell}^{-1} = R_{-\theta_\ell} \quad (2.28)$$

avec θ_ℓ une séquence d'angles équi-répartis tels que $\theta_\ell = 2\pi 2^{-\lfloor j/2 \rfloor} \ell$, avec $\ell = 0, 1, \dots$

L'atome de curvelet mère φ_j s'exprime dans le domaine fréquentiel par la moyenne de ses

transformées de Fourier, $\widehat{\varphi}_j(\boldsymbol{\omega}) = U_j(\boldsymbol{\omega})$, que l'on écrit en coordonnées polaires de la manière suivante :

$$U_j(r, \theta) = 2^{-3j/4} W_j(2^{-j}r) V_j\left(\frac{2^{\lfloor j/2 \rfloor} \theta}{2\pi}\right) \quad (2.29)$$

Le support de $U_j \in \mathbb{C}$ est un secteur polaire défini par les supports de $W_j \in \mathbb{C}$ et $V_j \in \mathbb{C}$, représentant respectivement une fenêtre radiale et une fenêtre angulaire. Ces fenêtres doivent obéir à des conditions d'admissibilités :

$$\sum_{j=-\infty}^{+\infty} W_j^2(2^{-j}r) = 1 \quad \text{pour} \quad \forall r \in [3/4, 3/2] \quad (2.30)$$

$$\sum_{\ell=-\infty}^{+\infty} V_j^2(t - \ell) = 1 \quad \text{pour} \quad \forall t \in [-1/2, 1/2] \quad (2.31)$$

et doivent être régulières, non-négatives et à valeurs réelles. Le support de W_j , respectivement de V_j , est $r \in [1/2, 2]$, respectivement $t \in [-1, 1]$.

La fenêtre W_0 correspond à une approximation du signal, celle-ci doit obéir à la condition suivante :

$$W_0^2(r) + \sum_{j>0} W_j^2(2^{-j}r) = 1 \quad (2.32)$$

La curvelet à l'échelle la plus grossière (approximation) est non directionnelle et peut se définir comme :

$$\varphi_{0,\mathbf{k}}(\mathbf{x}) = \varphi_0(\mathbf{x} - 2^{-j_0}\mathbf{k}) \quad (2.33)$$

La transformée en curvelets complète consiste donc en des éléments permettant d'obtenir des détails d'un signal selon des paramètres d'échelle, de position et de direction, $\{\varphi_{j,\ell,\mathbf{k}}\}_{j>0,\ell,\mathbf{k}}$, et un élément donnant une approximation grossière de la fonction f , $\{\varphi_{0,\mathbf{k}}\}_{\mathbf{k}}$. Le découpage du domaine fréquentiel de cette transformée est illustré par la figure 2.11.

La transformée en curvelets possède différentes propriétés intéressantes :

- **Conservation de l'énergie** : nous pouvons exprimer facilement une fonction $f \in L^2(\mathbb{R}^2)$ comme une série de curvelets, et on a la formule de reconstruction :

$$f = \sum_{j,\ell,\mathbf{k}} \langle f, \varphi_{j,\ell,\mathbf{k}} \rangle \varphi_{j,\ell,\mathbf{k}} \quad (2.34)$$

De plus, la loi de conservation de l'énergie est vérifiée :

$$\sum_{j,\ell,\mathbf{k}} |\langle f, \varphi_{j,\ell,\mathbf{k}} \rangle|^2 = \|f\|^2 \quad (2.35)$$

- **Loi d'échelle parabolique** : la localisation fréquentielle de $U_{j,\ell}$ implique, comme le montre la figure 2.11, un découpage par secteurs que l'on peut approximer par un rectangle de largeur 2^{-j} et hauteur $2^{-j/2}$. Ainsi pour chaque échelle, nous avons la relation suivante :

$$\text{hauteur} \approx 2^{-j/2}, \quad \text{largeur} \approx 2^{-j} \quad \Rightarrow \quad \text{largeur} \approx \text{hauteur}^2 \quad (2.36)$$

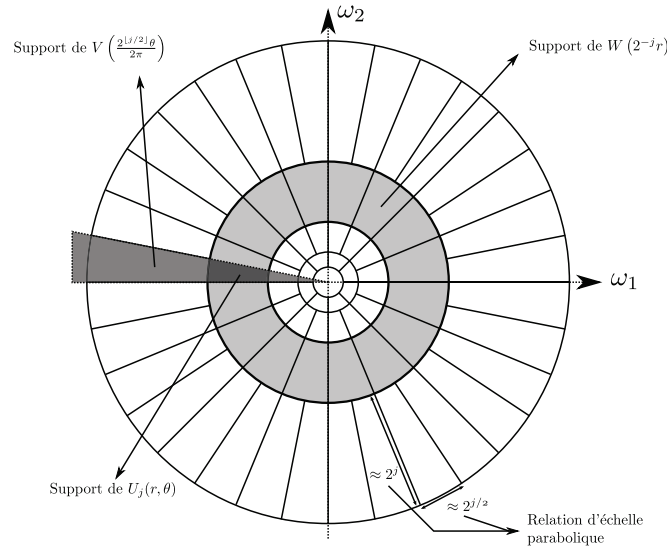


FIG. 2.11 – Découpage du domaine fréquentiel par la transformée en curvelets 2D continue. Visualisation des supports de $U_j(r, \theta)$, de $W_j(2^{-j}r)$ et de $V_j\left(\frac{2^{\lfloor j/2 \rfloor} \theta}{2\pi}\right)$

- **Comportement oscillatoire directionnel** : la curvelet mère φ_j capture les oscillations dans le sens de l'axe x et effectue un passe-bas dans le sens de l'axe y . Ainsi lorsque l'on applique la rotation R_{θ_ℓ} , on s'assure de capturer les oscillations dans la direction θ_ℓ et d'effectuer un passe-bas dans la direction orthogonale à celle-ci.
- **Bases optimales** de part sa construction, la transformée en curvelets est optimale pour la représentation des courbes C^2 [Candès 04, Candès 05a].

Pour plus d'informations sur les propriétés et la construction de la transformée en curvelets continue, nous invitons le lecteur à se référer à [Candès 05c, Candès 05d, Candès 03]. Comme nous souhaitons caractériser des séquences d'images numériques, nous devons nous placer dans le cadre de transformées discrètes. Nous effectuons maintenant une présentation de la transformée en curvelets dans le cadre discret.

2.2.2 Curvelets 2D discrètes

2.2.2.1 Description

Deux implémentations, présentées dans la section 2.2.2.2, de la transformée en curvelets dite de deuxième génération existent. Dans les deux cas, les coefficients de la transformée en curvelets 2D discrètes sont obtenus à partir d'une image numérique $f[\mathbf{x}]$, version discrétisée de l'équation 2.26 :

$$c^D(j, \ell, \mathbf{k}) = \sum_{\mathbf{x}} f[\mathbf{x}] \overline{\varphi_{j, \ell, \mathbf{k}}^D[\mathbf{x}]} \quad (2.37)$$

Comme dans le cas continu, l'atome de curvelet $\varphi_{j,\ell,\mathbf{k}}^D$ est défini dans le domaine fréquentiel par la moyenne de ses transformées de Fourier :

$$\widehat{\varphi}_{j,\ell,\mathbf{k}}^D = \widetilde{U}_{j,\ell}(\boldsymbol{\omega}) e^{i(\mathbf{x}_{\mathbf{k}}^{(j,\ell)}, \boldsymbol{\omega})} \quad (2.38)$$

où $\widetilde{U}_{j,\ell}$ représente la version discrète de $U_{j,\ell}$. La fenêtre, U_j , extrait les fréquences des couronnes dyadiques $\{2^j \leq r \leq 2^{j+1}\}$ et des angles $\{-\pi 2^{-j/2} \leq \theta \leq \pi 2^{-j/2}\}$ (voir équation 2.29). Seulement les couronnes et les rotations ne sont pas totalement adaptées à la discrétisation. Il faut donc remplacer ces concepts par leurs équivalents discrets : ici, les couronnes sont définies à partir de carrés concentriques (au lieu de cercles) qui sont ensuite découpés en secteurs. Ainsi, la famille $\{W_j\}_{j>0}$, $W_j(r) = W_j(2^{-j}r)$ se définit de la manière suivante :

$$\{\widetilde{W}_j\}_{j>0}, \quad \widetilde{W}_j(\boldsymbol{\omega}) = \sqrt{\Phi_j^2(\boldsymbol{\omega}) - \Phi_{j-1}^2(\boldsymbol{\omega})} \quad (2.39)$$

où Φ est définie par le produit de fenêtres passe-bas de dimension 1 : $\Phi_j(\boldsymbol{\omega}) = \phi(2^{-j}\omega_1)\phi(2^{-j}\omega_2)$. La fonction ϕ respecte $0 \leq \phi \leq 1$, est égale à 1 sur $[-1; 1]$ et à 0 sur $]-\infty; 2]$ et sur $[2; +\infty[$. Comme dans le cas continu, une approximation pour la résolution la plus basse est calculée (voir équation 2.32), et vérifie la relation suivante :

$$\Phi_0^2(\boldsymbol{\omega}) + \sum_{j>0} \widetilde{W}_j^2(\boldsymbol{\omega}) = 1 \quad (2.40)$$

Les supports définis par les éléments $\{\widetilde{W}_j\}_j$ définissent alors un pavage comme illustré par la figure 2.12.(a).

Comme les rotations ne sont pas adaptées au cas discret, le découpage angulaire V_j est aussi adapté. Ainsi la construction de $\widetilde{V}_{j,\ell}$ s'effectue par cadrants (Nord, Est, Sud, Ouest), comme le montre la figure 2.12.(b). Il est en effet beaucoup plus facile de définir un découpage angulaire pour un quadrant. Par exemple, pour le quadrant Est, l'équivalent discret de la famille $\{V_j\}_{j>0}$, $V_j(\theta) = V_j\left(\frac{2^{\lfloor j/2 \rfloor} \theta}{2\pi}\right)$ est :

$$\{\widetilde{V}_{j,\ell}\}_{j>0,\ell}, \quad \widetilde{V}_{j,\ell}(\boldsymbol{\omega}) = \phi\left(\frac{2^{\lfloor j/2 \rfloor} \omega_2 - \alpha_\ell \omega_1}{\omega_1}\right) \quad (2.41)$$

où $\alpha_\ell = 2\ell + 1 - 2^{\lfloor j/2 \rfloor}$ représente le centre du $\ell^{\text{ième}}$ secteur en partant du haut du quadrant. Les autres cadrants s'obtiennent en effectuant un changement des axes ω_1 et ω_2 . La figure 2.12.(b) illustre pour chaque quadrant (Nord, Ouest, Sud, Est) un découpage avec j croissant (respectivement $j = 0, 1, j = 2, 3, j = 4, 5$ et $j = 6, 7$). Dans le quadrant Nord, pour $j = 0, 1$ on remarque que nous avons deux secteurs. Plus les échelles augmentent plus le nombre de subdivisions grandit.

À l'aide des familles $\{\widetilde{W}_j\}_j$ et $\{\widetilde{V}_{j,\ell}\}_{j>0,\ell}$, nous pouvons maintenant définir la famille de fenêtres $\{\widetilde{U}_{j,\ell}\}_{j,\ell}$ par :

$$\{\widetilde{U}_{j,\ell}\}_{j,\ell}, \quad \begin{aligned} j = 0, & \quad \widetilde{U}_0(\boldsymbol{\omega}) = \widetilde{W}_0(\boldsymbol{\omega}) \\ j > 0, & \quad \widetilde{U}_{j,\ell}(\boldsymbol{\omega}) = \widetilde{W}_j(\boldsymbol{\omega}) \widetilde{V}_{j,\ell}(\boldsymbol{\omega}) \end{aligned} \quad (2.42)$$

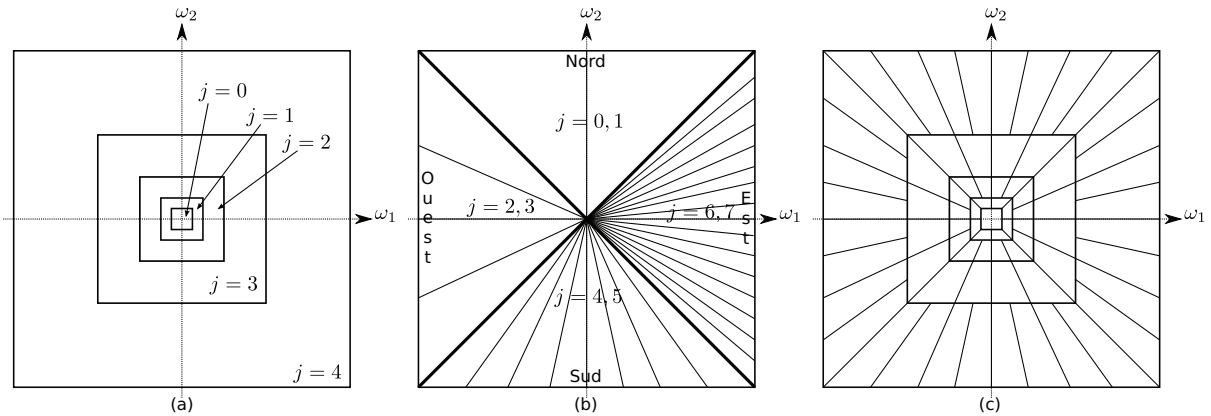


FIG. 2.12 – Pavage du domaine fréquentiel par la transformée en curvelets 2D discrète (c) obtenue à l'aide des fenêtres $\tilde{U}_{j,\ell}$. (a), respectivement (b), représente le pavage radiale \tilde{W}_j respectivement angulaire $\tilde{V}_{j,\ell}$.

Le découpage engendré par la famille $\{\tilde{U}_{j,\ell}\}_{j,\ell}$ est illustré par la figure 2.12.(c).

À cause du pavage qu'elle engendre, l'implémentation de la transformée en curvelets est délicate. Deux méthodes différentes ont été mises en place [Candès 05b] :

- transformée en curvelets via l'algorithme *unequispaced FFTs* (USFFT).
- transformée en curvelets via l'algorithme *wrapping*.

2.2.2.2 Implémentations

Transformée en curvelet via l'algorithme unequispaced FFTs (USFFT) La première implémentation consiste principalement à effectuer une rotation du spectre de f dans le domaine fréquentiel. Une interpolation permet de calculer précisément les valeurs dans la fenêtre $U_{j,\ell}$ d'analyse. Une transformée de Fourier 2D inverse est ensuite appliquée sur les coefficients sélectionnés.

L'algorithme suivant, ainsi que la figure 2.13 donne les différentes étapes de cette méthode :

1. calcul de la transformée de Fourier 2D pour obtenir les coefficients $\hat{f}[n_1, n_2]$;
2. pour chaque échelle j et chaque angle ℓ :
 - interpoler $\hat{f}[n_1, n_2]$ afin d'obtenir les valeurs de $\hat{f}[n_1, n_2 - n_1 \tan \theta_\ell]$;
 - multiplier les coefficients interpolés $\hat{f}[n_1, n_2 - n_1 \tan \theta_\ell]$ avec la fenêtre \tilde{U}_j afin d'obtenir les coefficients localisés dans la parallélogramme de l'échelle 2^{-j} et d'orientation θ_ℓ :

$$\tilde{f}_{j,\ell}[n_1, n_2] = \hat{f}[n_1, n_2 - n_1 \tan \theta_\ell] \tilde{U}_j[n_1, n_2]$$

- calcul de la transformée de Fourier 2D inverse sur $\tilde{f}_{j,\ell}$ afin de collecter les coefficients discrets $\varphi_{j,\ell,\mathbf{k}}^D$.

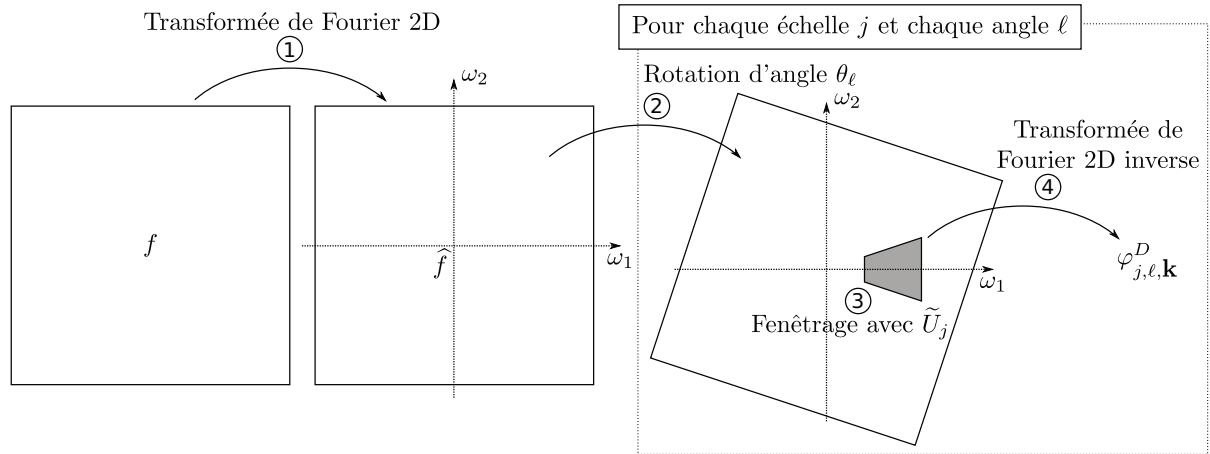


FIG. 2.13 – Les différentes étapes de l'implémentation USFFT de la transformée en curvelets.

Transformée en curvelet via l'algorithme wrapping Au contraire de l'implémentation par l'algorithme USFFT, cet algorithme effectue le fenêtrage directement à la bonne localisation fréquentielle, puis utilise la propriété de périodicité de la transformée de Fourier discrète afin d'obtenir les différents coefficients de la transformée en curvelets.

Les différentes étapes, illustrées par la figure 2.14, sont :

1. calcul de la transformée de Fourier 2D pour obtenir les coefficients $\hat{f}[n_1, n_2]$;
2. pour chaque échelle j et chaque angle ℓ :
 - calculer le produit $\tilde{U}_{j,\ell}[\omega_1, \omega_2] \hat{f}[\omega_1, \omega_2]$;
 - recaler ce produit autour de l'origine afin d'obtenir une zone d'analyse rectangulaire :

$$\tilde{f}_{j,\ell}[n_1, n_2] = \mathcal{W}(\tilde{U}_{j,\ell} \hat{f})[n_1, n_2]$$

où n_1 et n_2 varient comme $0 \leq n_1 \leq L_1$ et $0 \leq n_2 \leq L_2$. L_1 et L_2 correspondent à la largeur et à la hauteur du trapèze comme décrit sur la figure 2.15. L'opération de déformation consiste à utiliser la périodicité de la transformée de Fourier discrète afin de créer une grille rectangulaire plus facilement manipulable (cf. ci-après) ;

- calcul de la transformée de Fourier 2D inverse sur $\tilde{f}_{j,\ell}$ afin d'obtenir les coefficients discrets $\varphi_{j,\ell,\mathbf{k}}^D$.

L'opération de *wrapping* (étape 3 de la figure 2.14), illustrée sur la figure 2.15, s'effectue de la manière suivante :

1. sélection d'un parallélogramme de largeur L_2 et hauteur L_1 à recaler ;
2. répétition périodique de ce parallélogramme autour de l'origine ;
3. sélection d'un rectangle de largeur L_2 et hauteur L_1 centré sur l'origine. La transformée de Fourier 2D inverse s'effectue alors sur la fenêtre de dimension $L_1 \times L_2$.

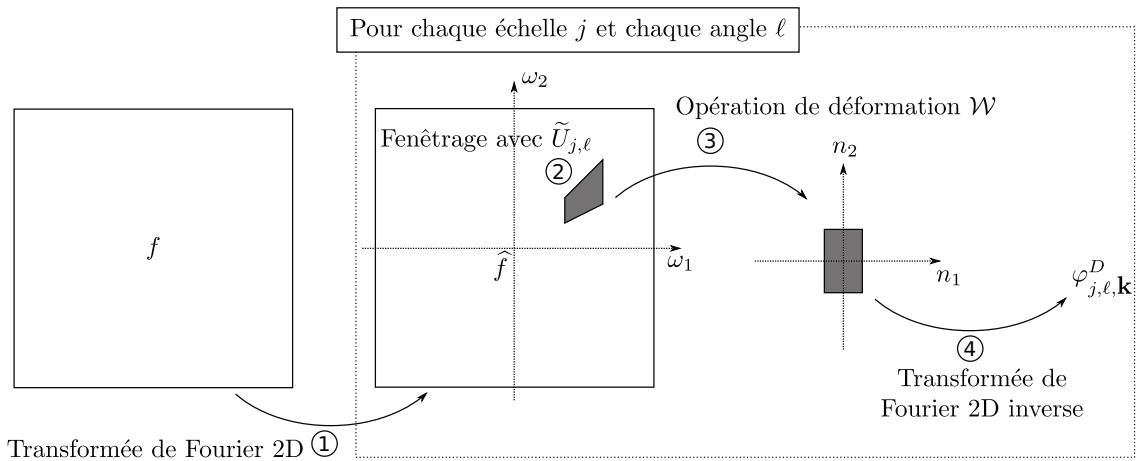


FIG. 2.14 – Les différentes étapes de l’implémentation par recalage de la transformée en curvelets par wrapping.

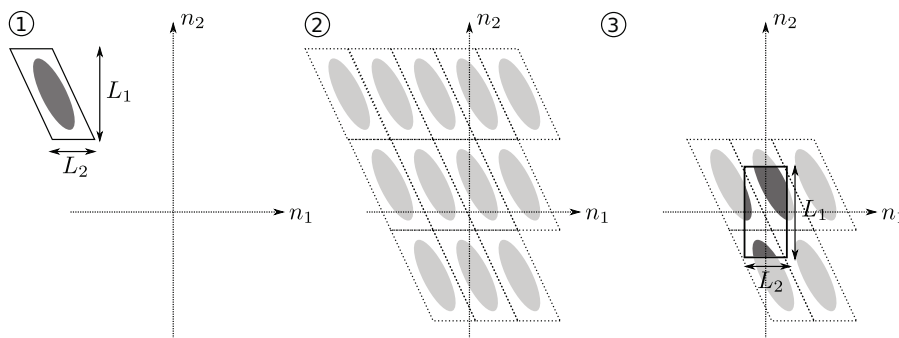


FIG. 2.15 – Opération de wrapping

2.2.2.3 Expérimentations

Atomes de curvelets La figure 2.16 illustre quelques atomes de curvelets $\varphi_{j,\ell,\mathbf{k}}$ dans le domaine fréquentiel et dans le domaine spatial à différentes échelles j et orientations ℓ .

Sur la dernière ligne de cette figure, l’atome à l’échelle la plus grossière, $j = 0$, est représenté. Ensuite, plus on remonte dans les lignes, plus l’échelle de l’analyse augmente. Ces différentes illustrations font ressortir les propriétés suivantes :

- la localisation spatiale et fréquentielle ;
- une des propriétés de la transformée en curvelets : l’observation du comportement oscillatoire directionnel, est visible. Le profil de l’atome de curvelet dans la direction θ_ℓ montre clairement le comportement oscillatoire, alors que dans la direction perpendiculaire, l’atome se comporte comme un noyau gaussien.

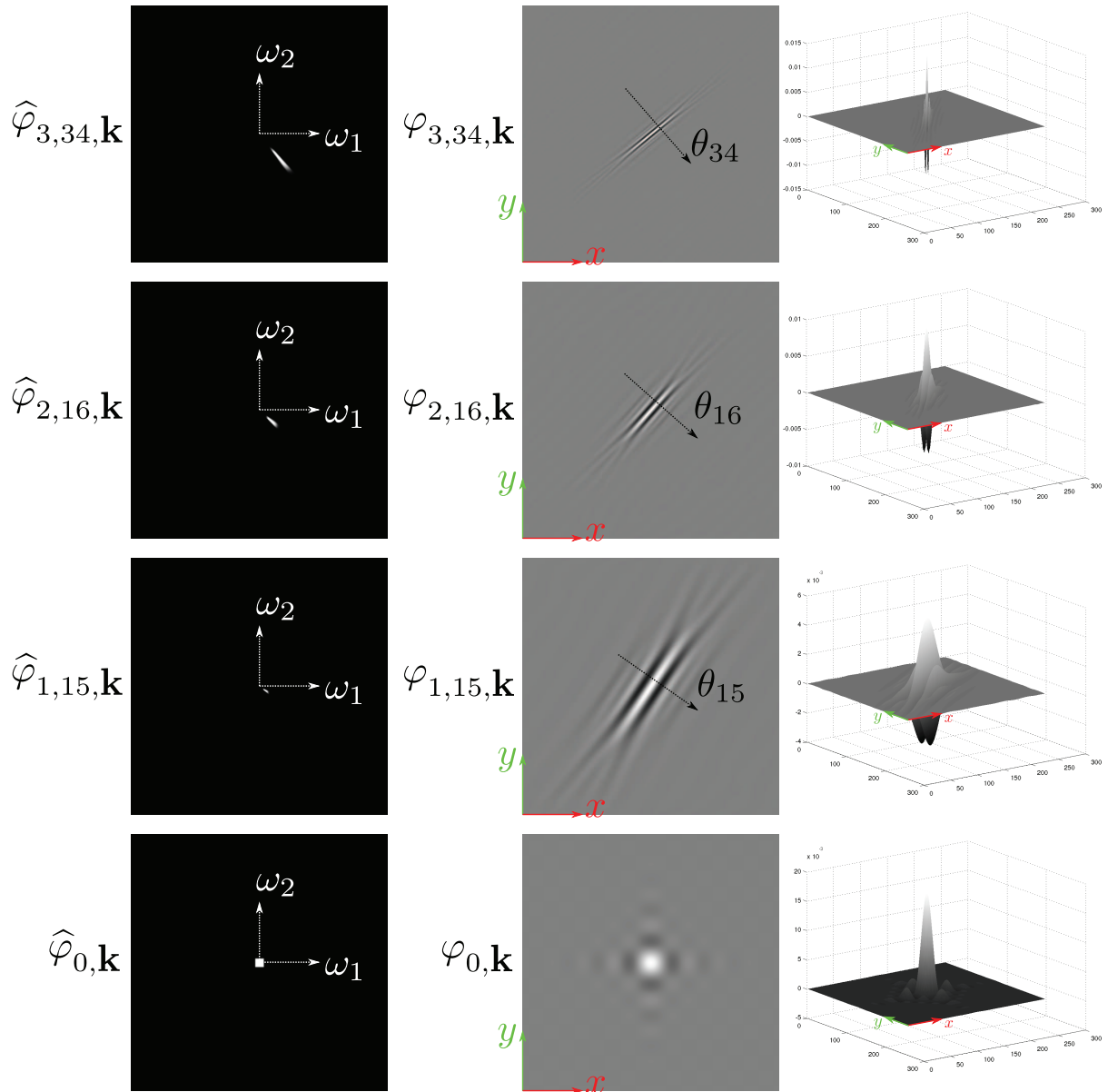


FIG. 2.16 – Atomes 2D de curvelets $\varphi_{j,\ell,k}$ à différentes échelles j et orientations ℓ .

Visualisation des coefficients de la transformée en curvelets La figure 2.17 montre le résultat obtenu par la transformée en curvelets sur une image issue d'une séquence d'images de la base DynTex. Le schéma de droite montre l'ensemble des coefficients de la transformée en curvelets résultant de la projection de la fonction f sur les atomes de curvelets $\varphi_{j,\ell,k}$.

Bien qu'il s'agisse d'une simple image, on peut remarquer qu'il n'est pas aisé de représenter l'ensemble des coefficients, contrairement à la transformée en ondelettes classique où l'ensemble des coefficients est stocké dans une matrice de même taille que l'image à décomposer.

Pour les dimensions supérieures (3D ou 2D+T), une représentation de ce type est impossible. Nous proposons donc d'autres formes de visualisation.

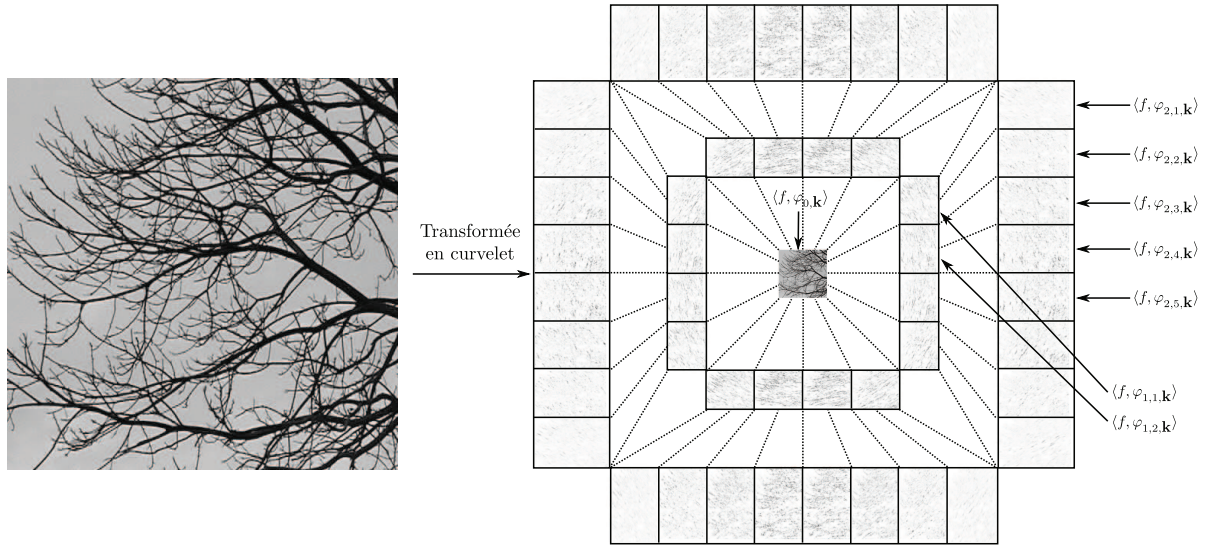


FIG. 2.17 – Résultat de la transformée en curvelets sur une image en utilisant 2 échelles de décomposition.

Parcimonie de la représentation en curvelets De part sa construction, la transformée en curvelets 2D est creuse pour la représentation des structures \mathcal{C}^2 d'une image. Autrement dit, pour représenter ces structures, il lui faudra moins de coefficients, par rapport à la transformée en ondelettes.

Afin d'illustrer cette propriété, le PSNR de la reconstruction partielle de la transformée en curvelets f_n^{curv} et de la transformée en ondelettes f_n^{wave} est calculé en sélectionnant un nombre croissant de coefficients. A chaque itération, la reconstruction de f est effectuée avec les n coefficients les plus pertinents afin d'obtenir $\{f_n^m\}_{m=\{curv,wave\}}$. Le PSNR est calculé afin de quantifier la qualité de l'estimée $\{f_n^m\}_{m=\{curv,wave\}}$ par rapport à l'originale f :

$$PSNR_{(dB)}^m(n) = 20 \log_{10} \left(\frac{255}{\sqrt{EQM_n^m}} \right) \quad \text{avec} \quad EQM_n^m = \frac{1}{N} \sum_{\mathbf{x}} (f[\mathbf{x}] - f_n^m[\mathbf{x}])^2 \quad (2.43)$$

où n représente les n coefficients les plus pertinents.

Pour la transformée en curvelets et la transformée en ondelettes Daubechies 8 appliquées sur l'image de la figure 2.17, les courbes de la figure 2.18 représentent le PSNR en fonction du nombre n de coefficients.

La lecture des courbes montre clairement que la transformée en curvelets permet une représentation plus parcimonieuse des images naturelles composées de structures \mathcal{C}^2 . Pour cette image, on peut remarquer qu'avec seulement 100 coefficients sélectionnés pour les deux transformées, on obtient une différence de 0.5dB sur les reconstructions.

Si plus de coefficients sont sélectionnés, 400 par exemple, un gain de 3dB, sur la reconstruction de la transformée en curvelets par rapport à celle de la transformée en ondelettes, est obtenue. Les premiers coefficients de la transformée en curvelets sont vraiment représentatifs de l'information et permettent d'approximer rapidement l'image.

Cette observation peut être également mise en évidence en regardant les reconstructions

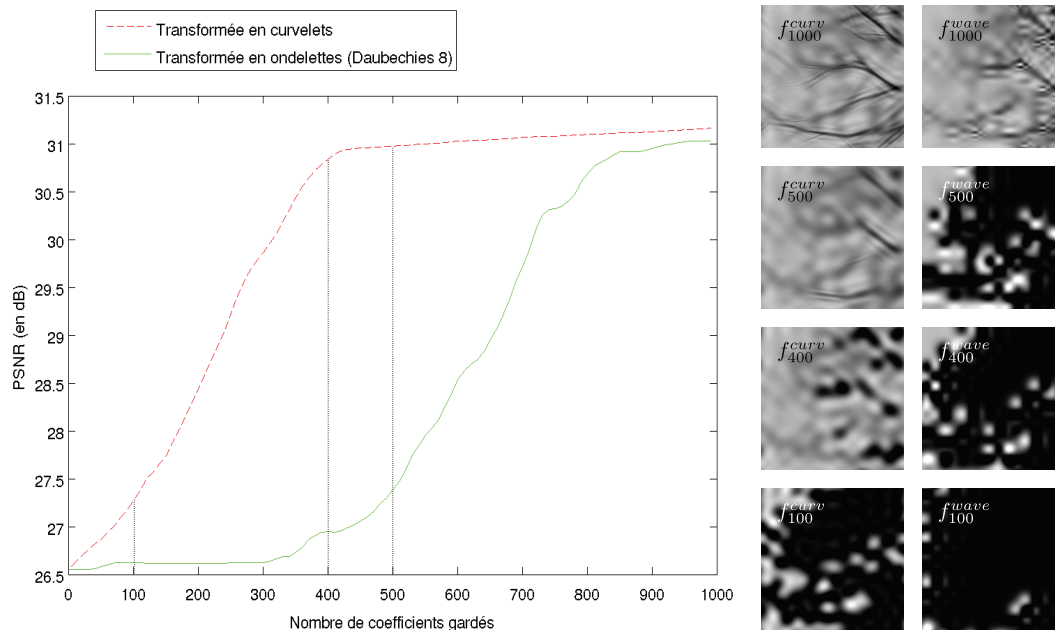


FIG. 2.18 – Comparaison de la parcimonie de la transformée en curvelets et de la transformée en ondelettes à l'aide du PSNR de la reconstruction partielle.

(partie droite de la figure 2.18) obtenues à l'aide des deux transformées. La transformée en curvelets sélectionne des atomes vraiment représentatifs de la géométrie de l'image, et sont donc, par ce fait, plus adaptés à l'image. On observe notamment qu'avec 500 coefficients, la forme géométrique de l'arbre est mieux reconstruite, ce qui n'est pas le cas avec la transformée en ondelettes pour ce même nombre de coefficients. La même constatation peut être effectuée pour 1000 coefficients : la forme géométrique de l'arbre, obtenue à l'aide de la transformée en curvelets, est mieux respectée et semble plus naturelle que celle obtenue avec l'autre transformée.

2.2.3 Transformée en curvelets tri-dimensionnelle

Dans les sections précédentes, les fondements de la transformée en curvelets¹ 2D, ainsi que certaines de ces propriétés, ont été présentés. La transformée en curvelets existe également pour des dimensions supérieures [Candès 05d], mais seul le cas 3D retiendra ici notre attention. En 2D, les curvelets sont adaptées à la capture des éléments présentant des lignes et des courbes. En 3D, quelles sont les structures retenues ?

La transformée en curvelets étant optimale pour la représentation d'objet de co-dimension 1, dans le cadre de séquence d'images, les structures optimales détectées sont des surfaces 2D.

¹Pour plus de détails sur la transformée en curvelets 2D, nous invitons le lecteur à se référer à [Candès 00a, Starck 00, Candès 04, Candès 05c, Candès 05d, Candès 05b, Fadili 09a]

2.2.3.1 Description

La réflexion précédente menée autour du découpage et de l'implémentation est identique dans le cas 3D. La transformée en curvelets 3D discrète du volume $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, x_3)$ correspondant maintenant aux coordonnées 3D, donne une collection de coefficients $c^D(j, \ell, \mathbf{k})$ définis par :

$$c^D(j, \ell, \mathbf{k}) = \sum_{x_1, x_2, x_3} f(\mathbf{x}) \overline{\varphi_{j, \ell, \mathbf{k}}^D(\mathbf{x})} \quad (2.44)$$

pour l'échelle $j \in \mathbb{Z}$, l'orientation $\ell \in \mathbb{Z}$ et la position $\mathbf{k} = (k_1, k_2, k_3)$.

La principale différence se situe dans la définition de la fenêtre $\tilde{U}_{j, \ell}$. En effet, celle-ci, ainsi que les deux fenêtres \tilde{W}_j et $\tilde{V}_{j, \ell}$, sont maintenant en trois dimensions. \tilde{W}_j consiste alors en un emboîtement de cubes concentriques. La famille $\{\tilde{W}_j\}_{j>0}$ s'écrit donc :

$$\{\tilde{W}_j\}_{j>0}, \quad \tilde{W}_j(\boldsymbol{\omega}) = \sqrt{\Phi_j^2(\boldsymbol{\omega}) - \Phi_{j-1}^2(\boldsymbol{\omega})} \quad \text{avec} \quad \Phi_j(\boldsymbol{\omega}) = \phi(2^{-j}\omega_1)\phi(2^{-j}\omega_2)\phi(2^{-j}\omega_3) \quad (2.45)$$

$\tilde{V}_{j, \ell}$ engendre un découpage angulaire 3D (un cône en quelque sorte). Celui-ci est orienté et paramétré par deux angles α_ℓ et β_ℓ . Ainsi la famille $\{\tilde{V}_{j, \ell}\}_{j>0, \ell}$ s'exprime par :

$$\{\tilde{V}_{j, \ell}\}_{j>0, \ell}, \quad \tilde{V}_{j, \ell}(\boldsymbol{\omega}) = \phi\left(\frac{2^{\lfloor j/2 \rfloor} \omega_2 - \alpha_\ell \omega_1}{\omega_1}\right) \phi\left(\frac{2^{\lfloor j/2 \rfloor} \omega_3 - \beta_\ell \omega_1}{\omega_1}\right) \quad (2.46)$$

La figure 2.19 illustre le découpage que l'on peut obtenir à partir des fenêtres \tilde{W}_j , $\tilde{V}_{j, \ell}$ et $\tilde{U}_{j, \ell}$ pour différentes échelles.

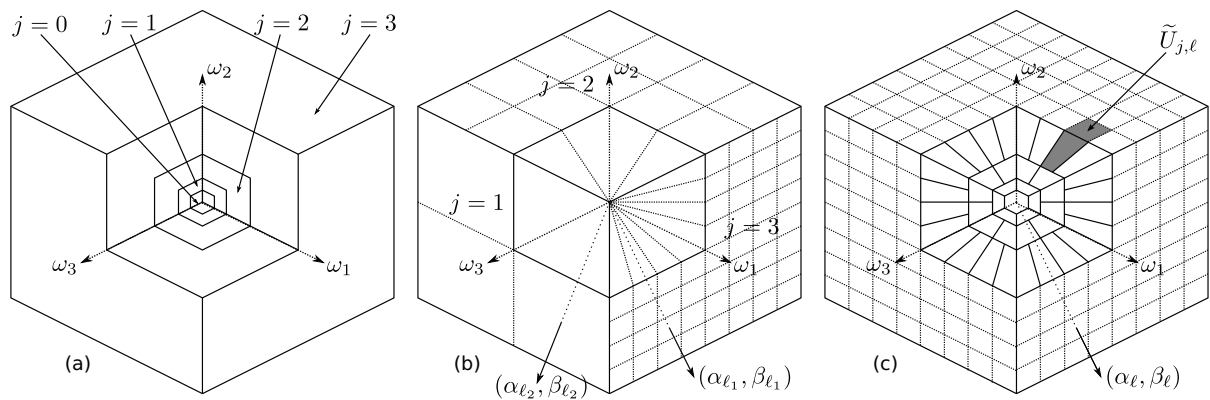


FIG. 2.19 – Pavage du domaine fréquentiel 3D de la transformée en curvelets 3D discrète (c) obtenue à l'aide des fenêtres $\tilde{U}_{j, \ell}$. La partie (a), respectivement la partie (b), représente le découpage radiale \tilde{W}_j , respectivement angulaire $\tilde{V}_{j, \ell}$.

Pour plus de détails, sur la transformée en curvelets 3D, nous invitons le lecteur à se référer à [Candès 05b, Ying 05].

Atomes de curvelets 3D La visualisation des coefficients des atomes de curvelets devient plus délicate. Dans le cas 2D, à l'aide d'un plan et d'une échelle de couleur, la représentation

des atomes est possible. L'ajout d'une dimension (passage au 3D) ne permet plus ce genre de visualisation. Toutefois, d'autres représentations sont envisageables. La figure 2.20 montre une visualisation différente d'atomes de curvelets à des échelles et orientations données.

Il s'agit d'une représentation à l'aide de coupes volumiques de l'atome est présentée. Celui-ci est maintenant un cube de données 3D, avec un coefficient pour chaque coordonnée $(\omega_1, \omega_2, \omega_3)$. A l'aide de trois plans orthogonaux, munis chacun d'une échelle de couleur, et d'un point $O(o_1, o_2, o_3)$, des coupes peuvent aussi être visualisées. Celles-ci permettent d'extraire les plans $(\vec{\omega}_1 O \vec{\omega}_2)$, $(\vec{\omega}_2 O \vec{\omega}_3)$, $(\vec{\omega}_3 O \vec{\omega}_1)$. Cette forme de visualisation montre les valeurs des coefficients dans trois plans d'origine commune O . Par contre, la structure et la forme de l'atome peuvent être difficiles à interpréter.

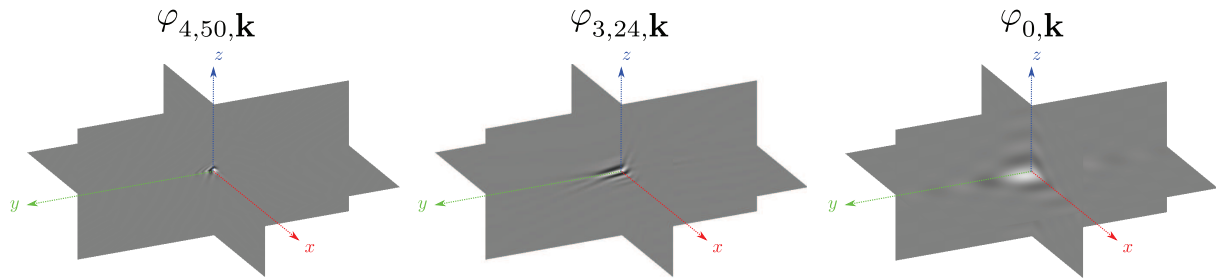


FIG. 2.20 – Atomes 3D de curvelets $\varphi_{j,\ell,\mathbf{k}}$ à différentes échelles j et orientations ℓ .

Parcimonie de la représentation en curvelets Dans le cas 2D, la transformée en curvelets s'avère plus parcimonieuse que la transformée en ondelettes pour la représentation d'une image naturelle. Cette propriété est valable également dans le cas 3D.

A chaque itération, une analyse multi-résolutions (transformée en curvelets ou en ondelettes) est effectuée sur une séquence d'images naturelle. Les n coefficients les plus pertinents sont ensuite sélectionnés, afin de reconstruire la séquence (f_n^{curv} pour la transformée en curvelets et f_n^{wave} pour la transformée en ondelettes). Le PSNR permet d'observer la qualité de la reconstruction $\{f_n^m\}_{m=\{curv,wave\}}$ par rapport à l'originale f .

La figure 2.21 illustre, sur une séquence "arbre", l'évolution du PSNR en fonction du nombre de coefficients retenus pour la transformée en curvelets et en ondelettes (pour cette dernière les ondelettes Daubechies 8 sont utilisées). Sur cette figure, quelques reconstructions $\{f_n^m\}_{m=\{curv,wave\}}$ sont également présentées.

La lecture des courbes montre, une nouvelle fois, que la transformée en curvelets est plus adaptée à la représentation de scènes naturelles par rapport à la transformée en ondelettes. En sélectionnant uniquement les 1500 coefficients les plus pertinents, on obtient une différence de plus de 3dB entre les deux transformées.

Concernant les premiers coefficients $n < 450$, le PSNR calculé sur la reconstruction par la transformée en ondelettes est meilleur que celui obtenu par la transformée en curvelets. Ceci s'explique par le fait que, lors des premières itérations, les coefficients sélectionnés permettent la construction des moyennes locales du signal d'origine f , ce qui a pour conséquence une augmentation rapide du PSNR. Ce phénomène est clairement observé dans les coupes réalisées pour f_{200}^{curv} et f_{200}^{wave} .

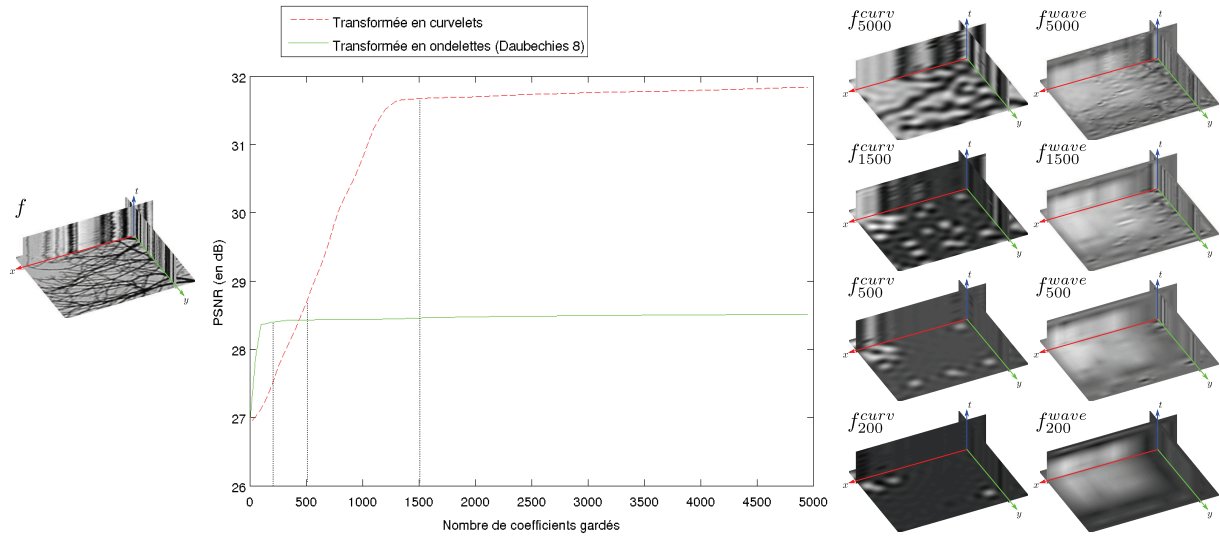


FIG. 2.21 – Comparaison de la parcimonie de la transformée en curvelets et de la transformée en ondelettes à l'aide du PSNR de la reconstruction partielle.

Dans la prise en compte des coefficients, $450 < n < 1300$, le PSNR de f_n^{curve} est en croissance rapide. Ceci s'explique par une sélection de coefficients très pertinents et représentatifs de la géométrie de l'arbre.

A partir de $n > 1300$, la croissance des PSNR de f_n^{curve} et de f_n^{wave} est moins significative. Ceci est dû à l'ajout progressif, pour chacune des transformées, des coefficients de détails permettant la convergence vers la séquence d'images originale. La progression du PSNR de f_n^{curve} est plus rapide car chaque coefficient ajouté affine mieux la géométrie de la séquence d'images originale par rapport aux coefficients choisis par la transformée en ondelettes.

2.2.3.2 Passage au 2D+T

Dans les sections précédentes, nous avons présenté la transformée en curvelets 2D et son extension à la 3D. Cette dernière est cependant définie pour des données tri-dimensionnelles (x, y, z) homogènes. Ce qui pose un certain nombre de difficultés dans le cas 2D+T (x, y, t) .

Le passage du 3D au 2D+t n'est pas trivial. En effet, dans le cas 3D, pour un voisinage donné, les distances entre le centre d'un pixel et ses voisins 6-connexes sont les mêmes. Autrement dit, $\Delta_x = \Delta_y = \Delta_z = 1$ comme le montre la figure 2.22.(a) si on représente un voisinage de pixel par un cube.

Par contre dans le cas 2D+t (figure 2.22.(b)), la distance spatiale (Δ_x et Δ_y) du parallélépipède est différente de la distance selon l'axe temporel (Δ_t). Il existe donc un rapport entre Δ_x , Δ_y et Δ_t que l'on peut écrire sous la forme :

$$\Delta_x = \Delta_y = c \cdot \Delta_t \quad (2.47)$$

avec c une constante permettant de rendre homogène les variables spatiales et temporelle.

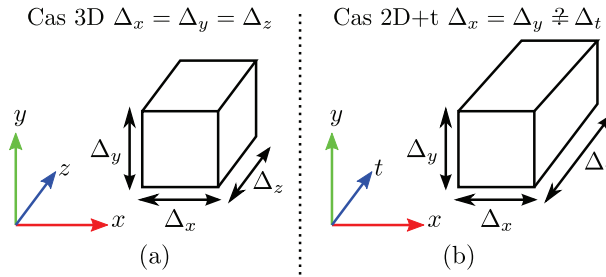


FIG. 2.22 – Illustration du problème d'homogénéité des variables spatiales et temporelles.

Dans la littérature, il est possible de trouver différentes utilisations de cette constante :

- $c = 1$: il s'agit du cas le plus simple et le plus souvent utilisé. Il consiste à dire que la relation entre des pixels spatio-temporels est la même qu'entre des pixels purement spatiaux. En général, l'homogénéité entre le spatial et le temporel n'est même pas évoquée.
- $c = ?$: c correspond à un facteur de dilatation suivant l'axe temporel. Le facteur de dilatation est déjà introduit lors de la construction de l'analyse multi-résolutions. Lorsque $c > 1$ on se focalise donc sur des échelles temporelles plus grandes.

Dans notre cas, pour simplifier et parce que l'étude de c est un cas complexe, nous fixerons pour le reste de ce manuscrit $c = 1$. Nous pondérerons toutefois l'influence de t par rapport à x et y dans le chapitre 3.2 pour un algorithme de segmentation s'appuyant sur la transformée en curvelets.

2.2.3.3 Application aux séquences d'images

Nous présentons dans cette section les résultats obtenus sur des séquences d'images. La contribution principale de cette étude porte sur le développement d'outils de visualisation et d'interprétation des coefficients de curvelets obtenus.

La visualisation des différents coefficients (comme sur la figure 2.17 pour le cas des images) n'est en effet plus possible directement. Dans le cas 2D, l'information "image" sur laquelle est calculée la transformée en curvelet est plongée dans un espace à 4 dimensions $(j, \ell, \omega_1, \omega_2)$. Par contre dans le cas 3D, cette information se place sur un espace 5 dimensions $(j, \ell, \omega_1, \omega_2, \omega_3)$. En effet, pour une échelle et une orientation donnée, un sous-volume de coefficients existe.

Pour des raisons de visualisation et de compréhension des phénomènes complexes présents dans des séquences d'images, le nombre de dimensions doit être réduit. En effet, le principe est de ne plus observer chaque coefficient, mais d'extraire une caractéristique globale ϑ sur ceux-ci, comme par exemple l'énergie des sous bandes, la moyenne, l'écart-type, ...

La représentation de cet espace à 3 dimensions (échelle j , orientation ℓ , caractéristique observée ϑ) peut s'effectuer de plusieurs manières. Ici, nous nous sommes intéressés à deux représentations différentes.

Représentation échelle par rapport à l'orientation La première des deux visualisations proposées consiste à représenter la caractéristique ϑ par une échelle de couleur. Nous obtenons alors un graphique ayant comme abscisse l'orientation ℓ et comme ordonnée l'échelle j (figure 2.23).

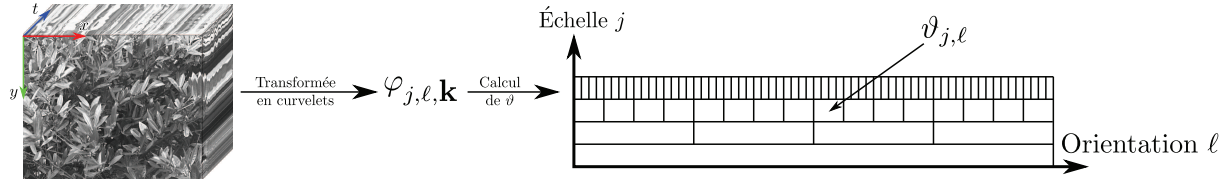


FIG. 2.23 – Principe de la représentation d'une caractéristique dans un espace 2D construit sur l'échelle et l'orientation

Cette représentation possède l'avantage d'écrire la distribution de la caractéristique en fonction de l'échelle et de l'orientation, mais ne prend pas en compte la position dans le cube fréquentiel. Comme il s'agit d'angles solides distribués selon un volume 3D, ici visualisés linéairement, l'information de proximité est nécessairement perdue.

Expérimentalement, nous nous sommes rendus compte que cette visualisation était souvent ambiguë, et par conséquent n'aidait pas à la représentation des phénomènes complexes mises en évidence pourtant par la transformée en curvelets. Il s'avère donc nécessaire de prendre en compte l'information de voisinage des orientations ℓ .

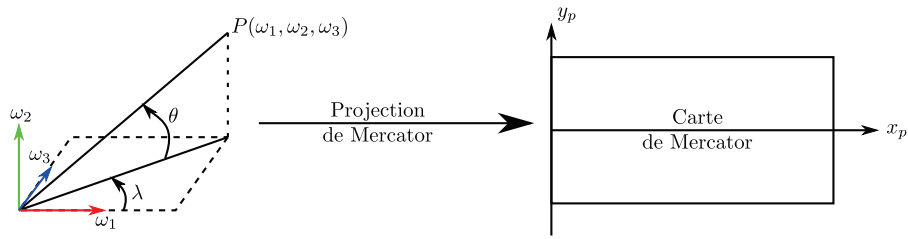
Visualisation par projection de Mercator Le deuxième outils mis en place se construit à partir de la projection de Mercator, classiquement utilisée en géographie pour la représentation des cartes de la Terre. La projection de Mercator est une projection cylindrique du globe terrestre sur une carte plane formalisée par Gerardus Mercator en 1569. Les parallèles et les méridiens sont des lignes droites, et l'inévitable étirement Est-Ouest en dehors de l'équateur est accompagné par un étirement Nord-Sud correspondant, de telle sorte que l'échelle Est-Ouest est partout semblable à l'échelle Nord-Sud. Une carte de Mercator ne peut couvrir les pôles : ils seraient infiniment hauts. Il s'agit d'une projection conforme, c'est-à-dire qu'elle conserve les angles.

La projection de Mercator (cf. figure 2.24) permet d'obtenir les coordonnées (x_p, y_p) sur une carte à deux dimensions en projetant la longitude λ et la latitude θ :

$$\forall \lambda \in]-\pi, \pi[, \forall \theta \in \left] -\frac{\pi}{2}, \frac{\pi}{2} \right[\left[\begin{cases} x_p = \lambda \\ y_p = \frac{1}{2} \ln \left(\frac{1 + \sin \theta}{1 - \sin \theta} \right) \end{cases} \right. \quad (2.48)$$

A partir des coordonnées 3D d'un point P , la longitude λ et latitude θ sont calculées à l'aide des équations suivantes :

$$\begin{aligned} \theta &= \arccos \left(\frac{y}{\sqrt{\omega_1^2 + \omega_2^2 + \omega_3^2}} \right) \\ \lambda &= \arctan 2(\omega_3, \omega_1) \end{aligned} \quad (2.49)$$

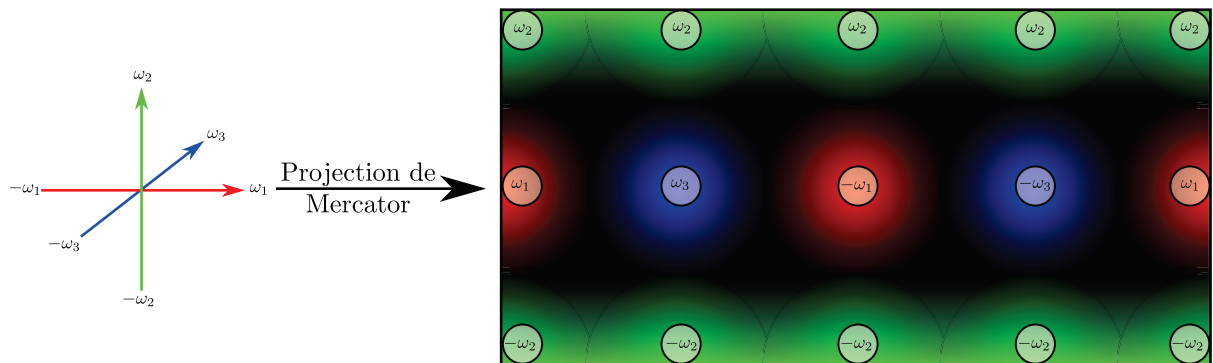
FIG. 2.24 – *Projection de Mercator*

avec :

$$\arctan 2(\omega_3, \omega_1) = \begin{cases} \arctan\left(\frac{\omega_3}{\omega_1}\right) & \omega_1 > 0 \\ \pi + \arctan\left(\frac{\omega_3}{\omega_1}\right) & \omega_1 < 0 \quad \omega_3 \geq 0 \\ -\pi + \arctan\left(\frac{\omega_3}{\omega_1}\right) & \omega_1 < 0 \quad \omega_3 < 0 \\ \frac{\pi}{2} & \omega_1 = 0 \quad \omega_3 > 0 \\ -\frac{\pi}{2} & \omega_1 = 0 \quad \omega_3 < 0 \\ \text{non défini} & \omega_1 = 0 \quad \omega_3 = 0 \end{cases} \quad (2.50)$$

La carte de Mercator permet donc le passage d'un espace à trois dimensions à un espace à deux dimensions dans lequel sont plongées les caractéristiques ϑ (énergie, entropie, ...) des coefficients de la transformée en curvelets. Pour une échelle j donnée, l'orientation ℓ d'une caractéristique ϑ indique dans le cadre fréquentiel une longitude et une latitude. Nous avons choisi de représenter la valeur de la caractéristique sur une échelle de couleur.

Une carte est ainsi obtenue sur chaque échelle de la décomposition. L'avantage par rapport à la représentation précédente est la préservation de l'information de voisinage de chaque angle. Par contre, nous n'obtenons pas une idée précise de comment s'imbriquent les angles entre les échelles. Contrairement à la représentation précédente, nous arrivons à compenser cet inconvénient facilement.

FIG. 2.25 – *Cartographie d'une projection de Mercator*

La figure 2.25 illustre la projection de l'espace à trois dimensions obtenue. Cette cartographie

facilite la compréhension des projections réalisées sur des séquences d'images.

Visualisation de l'énergie à travers les échelles Que ce soit dans le contexte applicatif de la caractérisation ou bien de la segmentation, l'énergie des sous-bandes de détails aux différentes échelles d'une analyse multi-résolutions est une caractéristique classiquement utilisée. Le choix de ϑ s'est donc naturellement portée sur celle-ci. L'analyse s'effectue à l'aide de la projection de Mercator. L'énergie dans le domaine transformée est calculée comme suit :

$$\forall j, \forall \ell \quad \vartheta_E^{(j,\ell)} = \frac{1}{t_1^{(j,\ell)} t_2^{(j,\ell)} t_3^{(j,\ell)}} \sum_{k_1=1}^{t_1^{(j,\ell)}} \sum_{k_2=1}^{t_2^{(j,\ell)}} \sum_{k_3=1}^{t_3^{(j,\ell)}} |\langle f, \varphi_{j,\ell,\mathbf{k}} \rangle|^2 \quad (2.51)$$

avec $t_1^{(j,\ell)}$, $t_2^{(j,\ell)}$ et $t_3^{(j,\ell)}$ la taille de la sous-bande de coefficients pour l'échelle j et l'orientation ℓ .

La figure 2.26 montre le résultat de la projection de Mercator de l'énergie des sous-bandes de détails de la transformée en curvelets sur des vidéos synthétiques. Ces vidéos ont été synthétisées afin de faire réagir les atomes de curvelets uniquement selon une direction précise (direction x , y ou t). Pour cela, le cube 3D est créé à partir d'un empilement de strates dans la direction désirée.

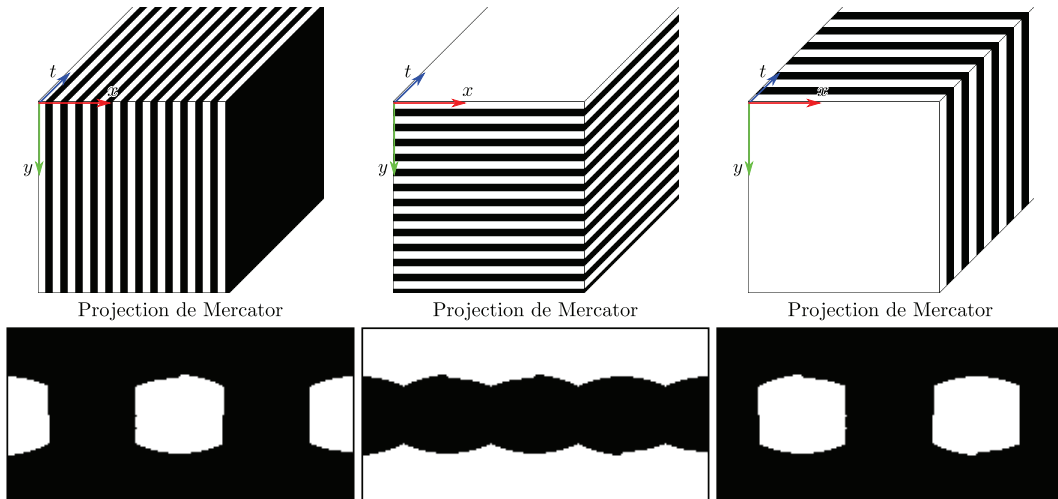


FIG. 2.26 – *Projection de Mercator des énergies des sous-bandes de détails de la transformée en curvelets de séquences d'images synthétiques.*

Nous présentons ici (figure 2.26) la carte de Mercator d'une seule échelle (fonction de la fréquence d'empilement), les autres possédant une énergie peu significative. En comparant les zones d'énergies des cartes, à la cartographie établie par la figure 2.25, on remarque que la transformée en curvelets réagit conformément à ce que nous pouvions attendre. Dans chacun des cas, la direction perpendiculaire aux strates est bien mise en évidence par la carte de la projection de Mercator.

La figure 2.27 montre les cartes de Mercator de l'énergie d'une séquence d'images composée de l'empilement d'une même image.

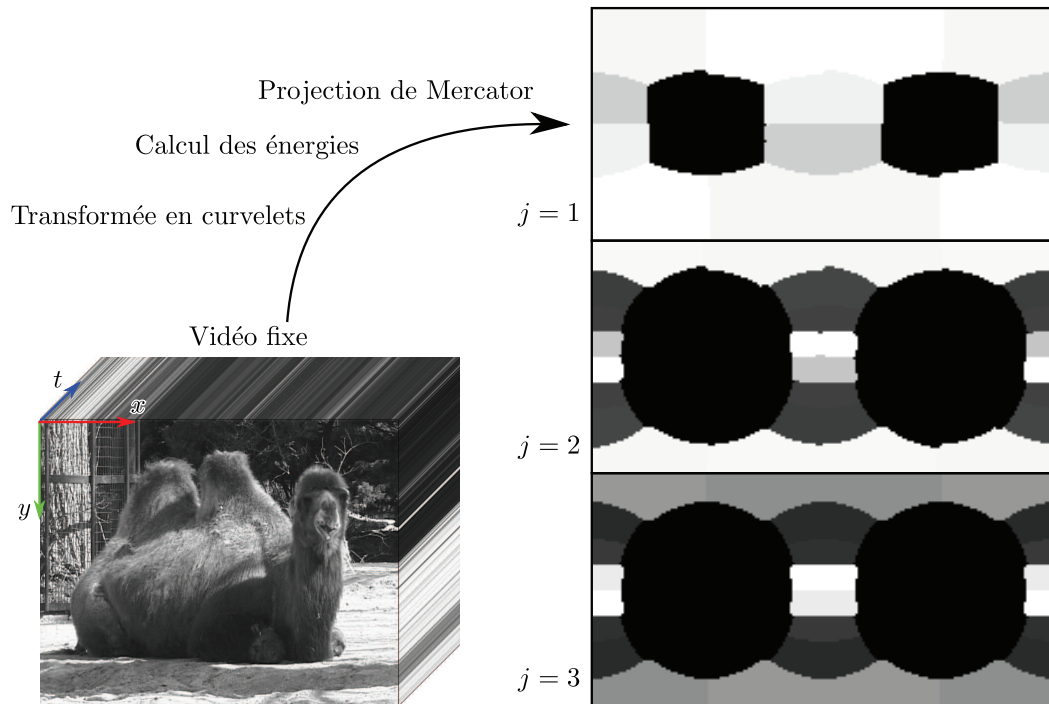


FIG. 2.27 – Cartes de Mercator des énergies des sous-bandes de détails de la transformée en curvelets d'une séquence d'images fixes, ne présentant donc aucun mouvement.

On remarque clairement que la transformée en curvelets ne réagit que sur les directions x et y . En effet, les zones mises en évidence (claires) sur les cartes de Mercator correspondent à des atomes de curvelets purement spatiaux. Sur cet exemple, on observe également la relation intra-échelle. La zone couverte par une portion de l'énergie se subdivise en quatre lors du passage de l'échelle $j = 1$ à $j = 2$. Les limites de cette représentation (la représentation des échelles les unes à côté des autres) précisées précédemment ne sont aucunement gênantes ici.

Nous illustrons maintenant notre approche sur des séquences d'images de scènes réelles issues de DynTex.

Tout d'abord, nous analysons la séquence d'un "escalator" vu du dessus et de près (vidéo 54pe210 de DynTex).

Sur la figure 2.28, l'énergie présente dans la sous-bande de détails d'une échelle ainsi que la reconstruction de la séquence à une échelle donnée sont présentées. Une énergie importante, à la fois dans la direction temporelle et horizontale, prédomine à travers toutes les échelles. Ce phénomène est dû aux stries verticales des marches de l'escalator se propageant dans le temps. Les stries d'une marche sont distribuées horizontalement. Du fait de leurs contours, ces stries possèdent donc dans cette direction une forte énergie. De plus, leur déplacement au cours du temps génère une apparition d'énergie selon cet axe. Les changements verticaux sont faibles ce qui explique les énergies peu significatives au niveau des pôles des cartes.

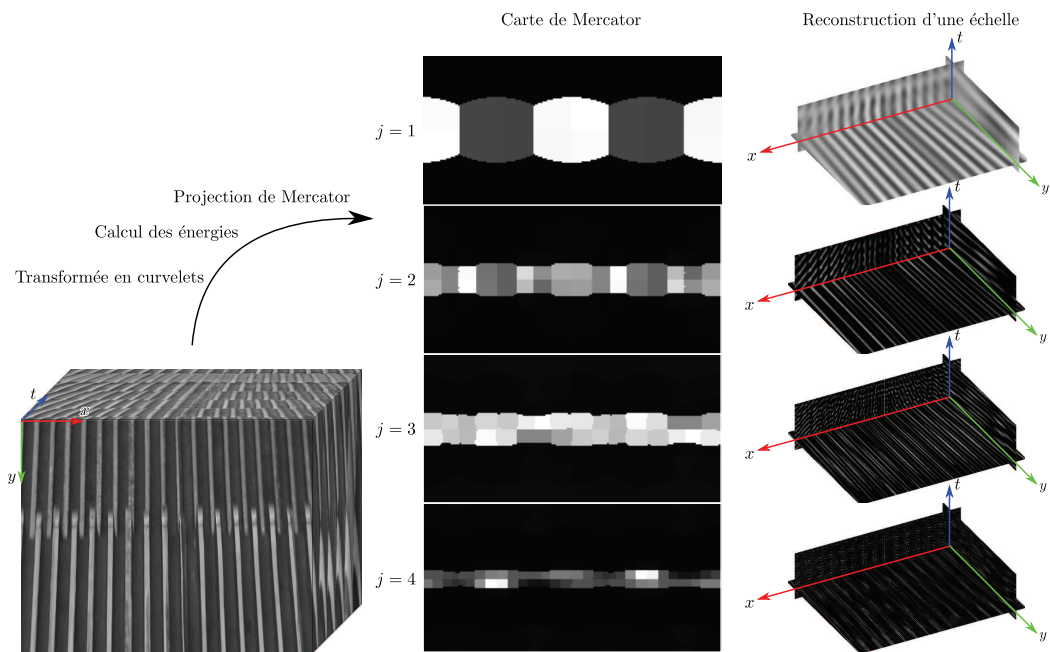


FIG. 2.28 – ♠ Cartes de Mercator des énergies des sous-bandes de détails de la transformée en curvelets sur la séquence d'images d'un escalator.

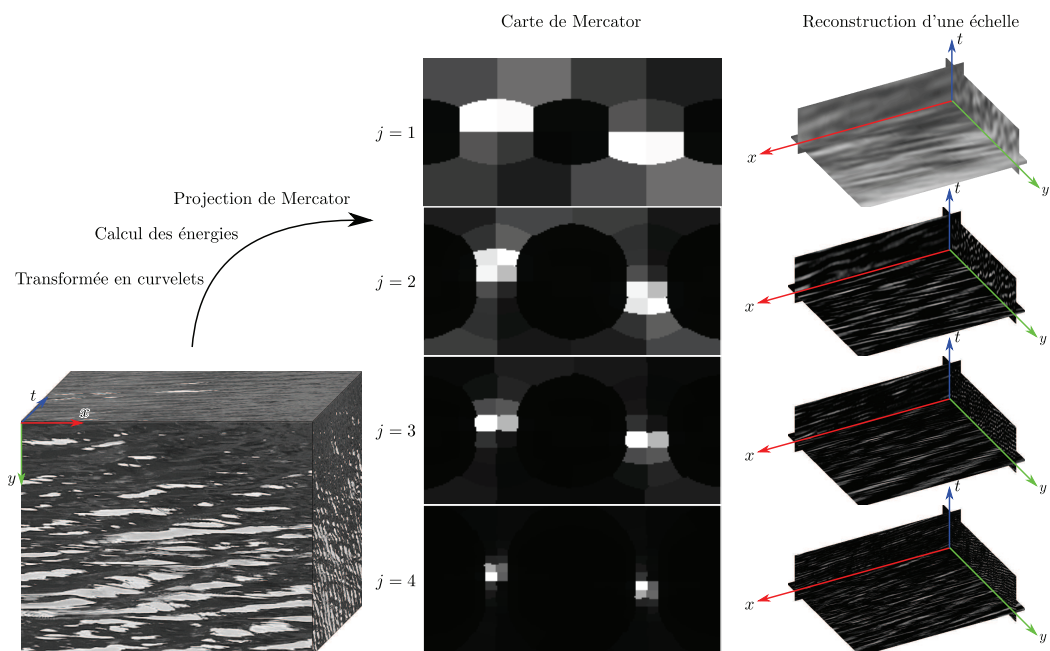


FIG. 2.29 – ♠ Cartes de Mercator des énergies des sous-bandes de détails de la transformée en curvelets sur la séquence d'images de la surface d'un lac.

La deuxième séquence que nous présentons est celle d'un lac (vidéo 54pf110 de DynTex) dont la surface présente des oscillations se propageant du haut de l'image vers le bas (du fond vers le devant de la scène). Comme dans le cas précédent, la figure 2.29 présente les cartes de Mercator des énergies ainsi que la reconstruction pour chaque sous-bande de détails de la transformée en curvelets. Cette fois-ci, l'énergie est principalement contenue dans les directions verticale et temporelle. L'énergie dans la direction verticale s'explique par la direction des risées de la surface de l'eau. Sur la direction temporelle, l'énergie est plus importante que sur la figure précédente car le phénomène est beaucoup plus rapide (i.e. la périodicité temporelle est beaucoup plus courte).

La dernière séquence (54ac210 de DynTex) représentée par la figure 2.30 est un champ d'herbes ondulant au vent. Sur les échelles basses, quasiment toutes les directions sont mise en évidence par la transformée en curvelets. Ce phénomène s'explique par les mouvements complexes provoqués par l'enchevêtrement de l'ensemble des herbes. Par contre, plus les échelles augmentent, plus l'énergie se concentre autour de l'axe temporel, ce qui traduit le mouvement très rapide des herbes dans le vent.

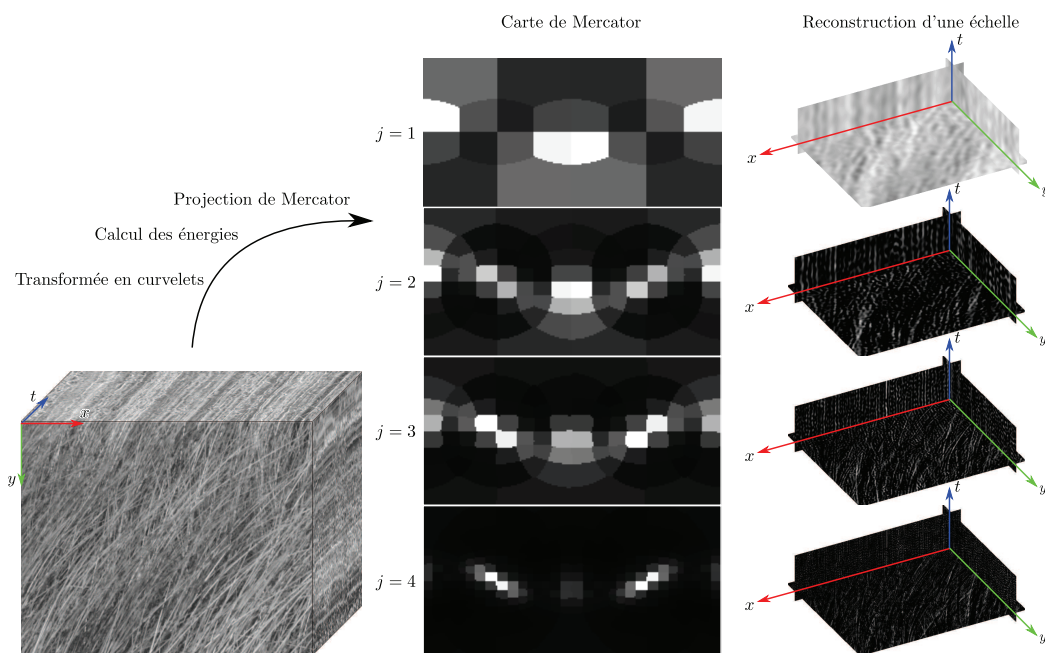


FIG. 2.30 – ♠ Cartes de Mercator des énergies des sous-bandes de détails de la transformée en curvelets sur la séquence d'images d'un champ d'herbes.

Projection d'autres quantités $\vartheta^{(j,\ell)}$ A l'aide des cartes de Mercator, toutes les caractéristiques mesurables des sous-bandes de détails de la transformée en curvelets peuvent ainsi être interprétées : projection de l'écart-type, moyenne, entropie, ...

Conclusion du chapitre Différentes transformées multi-échelles pour l'analyse des séquences d'images ont été présentées. Après une introduction de l'analyse multi-résolutions dans le cas mono-dimensionnel, nous avons abordé quatre transformées en ondelettes 2D+T. Elle diffèrent par la façon dont sont prises en compte les variables spatiales et temporelle dans la transformée.

Nous avons ensuite présenté une transformée multi-échelles géométrique non-adaptative : la transformée en curvelets. Après avoir évoqué la théorie et la mise en oeuvre de son calcul dans le cas 2D, puis 3D, nous nous sommes focalisés sur les problèmes de visualisation de l'espace des coefficients de la transformée en curvelets.

Fort de ces outils d'analyses 2D+T, le chapitre suivant se concentre sur l'étude des textures dynamiques. Pour cela, nous proposons un modèle les formalisant, et facilitant la compréhension des phénomènes complexes oscillants sous-jacents.

CHAPITRE 3

Étude des textures dynamiques

Étude des textures dynamiques

Sommaire

3.1	Un modèle de textures dynamiques	90
3.1.1	Définition du modèle	90
3.1.2	Évaluation du modèle de textures dynamiques proposé	93
3.1.3	Perspectives du modèle	94
3.2	Étude de l'onde porteuse	95
3.2.1	Caractéristiques de l'onde porteuse	95
3.2.2	Segmentation spatio-temporelle utilisant directement la transformée en curvelets 2D+T	98
3.2.2.1	Schéma général	98
3.2.2.2	Resultats et discussion	100
3.2.2.3	Perspectives	101
3.2.3	Segmentation spatio-temporelle à l'aide d'une structure en octree et de la transformée en curvelets 2D+T	101
3.2.3.1	Schema général	101
3.2.3.2	Discussion sur les critères d'homogénéité	104
3.2.3.3	Résultats	107
3.2.3.4	Perspectives concernant la structure en octree	112

Dans la littérature, les textures dynamiques sont le plus souvent décrites comme des phénomènes possédant une certaine répétitivité à la fois spatiale et temporelle. Cette définition est peu précise mais permet d'englober l'ensemble des textures dynamiques. Il est toutefois nécessaire de la formaliser.

Dans ce chapitre, nous nous focalisons sur certaines classes de textures dynamiques souvent rencontrées, et proposons un modèle inspiré de la synthèse d'images et de l'observation de la base DynTex ; les textures dynamiques sont ainsi perçues comme la superposition de plusieurs phénomènes physiques.

Une fois le modèle défini, nous étudions plus particulièrement une de ces composantes afin, d'une part, de valider notre approche, et d'autre part d'analyser si cette composante est, à elle seule, discriminante. Deux méthodes de segmentation, au sens des textures dynamiques, sont ensuite présentées et appliquées à des séquences d'images synthétiques et naturelles.

3.1 Un modèle de textures dynamiques

3.1.1 Définition du modèle

La base DynTex offre une grande diversité de textures dynamiques. Des exemples sont présentés sur les figures 1.1 et 1.8 de la section 1.1. De part la variabilité des phénomènes sous-jacents, les textures dynamiques sont difficiles à définir. Nous nous appuyons sur la définition souvent rencontrée dans la littérature (phénomène variant dans le temps et possédant une certaine répétitivité à la fois spatiale et temporelle) et sur celle que nous avons proposée (cf. définitions 5 et 6) pour essayer d’apporter une première formalisation.

Dans l’objectif de les indexer, ou de les analyser, il est primordial de comprendre les différents phénomènes sous-jacents. De la compréhension des phénomènes physiques, direction, rapidité de propagation, stationnarité, nous espérons tirer un certain nombre d’enseignements nous permettant d’optimiser la création de descripteurs.

L’objectif de la section suivante est donc de proposer un modèle formel pour une catégorie répandue de textures dynamiques.

Composante texture dynamique d’une vidéo Si nous observons les séquences d’images présentées sur la figure 3.1, issues de la base DynTex, plusieurs constatations peuvent être effectuées :

- une texture dynamique possède un support spatio-temporel différent de celui de la vidéo (c’est le cas par exemple de la séquence (a)).
- plusieurs textures dynamiques peuvent être présentes sur une même vidéo : séquences (b), (c) et (d).

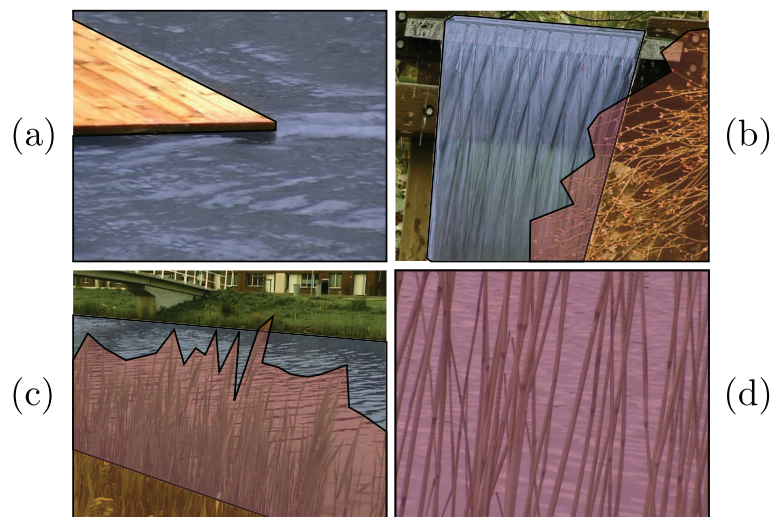


FIG. 3.1 – ♠ Exemples de textures dynamiques avec leur support respectif. Les colorations bleue et rouge correspondent à des supports spatio-temporels de textures dynamiques différents ; la coloration violette correspond à l’intersection de deux supports.

- les supports des différentes textures dynamiques peuvent être disjoints. Certaines textures dynamiques sont en effet partiellement transparentes, comme l’eau (séquence (d)), la fumée, ... D’autres sont composées d’éléments discrets et peu denses laissant apparaître l’arrière-plan comme les herbes (séquences (c) et (d)), les arbres (séquence (b)), ...

De plus, comme nous l’avons établi dans le chapitre 1, une séquence d’images naturelle, artificielle ou synthétique peut contenir une composante texture statique et/ou une composante texture dynamique. Cette dernière est composée d’au moins une texture dynamique.

Ces constatations nous conduisent à la définition suivante :

Définition 7. La composante texture dynamique d’une vidéo T_V se définit comme la somme de N textures dynamiques $\Upsilon_i \in \mathbb{R}$, chacune de support spatio-temporel Ω_i pouvant être disjoints :

$$T_V(\mathbf{x}) = \sum_{i=1}^N \Upsilon_i^{\Omega_i}(\mathbf{x}) \quad (3.1)$$

où $\mathbf{x} = (x, y, t)^T$ représente les coordonnées d’un voxel dans le cube vidéo. Nous nous intéressons ensuite à la formalisation d’une texture dynamique Υ_i .

Texture dynamique La figure 3.2 représente des coupes spatio-temporelles de deux textures dynamiques : “Surface de la mer” et “Arbre dans le vent”.

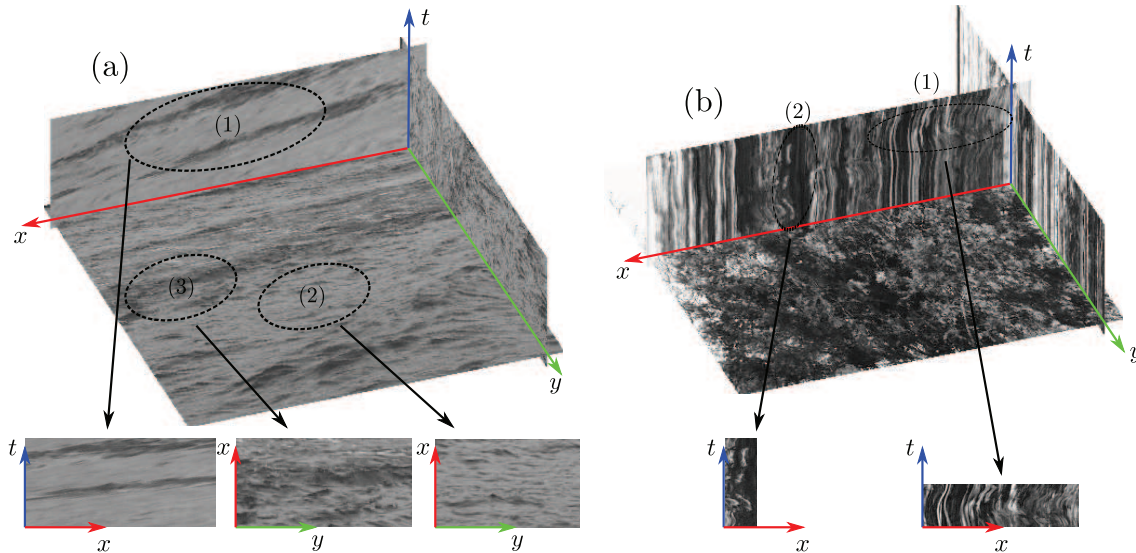


FIG. 3.2 – ♠ Coupes spatio-temporelles de textures dynamiques. On peut observer l’onde porteuse (1), les phénomènes locaux (2) et l’interférence entre les deux (3) sur les séquences “Surface de la mer” et “Arbre dans le vent”.

Sur chacune de ces coupes 2D+T, nous observons trois phénomènes distincts :

- une onde porteuse (1) : onde à grande échelle se propageant au cours du temps. Ce phénomène “porte” l’information de la texture.
- des phénomènes localement oscillants (2) : de petites oscillations bien localisées dans la séquence d’images, de support réduit, et de fréquences élevées, apparaissant de manière stochastique.
- la superposition des deux phénomènes observés (3) : en effet, à certains moments des séquences, les deux phénomènes observés s’interfèrent.

De ces observations, pouvant s’étendre sur de nombreuses séquences d’images de la base DynTex, et de résultats récents obtenus en synthèse d’images [Finch 04], nous proposons la définition suivante :

Définition 8. *Une texture dynamique Υ_i peut s’écrire comme la composition d’une onde se propageant à grande échelle et de phénomènes localement oscillants ; elle suit alors le modèle suivant :*

$$\forall i, \quad \Upsilon_i^{\Omega_i}(\mathbf{x}) = \mathcal{P}_i(\mathbf{x}) + \mathcal{L}_i(\mathbf{x}) \quad (3.2)$$

où $\mathbf{x} = (x, y, t)^T$ représente les coordonnées d’un voxel dans le cube vidéo. \mathcal{P}_i et \mathcal{L}_i sont deux fonctions décrivant respectivement l’onde porteuse et les phénomènes locaux d’une texture dynamique Υ_i .

Les textures dynamiques sont des phénomènes très complexes, elles possèdent le plus souvent plusieurs ondes porteuses de différentes amplitudes. Par soucis de simplicité, nous ne prendrons en compte que l’onde dominante.

La définition 8 formalise bien les classes de textures dynamiques suivantes :

- les structures déformables possédant un mouvement stochastique ou déterministe comme les écoulements de fluides (lac, mer, ruisseau, ...), les oscillations engendrées par le vent (l’herbe, les arbres, les drapeaux, ...), la propagation de fumée, ...
- les structures rigides possédant un mouvement déterministe comme les escalators, les ailes d’un moulin, ...
- une partie des structures rigides possédant un mouvement stochastique comme un banc de poisson, un essaim d’abeilles, ...

Cette définition s’adapte assez bien à une grande catégorie de textures dynamiques. Nous nous focalisons maintenant sur les séquences d’images les mieux adaptées à notre modèle.

L’onde porteuse \mathcal{P} Il s’agit d’un phénomène très complexe. Son support spatio-temporel dépend bien sûr de la séquence. L’onde porteuse se caractérise par sa vitesse de propagation, sa direction et son degré de stationnarité. De plus, il s’agit d’un phénomène périodique en temps et en espace.

En nous inspirant de modèles proposés en synthèse d'images [Finch 04], nous proposons la définition suivante :

Définition 9. *L'onde porteuse \mathcal{P} d'une texture dynamique Υ s'écrit comme une somme d'exponentielles complexes spatio-temporelles*

$$\mathcal{P}_i(\mathbf{x}) = \sum_{p_n \in \mathcal{P}_i} A_{p_n}(\mathbf{x}) \operatorname{Re} \left(e^{j(\boldsymbol{\omega}_{p_n} \cdot \mathbf{x} + \psi_{p_n})} \right) \quad (3.3)$$

d'amplitude $A_{p_n} \in \mathbb{R}^{+*}$, de pulsation $\boldsymbol{\omega}_{p_n} \in \mathbb{R}^3$ et de déphasage $\psi_{p_n} \in \mathbb{R}$ propres à chacune d'elles.

Ce phénomène, comme son nom l'indique, propage l'information de la texture et plus précisément les phénomènes locaux.

Les phénomènes locaux Il s'agit de phénomènes localement oscillants à la fois en espace et en temps. Le modèle prend en compte cette localisation spatio-temporelle grâce à un noyau gaussien spatio-temporel dont les paramètres prennent en compte la rapidité d'apparition/disparition des phénomènes ainsi que leur support spatial.

Définition 10. *Les phénomènes locaux \mathcal{L} d'une texture dynamique Υ se définissent par :*

$$\mathcal{L}_i(\mathbf{x}) = \sum_{\ell \in \mathcal{L}_i} \mathcal{N}_G(\mu_\ell, \Sigma_\ell)(\mathbf{x}) \sum_{\ell_k \in \mathcal{L}_i} A_{\ell_k}(\mathbf{x}) \operatorname{Re} \left(e^{j(\boldsymbol{\omega}_{\ell_k} \cdot \mathbf{x} + \psi_{\ell_k})} \right) \quad (3.4)$$

avec pour un phénomène local ℓ particulier, $\mathcal{N}_G(\mu_\ell, \Sigma_\ell)$ un noyau gaussien associé qui décrit sa localisation spatio-temporelle. $A_{\ell_k} \in \mathbb{R}^{+*}$, $\boldsymbol{\omega}_{\ell_k} \in \mathbb{R}^3$ et $\psi_{\ell_k} \in \mathbb{R}$ représentent respectivement l'amplitude, la pulsation et le déphasage associés à ce phénomène local ℓ .

Nous proposons d'évaluer notre modèle, dans un premier temps de manière qualitative, puis une étude quantitative sera proposée.

3.1.2 Évaluation du modèle de textures dynamiques proposé

L'objectif est de synthétiser une texture dynamique de type "Surface de la mer" en choisissant heuristiquement l'ensemble des paramètres du modèle.

La figure 3.3 présente le résultat d'une synthèse à l'aide de notre modèle. Sur la première ligne, on observe une texture dynamique synthétisée (c) ainsi que sa composante onde porteuse (a) et sa composante des phénomènes locaux (b). La seconde ligne présente trois textures dynamiques issues de la base DynTex.

Des similitudes sont constatées, au regard du plan (yt) pour les séquences (1.c) et (2.c). Les phénomènes locaux de ces deux textures dynamiques sont très similaires. Par contre, leur onde porteuse diffère : celle de la synthèse (1.c) se propage beaucoup plus lentement que celle de la séquence (2.c).

Ces résultats qualitatifs témoignent de la pertinence du modèle pour la représentation de certaines textures dynamiques, comme par exemple les ondes à la surface de l'eau, ou le mouvement d'un drapeau dans le vent, ...

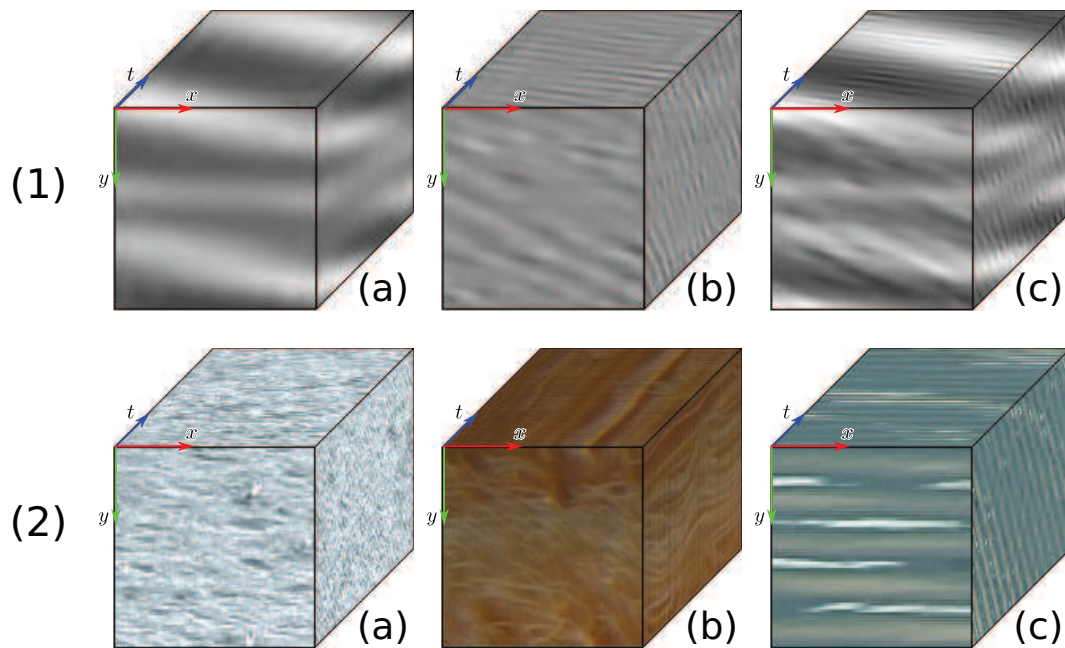


FIG. 3.3 – ♠ *Textures dynamiques synthétiques (ligne (1)) obtenues à l'aide du modèle décrit précédemment (Définition 8). (a) Vidéo représentant l'onde porteuse, (b) phénomènes localement oscillants, (c) texture dynamique composée de (a) et (b). La seconde ligne correspond à des textures dynamiques issues de DynTex (645b410, 649i810 et 64ac510).*

Toutefois, cette recherche s'est avérée longue et fastidieuse, puisqu'il faut, d'une part, paramétrer chaque exponentielle complexe de l'onde porteuse (une dizaine pour l'exemple de la figure 3.3), et d'autre part, paramétrer chaque exponentielle complexe de chaque phénomène local (environ 4 à 5 pour une dizaine de phénomènes locaux dans notre exemple).

L'approche qualitative permet de valider la pertinence du modèle. L'idéal pour évaluer le modèle est une comparaison entre une séquence d'images réelle et sa version synthétisée. Dans l'annexe B, une proposition d'approche est faite afin d'envisager une évaluation quantitative du modèle.

3.1.3 Perspectives du modèle

Dans cette section, nous proposons un modèle formalisant une certaine classe de textures dynamiques. Celui-ci repose sur une décomposition de textures dynamiques en deux composantes : une onde porteuse et une composante caractérisant les phénomènes locaux oscillants. Ce modèle a été évalué qualitativement. Une méthode fondée sur un algorithme d'optimisation par essais particuliers a été proposée afin de valider quantitativement cette approche (cf. annexe B). Toutefois, seulement des résultats partiels ont été obtenus.

Le modèle proposé, de part sa construction (cf. définition 8), permet l'ajout d'autres composantes afin de mieux traduire la complexité des phénomènes présents dans une texture dynamique.

Dans la section suivante, nous étudions plus particulièrement la composante onde porteuse du modèle. En nous appuyant sur deux méthodes de segmentation, nous montrons comment la transformée en curvelets s'avère discriminante pour l'étude de l'onde porteuse d'une texture dynamique.

3.2 Étude de l'onde porteuse

Nous nous focalisons dans cette section sur l'étude de l'onde porteuse des textures dynamiques à l'aide de la transformée en curvelets 2D+T. En effet, les propriétés de celles-ci permettent de mettre en évidence certaines particularités de l'onde porteuse.

Pour compléter et valider cette étude, deux méthodes de segmentation reposant sur les ondes porteuses sont présentées. Elles diffèrent par la manière qu'elles ont d'utiliser les coefficients de la transformée en curvelets. La première s'appuie directement sur la localisation des atomes de la transformée en curvelets pour trouver les différentes régions. La deuxième effectue un découpage spatio-temporel et analyse le contenu de celui-ci afin de déterminer les différentes régions.

3.2.1 Caractéristiques de l'onde porteuse

Dans une séquence vidéo, l'information temporelle renseigne, sur la trajectoire ainsi que sur la vitesse des différents éléments la composant.

La figure 3.4 illustre les trajectoires de trois objets évoluant à différentes vitesses (par soucis didactique, nous prenons ici une vitesse uniforme selon l'axe x). Les trois objets \mathcal{O}_1 , \mathcal{O}_2 et \mathcal{O}_3 se déplacent à une vitesse respectivement nulle, \vec{v}_1 et \vec{v}_2 avec $\|\vec{v}_2\| > \|\vec{v}_1\|$.

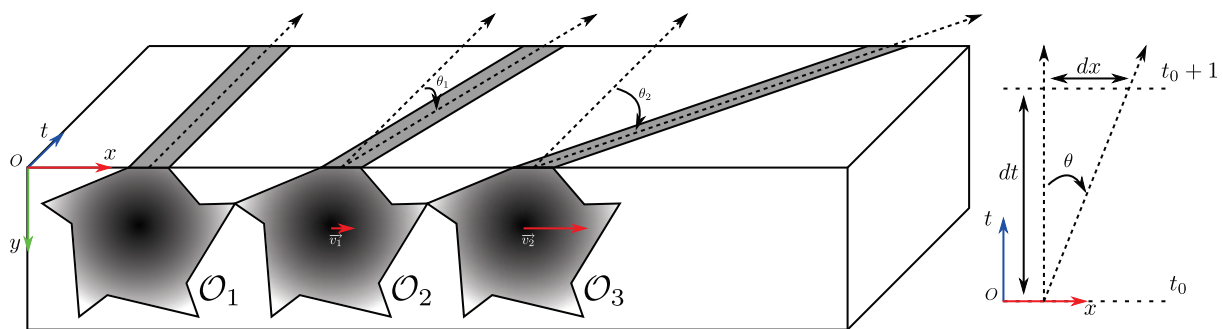


FIG. 3.4 – Illustration de trajectoires spatio-temporelles d'objets évoluant selon la direction de l'axe \vec{x} à différentes vitesses.

Le plan $(\vec{x}O\vec{t})$ permet d'observer les trajectoires des différents objets \mathcal{O}_n , $n = \{1, 2, 3\}$. Chacune d'elle possède un angle par rapport à l'axe \vec{t} proportionnel à la vitesse de l'objet observé. Plus l'angle est grand plus la vitesse de l'objet est élevée (cf. la trajectoire de l'objet

\mathcal{O}_3). Dans le cas d'un objet immobile (objet \mathcal{O}_1), cet angle est nul. Une relation directe existe entre la vitesse instantanée v_x et l'angle θ :

$$v_x = \frac{dx}{dt} = \tan \theta \quad (3.5)$$

Cet angle définit dans la suite de ce travail la direction spatio-temporelle.

Nous avons présenté ici des déplacements à vitesse constante. Ceux-ci peuvent bien sûr changer de direction, d'amplitude (accélération, décélération), ... conduisant à des formes de trajectoires plus complexes.

Le modèle de textures dynamiques proposé dans la section précédente (définition 8) montre clairement que celles-ci peuvent être composées de plusieurs ondes porteuses, chacune d'elles possède une direction de propagation propre définie par les paramètres du modèle.

La direction de propagation et la direction spatio-temporelle sont des concepts liés. La figure 3.5 illustre différentes ondes porteuses simples. Il s'agit d'une seule onde se propageant à différentes vitesses. Une vitesse nulle dans le cas de \mathcal{P}_1 , une vitesse faible pour \mathcal{P}_2 et plus grande pour \mathcal{P}_3 .

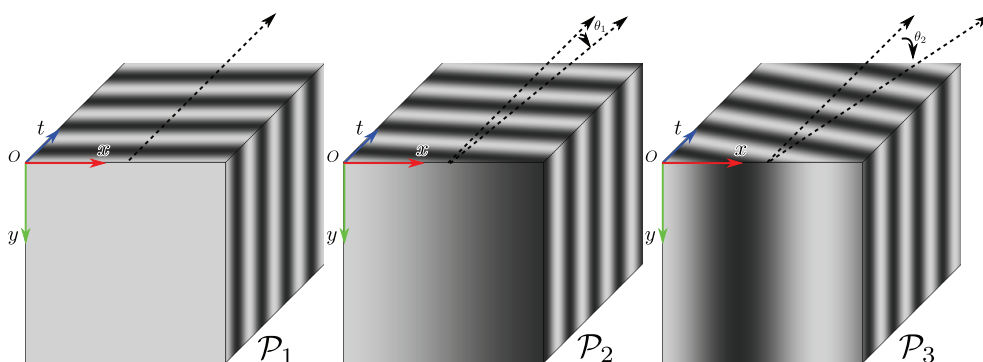


FIG. 3.5 – ♠ Illustration du lien entre les directions spatio-temporelles et les directions de propagation des ondes porteuses. La vitesse de propagation de \mathcal{P}_1 , respectivement \mathcal{P}_2 et \mathcal{P}_3 est nulle, respectivement faible et un peu plus grande.

Si l'on regarde l'axe perpendiculaire aux stries engendrées par l'onde porteuse, nous pouvons remarquer (comme dans le cas des directions spatio-temporelles) que plus l'angle entre l'axe temporel \vec{t} est grand plus la vitesse de propagation est grande. Ainsi de part leur nature (cf. définition 9), les ondes porteuses oscillent perpendiculairement à cette direction spatio-temporelle.

Une curvelet 2D+T (cf. section 2.2.3) possède un comportement oscillatoire selon une direction θ_ℓ choisie (cf. les propriétés de la section 2.2.1). Grâce à cette propriété, l'onde porteuse d'une texture dynamique sera représentée de manière creuse par une curvelet : les coefficients générés auront donc une amplitude élevée et ils porteront une énergie importante, observable notamment sur les cartes de Mercator.

La figure 3.6 montre la projection de Mercator (cf. section 2.2.3.3) de l'énergie d'une séquence d'images synthétique sur laquelle deux textures dynamiques sont présentes. Cette séquence est construite à partir de la séquence des marches d'un escalator. Quatre sous-séquences la constituent, chacune d'elles ayant une direction spatio-temporelle spécifique : seule l'information temporelle permet de faire la distinction entre les quatre régions, les textures 2D étant identiques.

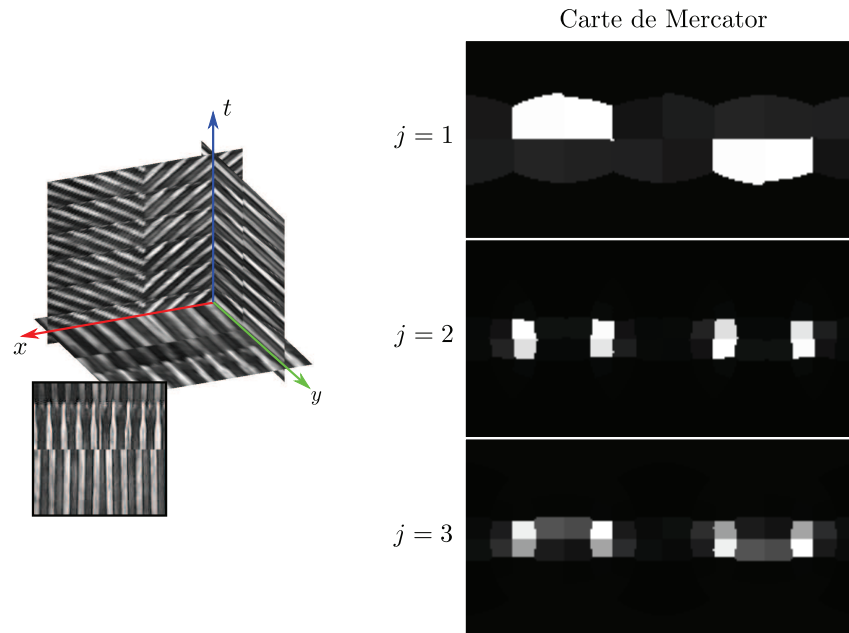


FIG. 3.6 – ♠ Projection de Mercator de l'énergie de la transformée en curvelets d'une séquence d'images de synthèse.

La projection de l'énergie de la transformée en curvelets 2D+T sur des cartes de Mercator (cf. Figure 3.6) apporte plusieurs informations sur les directions spatio-temporelles présentes dans la séquence. On peut observer sur les niveaux de décomposition $j = 2$ et $j = 3$, quatre pics d'énergie bien visibles. Ils sont représentatifs de deux directions spatio-temporelles correspondant aux deux sous-séquences. Pour une direction spatio-temporelle, nous obtenons deux pics symétriques sur les cartes de Mercator dus à la symétrie de la transformée en curvelets 2D+T.

Ainsi, une séquence où plusieurs textures dynamiques sont présentes, constituées chacune d'elles, d'une ou plusieurs ondes porteuses, fera apparaître des pics d'énergie selon différentes orientations et échelles de la transformée en curvelets. Nous proposons deux algorithmes de segmentation de séquence d'images au sens des directions spatio-temporelles prédominantes. Ces deux algorithmes diffèrent par la manière de prendre en compte l'énergie des coefficients de la transformée en curvelets 2D+T.

3.2.2 Segmentation spatio-temporelle utilisant directement la transformée en curvelets 2D+T

3.2.2.1 Schéma général

La section précédente montrait que l'énergie des coefficients de la transformée en curvelets 2D+T mettait en évidence les directions spatio-temporelles présentes dans une texture dynamique (cf. figure 3.6). La première méthode de segmentation se fonde directement sur cette observation.

La direction spatio-temporelle d'une texture dynamique donnée engendrera de nombreux coefficients importants pour l'échelle j et l'orientation ℓ en relation avec la direction et les fréquences de celle-ci. La position de ces coefficients sera estimée par le volume $c(j, \ell, \mathbf{k})$ où \mathbf{k} représente leur position. Nous proposons l'algorithme 1 pour segmenter une texture dynamique.

Algorithme 1 Segmentation de textures dynamiques au sens des directions spatio-temporelles à l'aide de l'énergie de la transformée en curvelets 2D+T

Tache : Segmentation d'une séquence de textures dynamiques

Paramètres :

- La vidéo f à segmenter
- Paramètre $\lambda \in [0, 1]$ pour la définition des directions significatives
- Seuil γ pour la définition d'un coefficient significatif
- Seuil ρ pour le seuillage du support

Boucle principale :

// Calcul de la transformée en curvelet de f

$\forall j, \forall \ell, c(j, \ell, \mathbf{k}) = \langle f, \varphi_{j, \ell, \mathbf{k}} \rangle$

// Pour chaque échelle et chaque orientation ; calcul de l'énergie

$\forall j, \forall \ell, E_{j, \ell} = \frac{1}{N_{k_1} N_{k_2} N_{k_3}} \sum_{k_1} \sum_{k_2} \sum_{k_3} |c(j, \ell, \mathbf{k})|^2$ avec $\mathbf{k} = (k_1, k_2, k_3)$

// Sélection des énergies significatives

pour chaque j et chaque ℓ faire

si $E_{j, \ell} > \lambda \cdot \max_{j, \ell} (E_{j, \ell})$ **et si** $E_{j, \ell}$ est un maximum local **alors**

// On met à 1 les coefficients significatifs et à 0 les autres

pour chaque \mathbf{k} faire

si $c(j, \ell, \mathbf{k}) > \gamma$ **alors**

$c(j, \ell, \mathbf{k}) = 1$

sinon

$c(j, \ell, \mathbf{k}) = 0$

fin si

fin pour

// Une nouvelle région \mathcal{R}_n est créée. \mathcal{S} sélectionne les pixels supérieurs à ρ

$\mathcal{R}_n = \mathcal{S}(|c(j, \ell, \mathbf{k}) \varphi_{j, \ell, \mathbf{k}}|)$

fin si

fin pour

La première étape de cet algorithme correspond au calcul de la transformée en curvelets de toute la séquence f , donnant la famille des coefficients $c(j, \ell, \mathbf{k})$. L'énergie pour chaque échelle et chaque orientation est ensuite calculée. Les maxima de ces énergies correspondent aux principales directions spatio-temporelles.

L'étape suivante sélectionne les échelles j et orientations ℓ pour lesquelles l'énergie est prédominante. Ceci s'effectue en plusieurs étapes. Tout d'abord, les énergies trop faibles sont mises à zéro. Une énergie est jugée trop faible si elle est inférieure à $\lambda \cdot \max_{j, \ell} (E_{j, \ell})$, avec $\lambda \in [0, 1]$ fixé de manière heuristique. Ensuite, l'énergie $E_{j, \ell}$ sera conservée si elle correspond à un maximum local.

La définition de maximum local dans notre cas mérite d'être détaillée. En effet, classiquement, un maximum est relatif à un voisinage. Dans notre cas, les données sont structurées de manière différente : les énergies obtenues par la transformée en curvelets sont fonction d'une échelle et d'une orientation. Ainsi, plutôt que d'effectuer une recherche de maxima locaux dans un espace à trois dimensions et d'imposer une définition de voisinage local (fonction d'un angle et d'une résolution), nous proposons l'approche suivante (illustrée sur le figure 3.7) :

- après avoir calculé les énergies sur toutes les échelles et orientations, les cartes de Mercator sont créées (cf. section 2.2.3.3). Il s'agit d'une projection conforme (conservation des angles), la recherche des maxima locaux de manière analogue à ceux d'une image est donc réalisable de façon simple pour chaque échelle j ;
- un regroupement à travers les échelles est alors effectué, qui permet d'associer des maxima locaux non connectés à une échelle donnée.

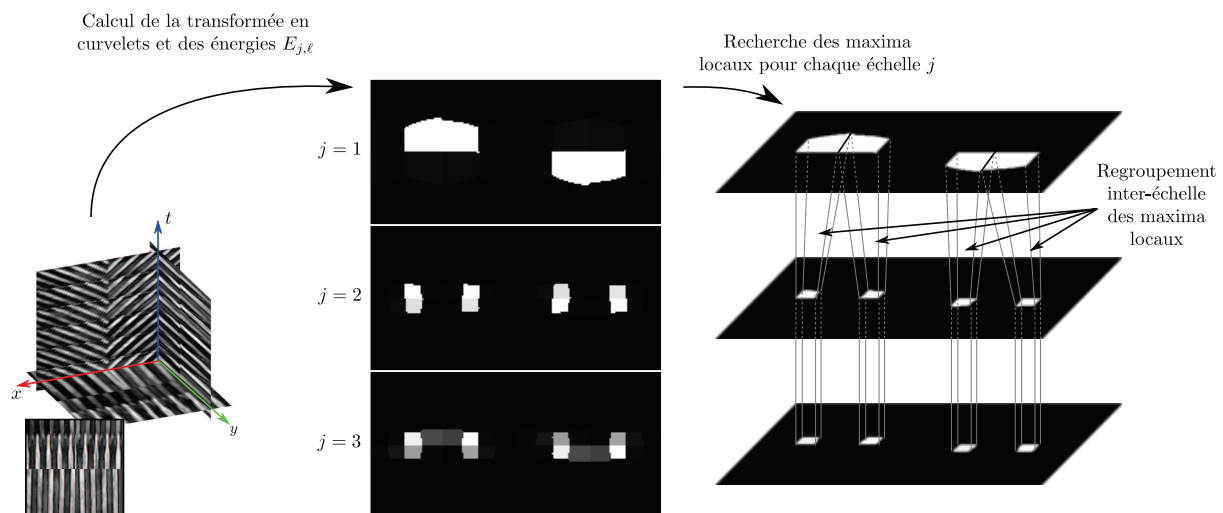


FIG. 3.7 – Illustration de la méthode permettant de rechercher et de connecter les maxima locaux des énergies de la transformée en curvelets $2D+T$.

Les maxima locaux ainsi détectés représentent les principales directions spatio-temporelles des textures dynamiques présentes dans la séquence d'images. Il s'agit de l'information que nous cherchons à isoler.

La segmentation des différentes régions s'effectue en sélectionnant tout d'abord les coefficients les plus pertinents dans la sous-bande de détails $c(j, \ell, \mathbf{k})$. Ceux-ci sont mis à 1, les autres à 0. Cette sous-bande de détails est ensuite reconstruite, pour être ensuite seuillée par le paramètre ρ permettant de sélectionner le support de chaque atome (illustration en 2D sur la figure 3.8). On trouve ainsi les régions correspondantes.

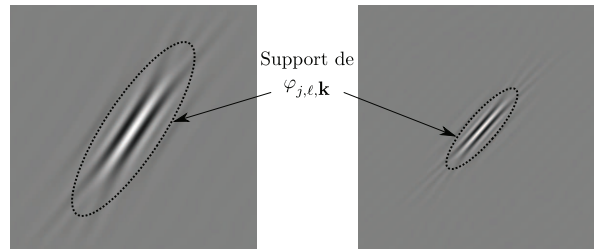


FIG. 3.8 – Illustration du support d'un atome de curvelet dans le cadre 2D.

Nous présentons dans la section suivante quelques uns des résultats obtenus.

3.2.2.2 Resultats et discussion

L'algorithme 1 a été testé sur des séquences synthétiques. Celle présentée sur la figure 3.9.(a) est composée de deux textures dynamiques de DynTex. Elle correspond à la juxtaposition des séquences d'un lac et d'un escalator. Les vidéos (b) et (c) représentées sur la figure 3.9 montrent les régions obtenues à l'aide de l'algorithme. Les zones noires correspondent aux voxels hors région.

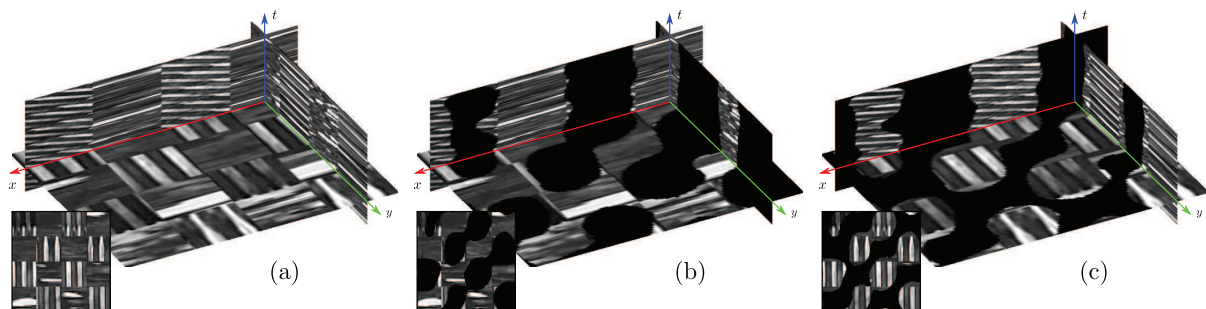


FIG. 3.9 – ♠ Les deux régions, (b) et (c), résultant de la segmentation de la vidéo de synthèse (a) à l'aide des atomes de la transformée en curvelets 2D+T.

Nous pouvons observer que les deux régions sont assez bien détectées. En effet, l'observation de chacune des régions permet d'isoler la texture dynamique qu'elle contient. On peut cependant noter une forte imprécision aux frontières des régions. Celle-ci est bien sûr liée au support de l'atome de curvelet.

3.2.2.3 Perspectives

La méthode précédente, effectuant une segmentation des textures dynamiques à l'aide de la transformée en curvelets 2D+T, possède certaines limites. La principale étant son imprécision sur les frontières des régions des textures dynamiques. Afin d'améliorer ceci, différentes pistes sont envisageables :

- plutôt que de sélectionner le support de l'atome, on peut localiser le centre (x, y, t) de l'atome dans la séquence. En effectuant cette opération pour l'ensemble des coefficients sélectionnés, un nuage de points est obtenu. A chaque point est associé un maximum local de l'énergie de la transformée en curvelet au sens où nous l'avons défini précédemment. Chaque point est représentatif d'une région. On utilise ensuite l'algorithme des k plus proches voisins (sur un critère de distance) afin de classer les voxels de la séquence selon les différentes régions.
- le défaut de cette méthode vient principalement du fait que la taille du support de la curvelet devient de plus en plus importante à mesure que les échelles diminuent. Il faut donc chercher à mieux localiser les événements. Plutôt que d'effectuer une transformée en curvelets sur toute la séquence, on peut décider de regarder un sous-cube de la vidéo afin d'en étudier le contenu. On en déduira les directions spatio-temporelles présentes dans ce sous-cube. La prise en compte des informations de tous les sous-cubes permettra une segmentation plus précise des régions.

3.2.3 Segmentation spatio-temporelle à l'aide d'une structure en octree et de la transformée en curvelets 2D+T

3.2.3.1 Schema général

La section précédente a permis de mettre en évidence le fait que les énergies les plus importantes de la transformée en curvelets permettent de détecter différentes directions spatio-temporelles. Celles-ci sont ensuite localisées pour segmenter la séquence au sens des textures dynamiques. Les frontières des différentes régions ne sont toutefois pas détectées précisément à cause de l'incertitude liée au support de l'atome de la curvelet.

Le principe de la seconde approche repose sur une analyse plus locale de la séquence à l'aide d'une structure en octree¹.

Nous proposons un découpage de la séquence en un ensemble de sous-cubes. Ces volumes étant de taille réduite, ils conduiront à une meilleure localisation des frontières.

La dimension inférieure du sous-cube est cependant un paramètre important. En effet, pour que le calcul de l'énergie de la transformée en curvelets 2D+T s'avère pertinent, il est nécessaire que cette dimension ne soit pas trop petite. Autrement dit, un nombre conséquent de coefficients doit être présent à une échelle et une orientation données afin que le calcul de l'énergie ait un sens.

Le tableau 3.1 donne la taille du plus petit sous-cube de coefficients en fonction du nombre d'échelles et d'orientations choisies par rapport à la taille de la séquence.

¹Extension à la 3D du concept du quadtree [Finkel 74], il a été proposé par Gervautz *et al.* [Gervautz 90] afin de quantifier les images couleurs

Taille Vidéo	128×128×128					
Nombre d'échelles	3		4		5	
Nombre de sub-divisions	2	4	2	4	2	4
Nombre de coefficients	79507	64715	9261	8712	1331	1089
Taille Vidéo	64×64×64					
Nombre d'échelles	3		4		5	
Nombre de sub-divisions	2	4	2	4	2	4
Nombre de coefficients	9261	8712	1331	1089	125	125
Taille Vidéo	32×32×32					
Nombre d'échelles	3		4		5	
Nombre de sub-divisions	2	4	2	4	2	4
Nombre de coefficients	1331	1331	125	125	27	27
Taille Vidéo	16×16×16					
Nombre d'échelles	3		4		5	
Nombre de sub-divisions	2	4	2	4	2	4
Nombre de coefficients	125	125	27	27	1	1
Taille Vidéo	8×8×8					
Nombre d'échelles	3		4		5	
Nombre de sub-divisions	2	4	2	4	2	4
Nombre de coefficients	27	27	1	1	1	1

TAB. 3.1 – Tableaux indiquant la taille du plus petit sous-cube de coefficients en fonction du nombre d'échelles et d'orientations choisit dans la transformée en curvelet 2D+T selon la taille de la séquence.

Algorithme 2 Construction d'une structure en octree pour une vidéo

Tache : Création de la structure en octree d'une vidéo selon un critère d'homogénéité.

Paramètres :

- Le cube vidéo $\mathcal{C}_{x,y,t}(t_x, t_y, t_t)$ de dimension (t_x, t_y, t_t) et d'origine (x, y, t)
- Taille minimale d'un sous-cube t_{min}

Fonction octree($\mathcal{C}_{x,y,t}(t_x, t_y, t_t)$)

tant que $t_x > t_{min}$ **et** $t_y > t_{min}$ **et** $t_t > t_{min}$ **faire**

si $\mathcal{C}_{x,y,t}(t_x, t_y, t_t)$ n'est pas homogène **alors**

- ▶ octree($\mathcal{C}_{x,y,t}(t_x/2, t_y/2, t_t/2)$)
- ▶ octree($\mathcal{C}_{x+t_x/2,y,t}(t_x/2, t_y/2, t_t/2)$)
- ▶ octree($\mathcal{C}_{x,y+t_y/2,t}(t_x/2, t_y/2, t_t/2)$)
- ▶ octree($\mathcal{C}_{x+t_x/2,y+t_y/2,t}(t_x/2, t_y/2, t_t/2)$)
- ▶ octree($\mathcal{C}_{x,y,t+t_t/2}(t_x/2, t_y/2, t_t/2)$)
- ▶ octree($\mathcal{C}_{x+t_x/2,y,t+t_t/2}(t_x/2, t_y/2, t_t/2)$)
- ▶ octree($\mathcal{C}_{x,y+t_y/2,t+t_t/2}(t_x/2, t_y/2, t_t/2)$)
- ▶ octree($\mathcal{C}_{x+t_x/2,y+t_y/2,t+t_t/2}(t_x/2, t_y/2, t_t/2)$)

sinon

- ▶ Arrêter le découpage du cube.

fin si

fin tant que

Ce tableau montre que si l'on prend un sous-cube vidéo de dimensions $16 \times 16 \times 16$ voxels ou $8 \times 8 \times 8$ voxels, la plus petite structure de coefficients de la transformée en curvelets 2D+T contient uniquement 1 coefficient pour certains paramètres d'échelles.

Autrement dit, à la lecture du tableau, si par exemple le nombre d'échelles est fixé à $j = 5$, les sous-cubes vidéos de dimensions inférieures à $32 \times 32 \times 32$ ne contiennent pas assez de coefficients pour que le calcul de l'énergie soit pertinent.

Décomposer la séquence entièrement à l'aide de sous-cubes et calculer sur chacun d'entre eux la transformée en curvelets 2D+T prend un temps non négligeable. Il est donc préférable de découper en sous-cubes uniquement les régions se trouvant sur les frontières (les régions étant définies au sens d'une même direction spatio-temporelle).

La précision est nécessaire uniquement pour les frontières des régions spatio-temporelles. Autrement dit, il n'est pas nécessaire de découper plus finement une zone ayant une seule direction spatio-temporelle significative. C'est pourquoi une structure en octree est construite sur la séquence : celle-ci permet de partitionner l'espace tridimensionnel en le subdivisant récursivement en huit sous-espaces tant que celui-ci n'est pas homogène (cf. figure 3.10 et algorithme 2).

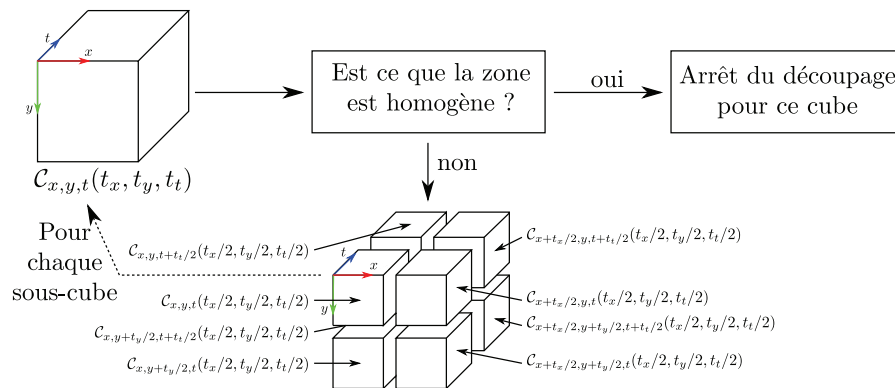


FIG. 3.10 – Principe général du découpage à l'aide de la structure en octree.

La structure en octree permet l'obtention d'un arbre représentant la décomposition. Celui-ci est constitué d'un ensemble de noeuds et de feuilles. Un noeud représente une subdivision d'une région de la séquence en 8 sous-cubes, et une feuille définit une région homogène en terme de direction spatio-temporelle. Afin d'établir une segmentation, il faut parcourir l'arbre de décomposition et comptabiliser les différentes directions spatio-temporelles.

La structure en octree s'avère donc être un moyen de subdiviser une séquence d'images en zones spatio-temporelles dites homogènes. La manière dont cette subdivision est réalisée dépend totalement du critère d'homogénéité. Dans la section suivante, nous proposons deux critères adaptés aux textures dynamiques.

3.2.3.2 Discussion sur les critères d'homogénéité

Les structures en quadtree ou en octree sont généralement utilisées dans un objectif de quantification [Gervautz 90]. Pour cela, la variance de la luminance est souvent prise en compte dans le critère d'homogénéité. Lorsqu'une zone possède une variance faible, elle est considérée comme homogène.

Dans le cadre des textures dynamiques, le critère d'homogénéité doit être construit à partir des caractéristiques propres au phénomène étudié. Suite aux résultats obtenus et présentés dans la section 3.2.2, il est naturel de bâtir notre critère d'homogénéité sur le nombre de directions spatio-temporelles significatives composant notre cube vidéo. Ainsi, si plusieurs directions sont détectées, le bloc vidéo ne sera pas considéré comme homogène, il sera donc redécoupé.

Ceci nous conduit à la définition :

Définition 11. *Une région est dite homogène au sens des textures dynamiques, si elle est dominée par une seule direction spatio-temporelle.*

Le premier critère d'homogénéité mis en place consiste donc, pour chaque cube, à effectuer la transformée en curvelets 2D+T puis à calculer l'énergie des sous-bandes de coefficients ainsi obtenus. Les maxima locaux d'énergies des sous-bandes sont ensuite extraits. Ils ne sont cependant pas utilisés ici pour reconstruire les supports des curvelets associés, mais sont tout simplement comptabilisés.

Ce critère se définit de la manière suivante :

Définition 12. *Un cube vidéo est considéré comme homogène si un seul maximum local est détecté dans les énergies des sous-bandes de coefficients de la transformée en curvelets 2D+T.*

L'algorithme 3 détermine si un cube est homogène au sens des directions spatio-temporelles d'une texture dynamique.

L'implémentation de ce critère d'homogénéité pour la construction de la structure en octree a conduit à des résultats intéressants, mais souffrant d'un certain nombre de problèmes.

Certains phénomènes ne sont pas détectés conduisant à une segmentation très grossière. En effet, l'énergie des sous-bandes de coefficients de la transformée en curvelets 2D+T et le volume spatio-temporelle du phénomène dans le cube vidéo étudié sont étroitement liés.

Nous avons constaté, pour le seuil λ choisi, qu'une direction spatio-temporelle est détectée seulement si elle est présente sur un volume d'environ 40% par rapport à la taille du cube d'expertise. Ce n'est malheureusement pas toujours le cas.

Pour que le critère d'homogénéité puisse détecter ces directions spatio-temporelles, le seuil λ doit être plus petit, ce qui conduit à accroître le nombre de directions détectées et ainsi le nombre de fausses détections.

Pour remédier à ce problème, nous proposons une autre solution fondée sur un second critère plus adapté.

Algorithme 3 Premier critère d'homogénéité pour la segmentation de textures dynamiques à l'aide d'une structure en octree

Tache : Permet de déterminer si le cube vidéo $\mathcal{C}_{x,y,t}(t_x, t_y, t_t)$ est homogène au sens des directions spatio-temporelles selon la définition 12.

Paramètres :

- Le cube vidéo $\mathcal{C}_{x,y,t}(t_x, t_y, t_t)$ de taille (t_x, t_y, t_t) et d'origine (x, y, t)
- Seuil $\lambda \in [0, 1]$

Boucle principale :

// Calcul de la transformée en curvelets du cube vidéo $\mathcal{C}_{x,y,t}(t_x, t_y, t_t)$

$\forall j, \forall \ell, c(j, \ell, \mathbf{k}) = \langle \mathcal{C}_{x,y,t}(t_x, t_y, t_t), \varphi_{j,\ell,\mathbf{k}} \rangle$

// Pour chaque échelle et chaque orientation - Calcul de l'énergie

$\forall j, \forall \ell, E_{j,\ell} = \frac{1}{N_{k_1} N_{k_2} N_{k_3}} \sum_{k_1} \sum_{k_2} \sum_{k_3} |c(j, \ell, \mathbf{k})|^2$ avec $\mathbf{k} = (k_1, k_2, k_3)$

// Nombre de directions trouvées

pour chaque j et chaque ℓ **faire**

si $E_{j,\ell} > \lambda \cdot \max_{j,\ell} (E_{j,\ell})$ **et si** $E_{j,\ell}$ est un maximum local **alors**

// Une nouvelle direction spatio-temporelle est détectée

nbDirection = nbDirection + 1

fin si

fin pour

// Le cube est t'il homogène ?

si nbDirection = 1 **alors**

▶ Le cube est homogène

sinon

▶ Le cube n'est pas homogène

fin si

Le second critère d'homogénéité proposé s'appuie sur une étude plus détaillée du contenu du sous-cube. A chaque itération et pour un cube donné, les huit sous-cubes le composant sont étudiés. On redéfinit donc le critère d'homogénéité comme suit :

Définition 13. *Un cube vidéo est considéré comme homogène, si les huit sous-cubes le composant possèdent le même maximum local dans les énergies des sous-bandes de coefficients de la transformée en curvelets 2D+T.*

Autrement dit, si les huit sous-cubes sont homogènes au sens de la direction et de l'échelle des ondes porteuses des textures dynamiques, et s'ils possèdent tous la même direction spatio-temporelle, alors le cube est considéré comme homogène.

L'étude des huit sous-cubes permet de déterminer plus finement le contenu du cube original. Autrement dit, une direction spatio-temporelle présente dans 40% du volume d'un des sous-cubes (ce qui représente 5% du volume général du cube) pénalisera l'homogénéité globale du cube. Toutefois, contrairement au premier critère, le seuil de l'énergie est inchangé.

Ce nouveau critère d'homogénéité est pris en compte dans l'algorithme 4.

Algorithme 4 Deuxième critère d'homogénéité dans le cadre de segmentation de textures dynamiques à l'aide d'une structure en octree

Tache : Permet de savoir si le cube vidéo $\mathcal{C}_{x,y,t}(t_x, t_y, t_t)$ est homogène au sens des directions spatio-temporelles selon la définition 13.

Paramètres :

- Le cube vidéo $\mathcal{C}_{x,y,t}(t_x, t_y, t_t)$
- Seuil $\lambda \in [0, 1]$

Boucle principale :

// Calcul de la transformée en curvelets sur les huit sous-cubes vidéo composant le cube vidéo $\mathcal{C}_{x,y,t}(t_x, t_y, t_t)$

$$\forall j, \forall \ell, c^1(j, \ell, \mathbf{k}) = \langle \mathcal{C}_{x,y,t}(t_x/2, t_y/2, t_t/2), \varphi_{j,\ell,\mathbf{k}} \rangle$$

$$\forall j, \forall \ell, c^2(j, \ell, \mathbf{k}) = \langle \mathcal{C}_{x+t_x/2,y,t}(t_x/2, t_y/2, t_t/2), \varphi_{j,\ell,\mathbf{k}} \rangle$$

$$\forall j, \forall \ell, c^3(j, \ell, \mathbf{k}) = \langle \mathcal{C}_{x,y+t_y/2,t}(t_x/2, t_y/2, t_t/2), \varphi_{j,\ell,\mathbf{k}} \rangle$$

$$\forall j, \forall \ell, c^4(j, \ell, \mathbf{k}) = \langle \mathcal{C}_{x+t_x/2,y+t_y/2,t}(t_x/2, t_y/2, t_t/2), \varphi_{j,\ell,\mathbf{k}} \rangle$$

$$\forall j, \forall \ell, c^5(j, \ell, \mathbf{k}) = \langle \mathcal{C}_{x,y,t+t_t/2}(t_x/2, t_y/2, t_t/2), \varphi_{j,\ell,\mathbf{k}} \rangle$$

$$\forall j, \forall \ell, c^6(j, \ell, \mathbf{k}) = \langle \mathcal{C}_{x+t_x/2,y,t+t_t/2}(t_x/2, t_y/2, t_t/2), \varphi_{j,\ell,\mathbf{k}} \rangle$$

$$\forall j, \forall \ell, c^7(j, \ell, \mathbf{k}) = \langle \mathcal{C}_{x,y+t_y/2,t+t_t/2}(t_x/2, t_y/2, t_t/2), \varphi_{j,\ell,\mathbf{k}} \rangle$$

$$\forall j, \forall \ell, c^8(j, \ell, \mathbf{k}) = \langle \mathcal{C}_{x+t_x/2,y+t_y/2,t+t_t/2}(t_x/2, t_y/2, t_t/2), \varphi_{j,\ell,\mathbf{k}} \rangle$$

// Pour chaque sous-cube vidéo, chaque échelle et chaque orientation - Calcul de l'énergie

$$\forall n \in \llbracket 1, 8 \rrbracket, \forall j, \forall \ell, E_{j,\ell}^n = \frac{1}{N_{k_1} N_{k_2} N_{k_3}} \sum_{k_1} \sum_{k_2} \sum_{k_3} |c^n(j, \ell, \mathbf{k})|^2$$

// Nombre de directions présentes

pour chaque $n \in \llbracket 1, 8 \rrbracket$ **et** chaque j **et** chaque ℓ **faire**

si $E_{j,\ell}^n > \lambda \cdot \max_{j,\ell} (E_{j,\ell}^n)$ **et** **si** $E_{j,\ell}^n$ est un maximum local de tous les sous-cubes **alors**

// Une nouvelle direction spatio-temporelle est détectée

nbDirection = nbDirection + 1

fin si

fin pour

// Le cube est t'il homogène ?

si nbDirection = 1 **alors**

▶ Le cube est homogène

sinon

▶ Le cube n'est pas homogène

fin si

L'implémentation de ce critère d'homogénéité donne de bien meilleurs résultats que celui de la définition 12. Les phénomènes faiblement présents dans un cube vidéo sont maintenant détectés, ce qui conduit à une meilleure segmentation due à un découpage plus précis.

Une autre difficulté apparaît cependant. Compte-tenu du choix de pondération que nous avons souhaité identique sur chacun des 3 axes (cf. section 2.2.3.2), une oscillation perpendiculaire à une orientation dans le plan spatial est traitée de la même manière qu'une orientation dans le plan temporel.

Ainsi certains phénomènes spatiaux prennent autant, voire plus d'importance que les ondes

porteuses des textures dynamiques pendant la phase de segmentation. Ces dernières étant de nature spatio-temporelles, une fonction de normalisation des énergies a été mise en place pour pénaliser les orientations purement spatiales.

Cette pondération se construit à partir de l'orientation θ_ℓ de l'atome de curvelet 2D+T. L'intersection de cet angle avec la sphère unitaire donne un point de coordonnées $(\omega_1, \omega_2, \omega_3)$ utilisé pour le calcul du coefficient de pondération c_{θ_ℓ} . Celui-ci est indépendant de ω_1 et ω_2 et se définit de la façon suivante :

$$\forall \omega_3 \in [0, 1], \forall \kappa \in [0, 1], \forall \gamma > 0, \quad c_{\theta_\ell}(\omega_3) = \sqrt{1 - \omega_3^2} - \kappa e^{-\gamma \omega_3^2} \quad (3.6)$$

avec κ le coefficient pondérateur d'une direction purement spatiale et γ la rapidité de transition.

Le comportement de cette fonction de pondération est illustrée sur la figure 3.11.

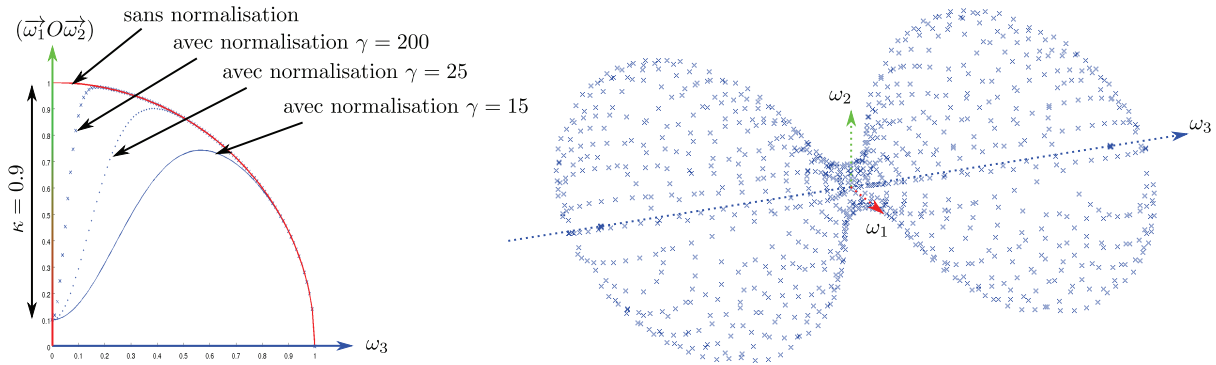


FIG. 3.11 – Fonction de normalisation des énergies de la transformée en curvelets 2D+T définie pour pénaliser les orientations purement spatiales. A gauche un profil 2D, et à droite une vue 3D.

La pondération des énergies, pour chaque échelle j et orientation ℓ de la transformée en curvelets 2D+T permet de mieux prendre en compte l'information temporelle et pénalise l'information spatiale. La structure de l'octree ainsi obtenue est plus proche des données étudiées et des phénomènes recherchés. Ainsi une texture statique est pénalisée par rapport à une texture dynamique.

3.2.3.3 Résultats

L'approche décrite précédemment a été validée dans un premier temps sur des séquences de synthèse et dans un second temps sur des séquences de la base DynTex.

Séquences de synthèse

Afin de tester la structure en octree utilisant l'information de l'énergie de la transformée en curvelets 2D+T comme critère d'homogénéité, des premiers tests ont été effectués sur des séquences de synthèse particulières. Les séquences sont représentées sur la partie gauche de la figure 3.12. Les séquences sont construites à partir d'une séquence "escalator" (54pd110 de DynTex). Elles diffèrent de la manière dont elles sont construites.

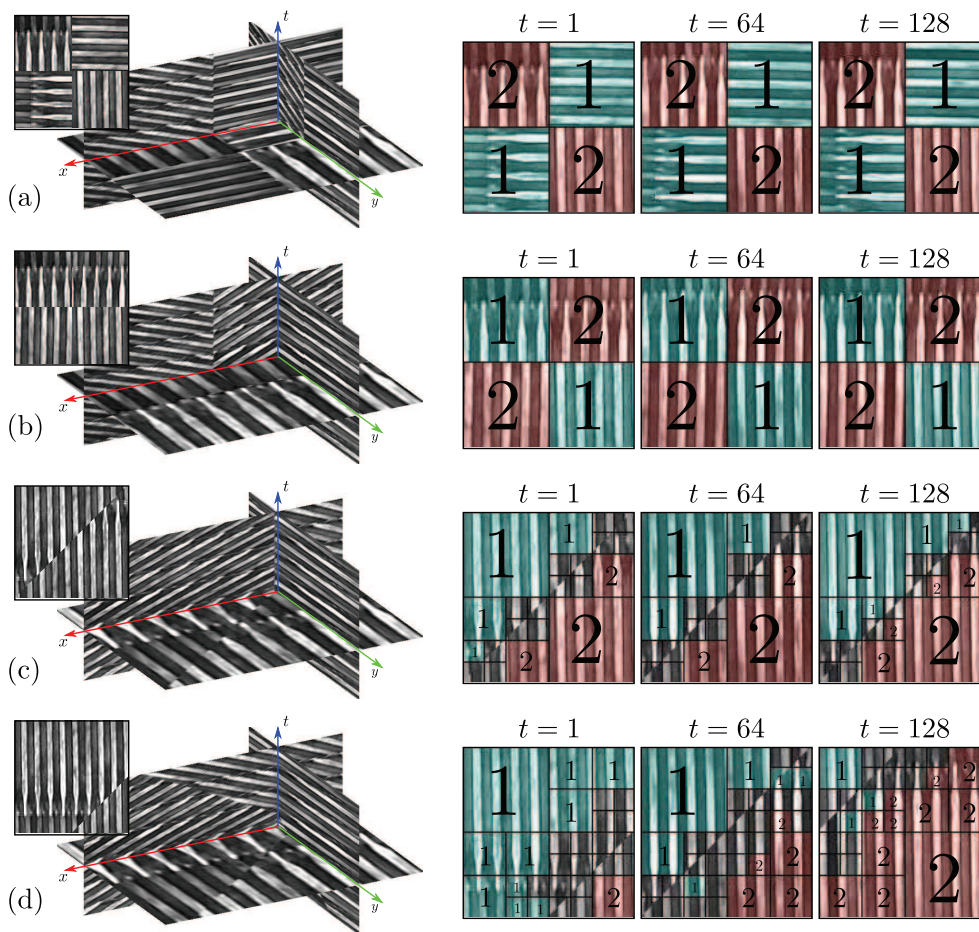


FIG. 3.12 – ♠ Résultats de la segmentation en régions spatio-temporelles, en utilisant une structure en octree avec l'énergie de la transformée en curvelets $2D+T$ comme critère d'homogénéité. A gauche, les images de synthèse originales, et à droite le résultat de la segmentation pour chacune d'elles. Chaque couleur (bleue et rouge respectivement labélisées par 1 et 2) représente une région différente. Une région non colorée correspond à une zone d'ambiguïté. Les traits noirs représentent les frontières des sous-cubes de la structure en octree.

Pour la séquence de synthèse (a), les deux régions sont différentes à la fois spatialement (angle de 90 degrés entre les deux motifs) et temporellement (directions spatio-temporelles différentes). Par contre, pour la séquence de synthèse (b), seul le temps permet la distinction entre les deux régions : elles sont, spatialement, placées en miroir l'une de l'autre, les directions spatio-temporelles sont donc différentes. Dans ces deux premières séquences synthétisées, les frontières entre les régions sont rectilignes (dans le sens des axes x et y) et immobiles au cours du temps.

La séquence de synthèse (c) est similaire à la précédente (b) (les deux régions sont placées en miroir l'une de l'autre), la frontière entre les deux suit ici une diagonale. Cela représente donc un cas très défavorable pour la segmentation utilisant une structure de type octree qui privilégie les structures orientées horizontalement et verticalement.

La dernière séquence de synthèse (d) diffère par sa frontière évoluant au cours du temps entre les deux régions : elle progresse du coin en bas à droite vers le coin en haut à gauche du cube vidéo. Ainsi une des régions grandit spatialement alors que l'autre diminue. Ici aussi, il s'agit d'un cas défavorable pour une structure en octree.

Le résultat de la segmentation des différentes vidéos de synthèse est présenté dans la partie droite de la figure 3.12. Les frontières de la structure en octree sont représentées par des traits noirs. Les différentes régions spatio-temporelles sont identifiées à la fois par un jeu de couleurs et de labels (couleur bleu et rouge respectivement labélisée par 1 et 2).

Dans les cas (a) et (b), la segmentation distingue parfaitement les différentes régions spatio-temporelles présentes dans les vidéos. Cette segmentation parfaite est due à deux facteurs :

- les frontières sont horizontales et verticales et donc parfaitement adaptées à la structure en octree.
- la fonction de normalisation mise en place influence la segmentation de la séquence (b), ce qui n'était pas le cas auparavant.

Dans le cas de la séquence (c), la méthode de segmentation détecte correctement les deux régions (celle en haut à gauche et celle en bas à droite). La frontière est finement découpée par la structure en octree. Ceci peut s'observer par la taille, la plus petite possible, des sous-cubes autour de la diagonale de la vidéo. Bien que la frontière soit détectée, la méthode de segmentation n'arrive pas à identifier les différentes régions (non présence de couleurs et de labels), car l'information contenue dans les sous-cubes n'est pas assez représentative. Ainsi, afin d'éviter d'avoir une erreur importante, nous préférons définir ces régions comme ambiguës.

Le dernier cas de synthèse s'avère être plus complexe concernant la frontière des deux régions. En effet, celle-ci se déplaçant au cours du temps, elle traverse et donc influe sur la segmentation de nombreux sous-cubes de la structure en octree. Ceci rend donc la détection locale de la texture plus complexe. On peut noter que de nombreux cubes sont classés comme ambigus tout au long de la frontière spatio-temporelle. Les deux régions principales sont cependant détectées correctement.

Séquences de la base DynTex

Nous présentons maintenant des résultats de notre méthode de segmentation sur des séquences issues de DynTex.

La première séquence est composée d'une seule texture dynamique comportant un objet la perturbant. Il s'agit de la vidéo (645c310) de la surface d'un lac sur laquelle un canard se déplace à contre-courant. La partie (a) de la figure 3.13 montre la première image de cette vidéo ainsi que les principales directions spatio-temporelles que l'on peut observer et que nous symbolisons ici par des flèches.

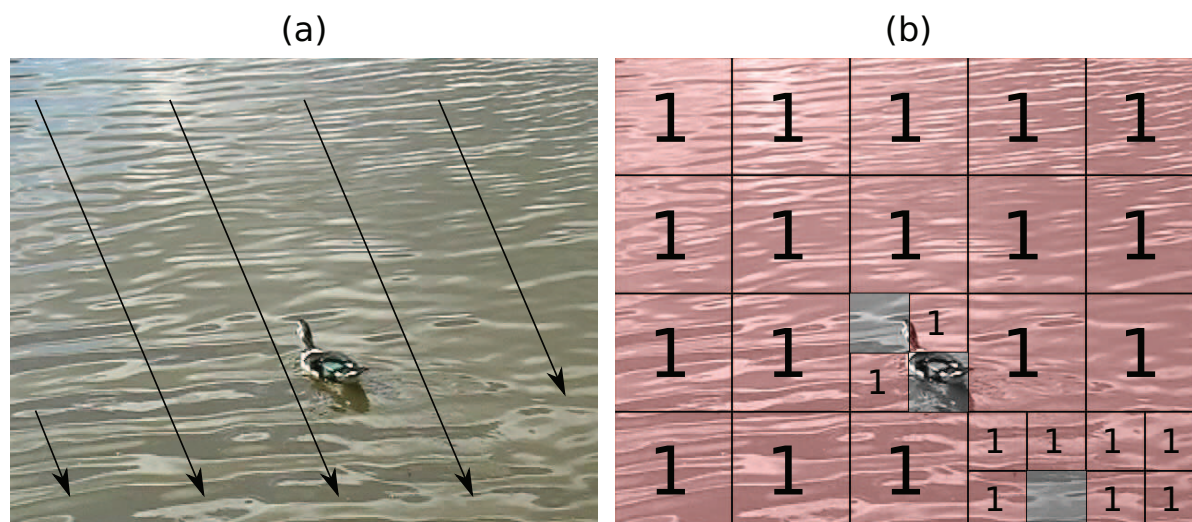


FIG. 3.13 – ♠ (a) Séquence originale 645c310. Les principales directions spatio-temporelles sont symbolisées par des flèches. (b) Résultat de la segmentation de la vidéo. La couleur rouge, labellisée par le nombre 1, représente la région détectée. Une région dépourvue de couleur correspond à une région considérée comme ambiguë. Les traits noirs représentent les frontières des sous-cubes de la structure en octree.

Le résultat de la segmentation, présentée par la figure 3.13.(b), montre la détection d’une seule région spatio-temporelle, ce qui est en accord avec les observations.

De plus, autour du canard, deux sous-cubes de la séquence sont identifiés comme régions ambiguës. Une de ces zones est directement localisée sur le canard et s’explique tous simplement par la présence de celui-ci. La deuxième pourrait faire penser à une fausse détection. Elle s’explique cependant par la future présence du canard dans cette région, la segmentation s’appuyant en effet sur l’information spatiale et temporelle de toute la séquence (le canard se dirige en effet dans cette direction). La structure en octree, grâce à la transformée en curvelets, analyse temporellement l’information, et intègre l’ensemble de la trajectoire du canard dans les différents sous-cubes visités.

Un sous-découpage, ainsi qu’une région ambiguë, peuvent être observés dans l’angle en bas à droite de la séquence. La trajectoire amont du canard coïncide avec cette partie de la vidéo. La surface de l’eau a donc été perturbée. Ce qui est détecté par les coefficients de la transformée en curvelets, et explique le sous-découpage et la mauvaise segmentation.

La deuxième séquence issue de DynTex (645b310) présentée sur la figure 3.14.(a) comporte cette fois deux textures dynamiques différentes.

La première est celle d’une rivière s’écoulant en direction de la caméra. Comme le montre les flèches symbolisant les directions spatio-temporelles, cette rivière s’écoule au milieu de la séquence de la droite vers la gauche, puis du bas, vers la caméra. En plus de ce phénomène, une autre texture dynamique est visible en arrière plan : un arbre oscillant de droite à gauche à cause du vent.

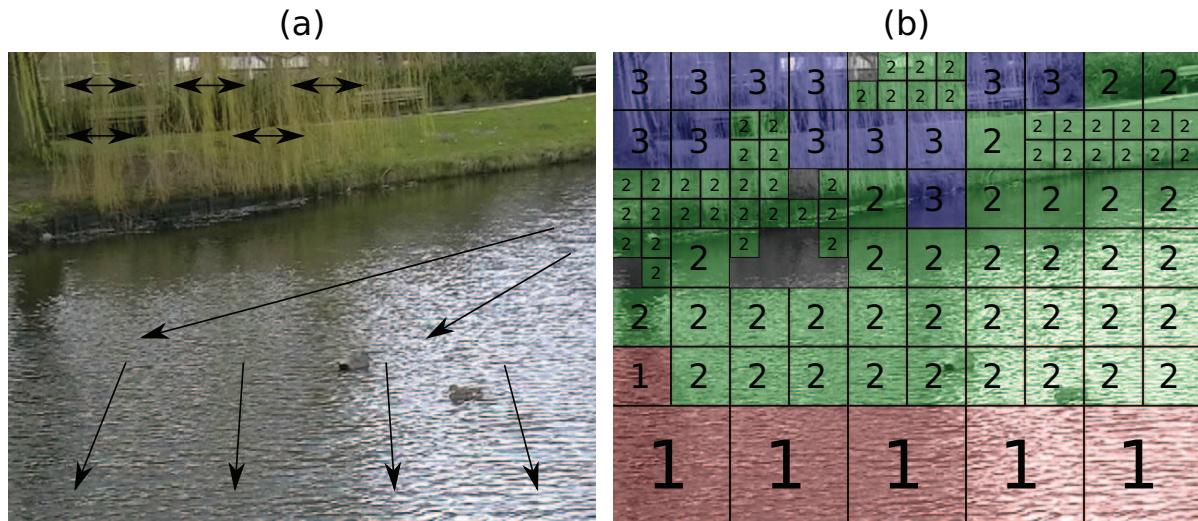


FIG. 3.14 – ♠ (a) Séquence originale 645b310. Les principales directions spatio-temporelles sont symbolisées ici par des flèches. (b) Résultat de la segmentation de la vidéo. Chaque couleur (rouge, verte et bleu, respectivement labelisée par les nombres 1, 2 et 3) représente une région différente. Une région dépourvue de couleur correspond à une région considérée comme ambiguë. Les traits noirs représentent les frontières des sous-cubes de la structure en octree.

La figure 3.14.(b) montre les différentes régions spatio-temporelles obtenues. La rivière qui est composée de deux directions spatio-temporelles différentes est bien segmentée en deux régions distinctes. La première (en rouge), au premier plan, représente la texture dynamique de la rivière se propageant vers le bas de l'image. La deuxième (en vert), au centre de la séquence, représente la même texture dynamique mais qui se propage cette fois-ci de droite à gauche. Nous observons également que la taille des sous-cubes à l'avant plan est plus grande que celle en arrière plan. Ceci s'explique par les effets de perspectives induit par la caméra.

Dans la région représentée en vert, une partie du rivage est présente, alors qu'il s'agit d'une partie statique de la séquence. Ce problème s'observe au moment de la fusion des différentes feuilles de l'arbre de la structure en octree. En effet, le critère de fusion repose uniquement sur les directions réellement spatio-temporelles et non purement spatiale. Or une zone statique n'est pas constituée, par définition, d'ondes porteuses.

Pour éviter ce problème, il faudrait effectuer une fusion des différentes feuilles en s'appuyant sur des critères supplémentaires (couleurs, informations spatiales, ...).

La texture dynamique détectée dans la région bleue présente des particularités complexes à analyser : mouvement oscillant superposé à un fond présentant, lui, une texture statique. L'algorithme, même dans ce cas difficile, segmente correctement la région où est présente la texture dynamique.

La dernière séquence présentée est celle d'une cascade présentant un arbre au premier plan (6482210). Cette séquence complexe, visible sur la partie gauche de la figure 3.15, est composée de plusieurs textures dynamiques. Celles-ci sont soit semi-transparentes (cas de la cascade) soit discrètes et peu denses pour laisser apparaître le fond (cas de la branche d'arbre).



FIG. 3.15 – ♠ (a) Séquence originale 6482210. Les principales directions spatio-temporelles sont symbolisées ici par des flèches. (b) Résultat de la segmentation de la vidéo. Chaque couleur (rouge, verte et bleu, respectivement labelisée par les nombres 1, 2 et 3) représente une région différente. Une région dépourvue de couleur correspond à une région considérée comme ambiguë. Les traits noirs représentent les frontières des sous-cubes de la structure en octree.

Bien que cette séquence soit relativement complexe, l'algorithme de segmentation arrive à différencier les deux textures dynamiques.

La cascade est présente dans la région verte et l'arbre dans la région rouge. Toutefois, nous pouvons observer plusieurs erreurs de segmentation autour des frontières entre les deux textures dynamiques. Cette région est une zone d'ambiguïté, et est en effet difficile à analyser puisque les deux textures dynamiques sont présentes. La conséquence est que le sous-cube est soit classé dans la région cascade (verte) soit classé dans la région arbre (rouge).

3.2.3.4 Perspectives concernant la structure en octree

Ce chapitre étudie l'utilisation de la transformée en curvelet 2D+T pour l'analyse des séquences d'images. Une méthode pour segmenter spatialement et temporellement des séquences vidéos de textures dynamiques est présentée. Celle-ci a été appliquée avec succès sur des cas de synthèse et des vidéos réelles issues de DynTex.

Des travaux complémentaires peuvent être envisagés pour améliorer le critère d'homogénéité afin d'augmenter la précision de découpage de l'octree. D'autre part, la fusion des différentes régions lorsque l'on parcourt l'arbre de l'octree peut être améliorée. En effet, pour l'instant le regroupement est uniquement effectué sur les directions spatio-temporelles, mais il est possible de regarder d'autres critères pour lever certaines ambiguïtés.

La segmentation obtenue peut servir à différencier les différentes textures dynamiques présentes dans une séquence. Ainsi pour chaque région, on peut par la suite extraire un vecteur caractéristique dans un objectif d'indexation de textures dynamiques.

Conclusion du chapitre Ce chapitre formalise tout d'abord un modèle de textures dynamiques. Celui-ci s'applique à une classe de textures dynamiques particulière (écoulement de fluide, drapeau, ...) et se définit comme la somme d'une ou plusieurs ondes porteuses et de phénomènes locaux. Les synthèses réalisées à l'aide de ce modèle témoignent de la pertinence de celui-ci. Une méthode s'appuyant sur un algorithme d'optimisation par essais particuliers a été proposée pour valider ce modèle quantitativement.

Pour la classe de textures dynamiques considérée, le problème de l'analyse se traduit sous la forme d'une décomposition de celle-ci en phénomènes locaux oscillants et en ondes porteuses. Ainsi dans la deuxième partie de ce chapitre, nous présentons une étude d'une des composantes du modèle : l'onde porteuse. L'utilisation de la transformée en curvelets 2D+T s'avère pertinente pour la caractérisation de celle-ci. En effet, nous avons proposée deux méthodes de segmentation au sens des textures dynamiques utilisant la transformée en curvelets 2D+T comme critère.

Dans le prochain chapitre, nous cherchons à décomposer les textures dynamiques selon le modèle que nous venons de présenter. Nous montrerons comment l'approche de l'analyse en composantes morphologiques permet de répondre à ce problème.

CHAPITRE 4

Décomposition de textures dynamiques

Décomposition de textures dynamiques

Sommaire

4.1	Problème de décomposition d’images	118
4.1.1	Discussion	118
4.1.2	L’analyse en composantes morphologiques (MCA)	119
4.1.2.1	Cadre théorique	120
4.1.2.2	Implémentation	121
4.1.2.3	Stratégies de seuillage	123
4.2	Application aux textures dynamiques	126
4.2.1	Choix du dictionnaire	126
4.2.2	Nouvelles stratégies de seuillage	128
4.2.2.1	Théorie	128
4.2.2.2	Évaluation	130
4.2.3	Expérimentations	136
4.2.3.1	Paramètres utilisés	136
4.2.3.2	Résultats	137

Le chapitre précédent propose un modèle formel d’une classe particulière de textures dynamiques. Comme le montre l’étude effectuée dans la section 3.1.2, la recherche des paramètres de ce modèle s’avère complexe et plutôt coûteuse en temps de calcul, même pour des vidéos de dimension réduite. Pour des séquences d’images de dimensions plus importantes (environ $720 \times 576 \times 128$ voxels, voire plus pour la dimension temporelle), la méthode qui a permis une première validation du modèle, est bien trop coûteuse en temps de calcul pour être utilisée.

Les approches de décomposition d’images de la littérature [Chan 01, Aujol 05b, Starck 05] semblent tout à fait pertinentes pour extraire des composantes de manière plus globale, sur une séquence d’images vérifiant les propriétés du modèle proposé.

Ainsi dans une première partie, après une discussion sur la problématique générale de la décomposition dans le domaine du traitement du signal, suivi de la justification du choix de l’utilisation de l’analyse en composantes morphologiques, nous présentons, dans notre contexte d’étude, le cadre théorique de cette approche, ainsi que son implémentation et les stratégies de seuillage existantes.

Dans la seconde partie, nous nous focalisons sur son application aux textures dynamiques. Nous proposons un dictionnaire adapté à ces dernières. Pour cela, nous étudions les bases de la littérature les plus représentatives des composantes de la texture dynamique que nous avons

identifiées.

Nous mettons ensuite l'accent sur les contraintes de temps de calcul des algorithmes de la littérature appliqués aux séquences d'images. Pour des images, l'algorithme de décomposition n'est pas coûteux en temps. L'ajout de la dimension temporelle change cependant considérablement les temps de convergence de l'algorithme. Afin de lever ce verrou, nous proposons de nouvelles stratégies de seuillage. Celles-ci seront étudiées les unes par rapport aux autres et comparées avec celles de la littérature. Finalement, des résultats de décomposition sur des séquences d'images de notre base seront présentés et interprétés.

4.1 Problème de décomposition d'images

4.1.1 Discussion

Décomposer un signal en différentes composantes a toujours représenté un problème majeur en traitement d'images. Nous pouvons citer, par exemple, la recherche des contributions des instruments de musique dans un morceau musical, la séparation des voix dans un enregistrement audio, l'extraction de la partie géométrique et de la partie texturée d'une image, ...

Plus formellement, l'objectif est de retrouver la matrice représentant les différentes sources S , qui mixées par la matrice A et corrompues par un bruit N , donnent l'observation X .

Ceci s'écrit :

$$X = A \cdot S + N \quad (4.1)$$

Il s'agit d'un problème inverse généralement mal posé. Il est possible de le résoudre en supposant par exemple, la stationnarité des signaux, la parcimonie, ...

Dans la littérature, pour répondre à ce type de problématique, plusieurs familles de méthodes existent.

Modèle de décomposition de Meyer Cette approche a été développée principalement pour séparer une image en deux composantes : une partie cartoon et une partie texturée. Ce type d'approche a connu un regain d'intérêt depuis les travaux d'Yves Meyer [Meyer 01]. La partie cartoon (généralement notée u) est supposée appartenir à l'espace BV^1 . La partie texturée peut être modélisée dans différents espaces : l'espace G de Meyer [Meyer 01], les espaces de Sobolev négatifs [Aujol 05b] ou certains espaces de Hilbert adaptés aux textures localement oscillantes [Aujol 06, Starck 04, Starck 05].

Afin de contrôler la variation totale et donc la décomposition, différents critères variationnels ont été proposés dans la littérature [Aujol 05a, Vese 02, El-Hamidi 10].

Séparation aveugle de sources La problématique de la séparation de sources consiste à estimer les signaux originaux émis par les sources à partir des mélanges observés. Si les poids des mélanges sont inconnus, on parle de séparation aveugle de sources (BSS) : les poids doivent

¹ $BV(\omega)$ est le sous-espace des fonction $u \in L^1(\omega)$ de telle sorte que la quantité suivante, appelée variation totale de u , est finie :

$$J(u) = \sup \left\{ \int_{\Omega} u(x) \operatorname{div}(\xi(x)) dx \right\}$$

tel que $\xi \in C_c^1(\Omega, \mathbb{R}^2)$, $\|\xi\|_{L^\infty(\Omega)} \leq 1$

être estimés conjointement aux sources. Une littérature très abondante, recensée par exemple dans [Bobin 08], existe sur ce domaine.

Décomposition de Wold Cette approche permet de décomposer efficacement les images texturées [Francos 93]. Une image est décomposée en deux parties : un champ déterministe et un champ stochastique. L'approche est utilisée notamment pour reconstruire des informations portés par une dimension à partir d'autres dimensions.

4.1.2 L'analyse en composantes morphologiques (MCA)

La séparation aveugle de sources peut s'appliquer si l'on est assuré de la séparabilité des sources. Il est fondamental de disposer de mesures de diversité ou de contraste entre elles. Le modèle proposé dans la définition 3.2 de la section 3.1, reposant sur la combinaison de plusieurs processus distincts, nous assure de cette séparabilité.

Parmi les méthodes de séparation aveugle de sources, deux d'entre elles ont retenues notre attention : l'analyse en composantes indépendantes (ICA) et l'analyse en composantes morphologiques (MCA).

La figure 4.1 illustre sur des images deux exemples caractéristiques de l'application de l'analyse en composantes morphologique

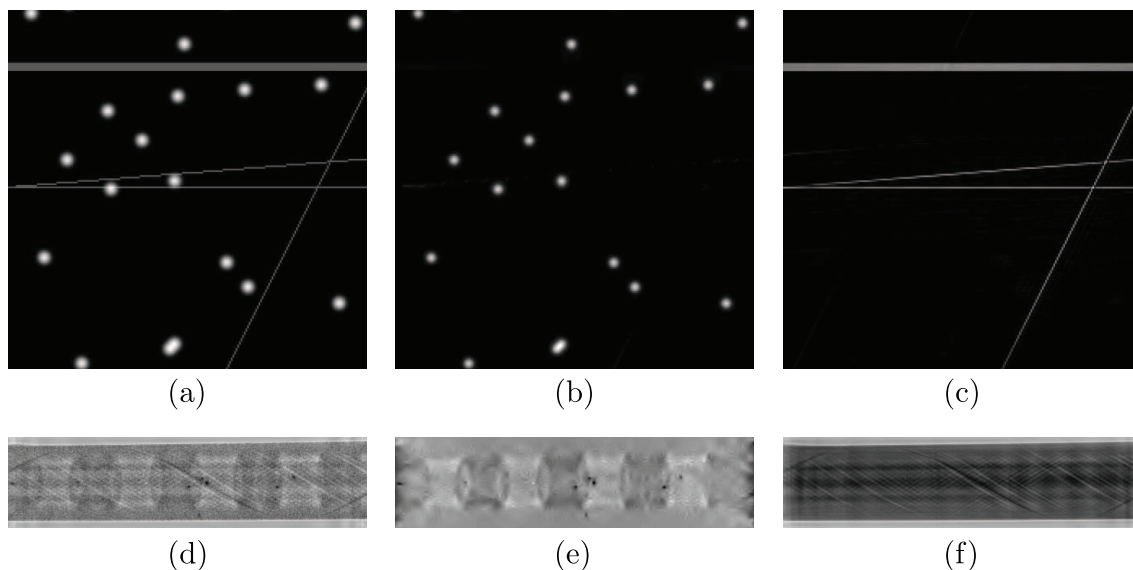


FIG. 4.1 – Exemples de décomposition en composantes morphologiques issus de [Starck 04, Fadili 09b]. (a) L'observation, (b) ses structures isotropiques (extraites à l'aide de la transformée par ondelettes à trous) et (c) ses lignes (projetées sur une base des ridgelets). (d) L'image originale, (e) ses structures isotropiques et son fond (extraites par la transformée en ondelettes) et (f) ses courbes (projetées sur une base de curvelets).

On observe qu'avec des bases adaptées (transformée en ondelettes à trous et en ridgelets dans le cas du premier exemple et transformée en ondelettes et en curvelets dans le deuxième) l'analyse en composantes morphologiques permet d'extraire les différentes structures caractéristiques de l'image originale (les structures isotropiques et les lignes dans le cas du premier exemple et les structures isotropiques et le fond dans le second).

L'analyse en composantes indépendantes reste très sensible à la présence de bruit dans les données observées [Davies 04, Koldovsky 06]. Or les conditions d'acquisition des vidéos, l'encodage, la présence de phénomènes non représentés dans le modèle, sont des sources importantes de bruit. Nous avons donc choisi d'utiliser pour notre étude l'analyse en composantes morphologiques.

Nous présentons dans la première partie de ce chapitre le cadre général de l'analyse en composantes morphologiques, puis dans une seconde partie, les modifications nécessaires à apporter pour son adaptation aux séquences d'images et plus particulièrement au traitement des textures dynamiques.

4.1.2.1 Cadre théorique

L'hypothèse de départ de l'analyse en composantes morphologiques (MCA) est d'écrire un signal observé y comme une superposition linéaire de N composantes morphologiques perturbées par un bruit ε , $\varepsilon \sim \mathcal{N}(0, \sigma^2)$:

$$y = \sum_{i=1}^N y_i + \varepsilon \quad (4.2)$$

où chaque y_i est une composante morphologique. y_i est compressible dans la base Φ_i avec les opérateurs d'analyse \mathbf{T}_i et de synthèse \mathbf{T}_i^T associés. Ainsi on peut écrire :

$$\forall i = 1, \dots, N, \quad y_i = \mathbf{T}_i^T \alpha_i \quad \text{et} \quad \alpha_i = \mathbf{T}_i y_i \quad (4.3)$$

avec α_i les coefficients de y_i projetés dans la base Φ_i .

L'approche MCA permet donc de trouver une solution acceptable au problème inverse de la décomposition dans des bases, et d'extraire les composantes $\{y_i\}_{i=1, \dots, N}$ à partir de l'observation dégradée y selon un critère de parcimonie que nous détaillerons ci-après.

Il s'agit là d'une hypothèse fondamentale : construire un dictionnaire $\Phi = [\Phi_1, \Phi_2, \dots, \Phi_N]$ tel que chaque composante y_i est représentée de manière creuse dans la base associée Φ_i . Autrement dit, y_i est bien représenté dans Φ_i et n'est pas, ou alors très peu représenté dans Φ_j , ($j \neq i$). Ceci se traduit par :

$$\forall i, j \neq i \quad \|\mathbf{T}_i y_i\|_0 < \|\mathbf{T}_j y_i\|_0 \quad (4.4)$$

où $\|\mathbf{X}\|_0$ représente la pseudo-norme ℓ_0 de la matrice \mathbf{X} (c'est à dire le nombre de coefficients non nuls). Plus la norme ℓ_0 est petite, plus la base est creuse pour le signal étudié. Par exemple, les droites de l'image (a) de la figure 4.1 sont représentées de manière creuse dans la base des ridgelets car on a besoin de peu de coefficients. Ce n'est pas le cas dans la base des ondelettes. Il va sans dire que les disques sont très mal décrits par les ridgelets ; le caractère isotropique des cercles est cependant mieux pris en considération par la transformée en ondelettes à trous.

Le choix des bases de décomposition est donc primordial. Chaque transformée possède ses propres caractéristiques et est mieux adaptée à l'extraction de phénomènes particuliers. L'article de Starck *et al.* [Starck 04] évoque quelques transformées, et leurs avantages, pouvant être utilisées dans un dictionnaire Φ . Nous discuterons du choix du dictionnaire dans la section 4.2.

Résoudre le problème précédent permet de trouver une solution au système : $y = \Phi\alpha$. Starck *et al.* proposent dans [Starck 04] et [Starck 05] de résoudre celui-ci, et de trouver les composantes morphologiques $\{y_i\}_{i=1,\dots,N}$ en résolvant le problème d'optimisation suivant :

$$\min_{y_1,\dots,y_N} \sum_{i=1}^N \|\mathbf{T}_i y_i\|_p^p \quad \text{tel que} \quad \left\| y - \sum_{i=1}^N y_i \right\|_2 \leq \sigma_\varepsilon \quad (4.5)$$

où le terme $\|\mathbf{T}_i y_i\|_p^p$ pénalise la non-parcimonie (l'évaluation de la parcimonie la plus intéressante est pour $0 \leq p \leq 1$). σ_ε représente l'écart-type du bruit ε .

Ce problème d'optimisation (4.5) est difficile à résoudre, particulièrement pour $p < 1$.

Dans le cas de $p \geq 1$, des schémas de décomposition tels que le forward-backward de Combettes *et al.* [Combettes 05] et celui de Douglas-Rachford [Lions 79, Combettes 04, Combettes 07] sont possibles.

Dans le cas où $0 \leq p \leq 1$, Starck *et al.*, dans [Starck 04], proposent un algorithme hybride et heuristique reposant sur une relaxation par blocs pour résoudre le problème d'optimisation (4.5). L'idée est que, si toutes les composantes y_j sauf la i -ème sont fixées à l'itération $k-1$, alors il est prouvé que la solution $\alpha_i^{(k)}$ est donnée par un seuillage dur des coefficients du résidu marginal projeté dans la base Φ_i :

$$\alpha_i^{(k)} = \delta_{\lambda^{(k)}} \left(\mathbf{T}_i \left(r_i^{(k)} \right) \right) \quad \text{avec} \quad r_i^{(k)} = y - \sum_{j \neq i} y_j^{(k-1)} \quad (4.6)$$

avec $\delta_{\lambda^{(k)}}$ l'opérateur de seuillage pour le seuil $\lambda^{(k)}$. Ces résidus marginaux r_i sont, par construction, susceptibles de contenir les informations marquantes de y_i (puisque ces dernières ne sont pas correctement représentées sur les bases Φ_j avec $j \neq i$).

Cette idée est à la base de l'algorithme de l'analyse en composantes morphologiques.

4.1.2.2 Implémentation

L'algorithme 5 donne les opérations principales de l'analyse en N composantes morphologiques d'un signal nD y observé.

Cet algorithme itère tant que le résidu total $\left\| y - \sum_{j=1}^N \tilde{y}_j^{(k-1)} \right\|_2$ n'est pas négligeable ($\tilde{y}_j^{(k-1)}$ étant l'estimation de la composante y_j à l'itération $(k-1)$).

A chaque itération k , pour chaque base Φ_i , l'algorithme détermine les coefficients les plus représentatifs $\alpha_i^{(k)}$ et calcule la nouvelle estimation \tilde{y}_i à partir de ceux-ci. Le seuil $\lambda^{(k)}$, après chaque itération, évolue selon la stratégie de seuillage définie.

L'algorithme de l'analyse en composantes morphologiques étant un algorithme hybride de seuillage itératif sur des bases, les deux points les plus importants de celui-ci sont le dictionnaire de base et la stratégie de seuillage. Nous présentons dans la section suivante les stratégies de seuillage de la littérature.

Algorithme 5 Analyse en composantes morphologiques d'un signal nD

Tache : Décomposition d'un signal nD sur un dictionnaire Φ .

Paramètres :

- Le signal y à décomposer
- Le dictionnaire $\Phi = [\Phi_1, \dots, \Phi_N]$
- La stratégie de seuillage appelée **stratégie**
- La condition d'arrêt σ_ε

Initialisation :

// Mise à zéro des composantes à estimer

pour $i = 1$ à N **faire**

$$\tilde{y}_i^{(0)} = 0$$

fin pour

// Initialisation de λ

$\lambda^{(1)} = \text{initialisation_lambda}(\text{stratégie})$

// Initialisation du nombre d'itérations

$k = 1$

Boucle principale :

tant que $\left\| y - \sum_{j=1}^N \tilde{y}_j^{(k-1)} \right\|_2 \leq \sigma_\varepsilon$ **faire**

// Pour chaque composante

pour $i = 1$ à N **faire**

// Calcul du résidu marginal

$$\tilde{r}_i^{(k)} = y - \sum_{j \neq i} \tilde{y}_j^{(k-1)}$$

// Projection du résidu marginal dans la base Φ_i

$$\tilde{\alpha}_i^{(k)} = \mathbf{T}_i \left(\tilde{r}_i^{(k)} \right)$$

// Seuillage dur de $\tilde{\alpha}_i^{(k)}$

$$\alpha_i^{(k)} = \delta_{\lambda^{(k)}} \tilde{\alpha}_i^{(k)}$$

// Nouvelle estimation de \tilde{y}_i

$$\tilde{y}_i^{(k)} = \mathbf{T}_i^T \left(\alpha_i^{(k)} \right)$$

fin pour

// Mise à jour du seuil λ

$\lambda^{(k+1)} = \text{mise_à_jour}(\lambda^{(k)}, \text{stratégie})$

// Progression d'une itération

$k = k + 1$

fin tant que

4.1.2.3 Stratégies de seuillage

Plusieurs stratégies de seuillage existent dans la littérature. Nous les présentons de manière formelle, puis nous discuterons de leurs avantages et inconvénients.

Un point commun de ces stratégies est l'étape d'initialisation. La valeur de $\lambda^{(1)}$ est égale au coefficient le plus fort du signal d'origine y projeté sur l'ensemble des bases du dictionnaire. Il s'écrit :

$$\lambda^{(1)} = \max_i \|\mathbf{T}_i y\|_\infty \quad (4.7)$$

Il s'agit de la meilleure valeur que l'on puisse attribuer à $\lambda^{(1)}$: donner une valeur plus grande ne modifiera pas le résultat de l'algorithme de décomposition. En effet, au dessus de cette valeur, aucun coefficient n'existe, donc il n'y a aucune information à projeter sur les bases.

Attribuer une valeur à $\lambda^{(1)}$ plus petite que $\max_i \|\mathbf{T}_i y\|_\infty$ peut par contre se révéler pénalisant pour le résultat de la décomposition. En effet, entre ces deux valeurs, de nombreux atomes appartenant éventuellement à différentes bases peuvent exister. Ainsi lors de la première itération de l'algorithme, ils seront repartis directement dans la base Φ_1 , ce qui n'est peut être pas le choix le plus judicieux.

Dans [Starck 04, Starck 05], deux comportements du seuil $\lambda^{(k)}$ sont proposées pour l'algorithme MCA : une décroissance linéaire ou exponentielle.

Stratégie de seuillage linéaire (SSL) Dans cette approche, à chaque étape, une constante δ_{SSL} définit la décroissance de λ au cours des itérations de l'algorithme. Celle-ci est donc fixée à l'initialisation de l'algorithme et est retirée à chaque étape à la valeur précédente du seuil. Cette stratégie se formalise de la manière suivante :

$$\lambda^{(k+1)} = \lambda^{(k)} - \delta_{SSL} \quad \text{avec} \quad \delta_{SSL} = \frac{\lambda^{(1)} - \lambda_{min}}{N_{iter}} \quad (4.8)$$

avec N_{iter} le nombre d'itérations de la boucle principale de l'algorithme (fixé par l'utilisateur) et λ_{min} la plus petite valeur de seuil que l'on s'autorise. Généralement, $\lambda_{min} = \tau \sigma_\varepsilon$ avec $\tau \in [3, 4]$, valeurs usuellement utilisées en débruitage d'images avec l'algorithme MCA [Fadili 10a].

Stratégie de seuillage exponentielle (SSE) Dans certains cas, la distribution des coefficients se concentre autour de l'origine. Ce phénomène peut se produire pour diverses raisons : choix non pertinent des bases conduisant par exemple à une représentation similaire dans les différentes bases, mauvaise complémentarité des bases, ...

Face à ces différentes situations, la stratégie SSL n'est pas optimale. En effet, lorsqu'on se rapproche de l'origine, la plage de coefficients sélectionnés sera disproportionnée par rapport au nombre de coefficients contenus dans la plage. Dans l'illustration de la figure 4.2, nous pouvons constater qu'environ 80% des coefficients sont présents dans le dernier intervalle. Ceux-ci vont être répartis en une seule fois ce qui conduit nécessairement à des erreurs.

C'est pourquoi, dans [Starck 04, Starck 05], les auteurs utilisent également une stratégie à décroissance exponentielle (SSE), permettant de seuiller de grandes plages de coefficients dans les premières itérations de l'algorithme, et de plus petites plages sur les dernières itérations (cf figure 4.2). Cette stratégie conduit à une meilleure répartition des coefficients sur chaque plage

lorsque ceux-ci sont proches de l'origine mais, tout comme la SSL, le nombre d'itérations doit être fixé par l'utilisateur.

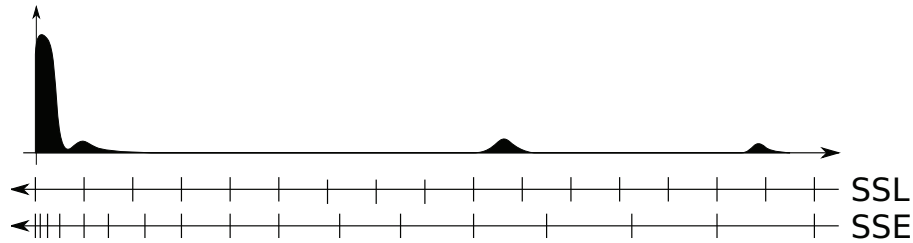


FIG. 4.2 – Représentation schématique de plages déterminées selon les stratégies SSL et SSE sur une distribution d'illustration.

Ainsi, la pente δ_{SSE} de la courbe de décroissance va diminuer au fur et à mesure des itérations de l'algorithme :

$$\lambda^{(k+1)} = \lambda^{(k)} * \delta_{SSE} \text{ avec } \delta_{SSE} = \left(\lambda^{(1)} - \lambda_{min} \right)^{-\frac{1}{N_{iter} - 1}} \quad (4.9)$$

avec N_{iter} le nombre d'itérations de la boucle principale de l'algorithme (fixé par l'utilisateur) et λ_{min} le seuil minimum.

Dans les deux stratégies que nous venons de voir, N_{iter} doit être déterminé initialement, ce qui est à la fois un avantage et un inconvénient. Fixer le nombre d'itérations permet de s'assurer de l'arrêt de l'algorithme. En effet, celui-ci sera forcé de s'arrêter après N_{iter} itérations, même si la solution n'est pas optimale, une décomposition aura été obtenue. Fixer ce nombre d'itérations a priori n'est cependant pas évident.

La figure 4.3 illustre l'évolution de SSL et SSE pour $N_{iter} = 25$ et $N_{iter} = 100$.

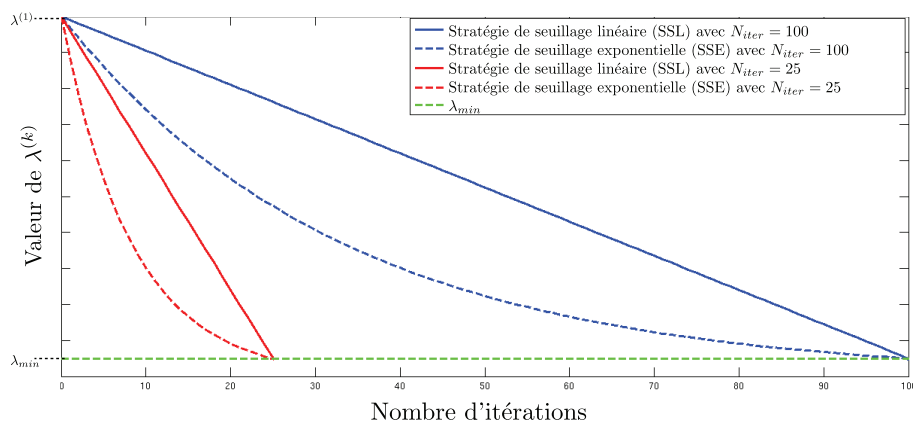


FIG. 4.3 – Évolution de $\lambda^{(k)}$ pour les stratégies SSL et SSE en fonction du nombre d'itérations k pour différentes valeurs de N_{iter}

On remarque dans les deux cas, que la plage de coefficients sélectionnés n'est pas la même selon la stratégie et selon la valeur de N_{iter} . La répartition des différents atomes dans les bases du dictionnaire Φ ne sera donc pas identique. Autrement dit, l'estimation des composantes morphologiques y_i peut grandement varier selon la rapidité de la décroissance de la pente de $\lambda^{(k)}$.

De plus, le paramètre N_{iter} est d'autant plus difficile à fixer qu'il dépend des caractéristiques du signal y observé. En effet, pour certaines images, une dizaine (signal simple, signal très creux pour les bases du dictionnaire Φ , ...) comme une centaine (image très complexe, ...) d'itérations peuvent être nécessaires.

Ces deux stratégies possèdent également un autre inconvénient. Elles ne sont pas adaptatives, autrement dit, on n'est pas certain d'avoir à chaque itération des coefficients à répartir dans les bases du dictionnaire.

Stratégie de seuillage *Mean of Max* (SSMoM) Afin d'obtenir un comportement adaptatif et pour pallier le problème d'initialisation de N_{iter} , Bobin *et al.*, dans [Bobin 07], proposent une autre stratégie.

L'idée sous-jacente est qu'à l'itération k , les prochains atomes à sélectionner sont ceux de la base Φ_i , où le résidu courant $y - \sum_{j=1}^N \tilde{y}_j^{(k-1)}$ possède les plus grandes corrélations.

Comme l'illustre la figure 4.4, à la fin de l'itération k , on calcule le résidu $r^{(k)}$:

$$r^{(k)} = y - \sum_{i=1}^N \tilde{y}_i^{(k)} \quad (4.10)$$

puis on effectue l'analyse de celui-ci sur l'ensemble des bases du dictionnaire.

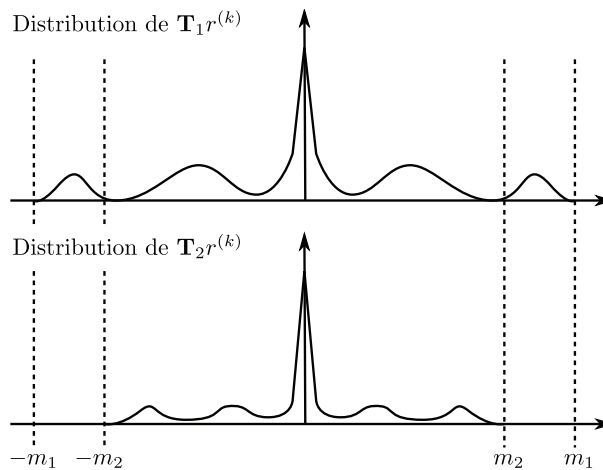


FIG. 4.4 – Illustration, dans le cadre de deux bases, du principe de la stratégie de seuillage *Mean of Max*.

On recherche ensuite, parmi l'ensemble des atomes de toutes les bases, les deux premiers maxima. Ceux-ci s'écrivent :

$$\begin{aligned} m_1 &= \max_i \|\mathbf{T}_i r^{(k)}\|_\infty \\ m_2 &= \max_{j, j \neq i_0} \|\mathbf{T}_j r^{(k)}\|_\infty \quad \text{avec } i_0 = \operatorname{argmax}_i \|\mathbf{T}_i r^{(k)}\|_\infty \end{aligned} \quad (4.11)$$

Le seuil de l'itération suivante sera décidé en effectuant la moyenne entre les deux maxima m_1 et m_2 . Ainsi :

$$\lambda^{(k+1)} = \frac{1}{2} (m_1 + m_2) \quad (4.12)$$

Cette stratégie a l'avantage de ne pas avoir de paramètre de décroissance à fixer et d'être adaptative (dépendante des caractéristiques du signal y observé). Par contre, contrairement aux deux stratégies originelles SSL et SSE, une projection supplémentaire à chaque itération sur les bases du dictionnaire est nécessaire.

Dans cette première partie, nous avons étudié l'analyse en composantes morphologiques (MCA), de sa théorie à sa mise en oeuvre. Deux points cruciaux de cette approche ont été abordés : le choix des bases constituant le dictionnaire ainsi que la stratégie de seuillage employée.

Dans la section suivante, nous proposons d'appliquer cette approche de décomposition sur des séquences d'images présentant des textures dynamiques. Après une discussion sur le choix du dictionnaire, nous proposons de nouvelles stratégies de seuillage afin d'améliorer les temps de calcul de l'algorithme. En effet, ceux-ci augmentent considérablement à cause de la dimension temporelle supplémentaire.

4.2 Application aux textures dynamiques

Bien que la théorie MCA soit totalement indépendante de la dimension d'étude, à notre connaissance, très peu de travaux [Woiselle 09, Backer 10] abordent l'analyse en composantes morphologiques dans le cadre des vidéos. L'extension de l'algorithme à la dimension temporelle impose d'adapter les bases du dictionnaire.

Dans cette deuxième partie, en nous appuyant sur le modèle de textures dynamiques que nous avons proposé précédemment, nous discutons des bases retenues pour la décomposition.

Les temps de calcul augmentant de manière importante sur les séquences d'images de textures dynamiques, nous proposons de nouvelles stratégies de seuillage afin de les réduire. Ces stratégies sont ensuite évaluées et comparées à celles de la littérature selon plusieurs critères.

Enfin, des résultats de décomposition sur des textures dynamiques de la base DynTex sont exposés et commentés.

4.2.1 Choix du dictionnaire

Un des points cruciaux dans l'approche MCA est la définition du dictionnaire. Un choix non adapté des transformations au regard de la dynamique des phénomènes présents dans la séquence est préjudiciable quant à la qualité du résultat : décomposition non pertinente, pseudo-norme ℓ_0 importante, coefficients non représentatifs.

Dans la section 3.1 nous avons formalisé une classe de textures dynamiques comme la composition de deux phénomènes distincts : une onde porteuse \mathcal{P} et des phénomènes locaux \mathcal{L} . Il est donc nécessaire d'associer à chacun d'eux la base la plus représentative afin d'effectuer une décomposition correcte.

Base pour l'onde porteuse \mathcal{P} Dans la section 3.2, nous avons étudié l'onde porteuse des textures dynamiques. Nous avons montré à l'aide de plusieurs méthodes de segmentation que la transformée en curvelet 2D+T (cf. section 2.2) apporte une discrimination pertinente, au sens des textures dynamiques, sur des phénomènes non locaux se propageant temporellement.

Elle semble donc particulièrement intéressante pour modéliser les ondes porteuses présentes dans une texture dynamique et constituera ainsi la première base de notre dictionnaire.

Base pour les phénomènes locaux \mathcal{L} La deuxième partie d'une texture dynamique repose sur des phénomènes localement oscillants. Par conséquent, afin de capturer au maximum ces événements, il faut une base pour une étude locale des oscillations.

Dans la littérature [Starck 04], la transformée en cosinus locale paraît la plus pertinente pour représenter ces informations. Ces coefficients $c(\omega_1, \omega_2, \omega_3)$ se définissent de la manière suivante :

$$c(\omega_1, \omega_2, \omega_3) = \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} \sum_{t=0}^{N_t-1} f(x, y, t) \cos\left(\frac{(2x+1)\omega_1\pi}{2N_x}\right) \cos\left(\frac{(2y+1)\omega_2\pi}{2N_y}\right) \cos\left(\frac{(2t+1)\omega_3\pi}{2N_t}\right) \quad (4.13)$$

avec f une séquence discrète de taille (N_x, N_y, N_t) . La taille de la fenêtre d'étude est déterminée par (N_x, N_y, N_t) . Ainsi, la transformée en cosinus locale est la mieux adaptée à la capture des phénomènes locaux présents dans les textures dynamiques.

Ces réflexions conduisent à la définition suivante :

Définition 14. *Une texture dynamique s'écrivant comme la somme d'une onde porteuse \mathcal{P} et de phénomènes locaux \mathcal{L} peut se décomposer à l'aide de l'analyse en composantes morphologiques en utilisant le dictionnaire Φ suivant :*

- Φ_1 : la transformée en curvelets 2D+T
- Φ_2 : la transformée en cosinus locale 2D+T

Nos expérimentations sont effectuées sur des séquences de la base de données DynTex. Les séquences traitées ont une durée d'environ 5 secondes (128 images) et une taille de 648×540 pixels (ce qui représente plus de 44 millions de voxels). Sur des volumes de cette taille, les transformées de notre dictionnaire (transformée en curvelets 2D+T et transformée en cosinus locale 2D+T) nécessitent un temps de calcul important, plusieurs minutes pour certaines.

Soit la fonction $T()$ mesurant le temps d'exécution d'une transformée d'un signal dans la base Φ_i durant un cycle de l'algorithme (analyse via \mathbf{T}_i et synthèse via \mathbf{T}_i^T). Pour notre dictionnaire, les temps obtenus sont présentés dans la table 4.1.

	Plateforme 1	Plateforme 2
$\Phi_1 : T(\mathbf{T}_1) \approx T(\mathbf{T}_1^T)$	≈ 259 secondes	≈ 109 secondes
$\Phi_2 : T(\mathbf{T}_2) \approx T(\mathbf{T}_2^T)$	≈ 120 secondes	≈ 85 secondes

TAB. 4.1 – Temps de calcul nécessaire pour effectuer une analyse \mathbf{T}_i ou une synthèse \mathbf{T}_i^T avec le dictionnaire (Φ_1 : transformée en curvelets, Φ_2 : transformée en cosinus locale) choisi, selon deux configurations matérielles différentes.

Nous travaillons ici sur deux plateformes différentes :

- **Plateforme 1** : Processeurs 32 bit 2.4GHz, 4Go de RAM
- **Plateforme 2** : Processeurs 64 bit 3.2GHz, 24Go de RAM

Temps d'exécution de l'algorithme de l'analyse en composantes morphologiques

- Stratégies SSL et SSE : le temps d'exécution des algorithmes SSL et SSE dépend principalement du nombre d'analyses et de synthèses effectuées (on négligera le temps d'exécution des autres opérations). Pour une itération, une analyse et une synthèse par base du dictionnaire sont calculées afin de projeter le résidu dans la base puis de reconstruire les coefficients sélectionnés (cf. algorithme 5). Pour ces deux stratégies, le nombre d'itérations N_{iter} de l'algorithme est un paramètre fixé.

Dans la littérature [Bobin 07], concernant ces stratégies, les auteurs s'accordent à dire qu'une centaine d'itérations est nécessaire à l'algorithme MCA pour établir une bonne séparation des composantes morphologiques. Ainsi pour le dictionnaire choisi, et pour une séquence d'images de 5 secondes, ceci représente un temps de calcul de : $100 * (T(\mathbf{T}_1) + T(\mathbf{T}_1^T) + T(\mathbf{T}_2) + T(\mathbf{T}_2^T))$, soit environ 21 heures dans le cadre de la plateforme 1 et environ 10 heures 45 dans le cadre de la seconde.

Si nous extrapolons ce résultat à l'ensemble des séquences de la base de données DynTex, et toujours pour une durée de séquence de 5 secondes, nous obtenons environ 612 jours de calcul pour effectuer la décomposition sur la plateforme 1, et 291 jours pour la seconde.

- Stratégie SSMoM : comme il est précisé dans la section précédente, il n'est plus nécessaire de fixer N_{iter} . Par contre, une projection supplémentaire sur les bases du dictionnaire est nécessaire (calcul du seuil $\lambda^{(k)}$).

Regardons maintenant le gain apporté par la stratégie SSMoM, en terme de nombre d'itérations de la boucle principale. Sur la base de données DynTex, nous observons qu'il faut en moyenne une cinquantaine d'itérations pour effectuer la décomposition. Ainsi, le temps de calcul est approximativement de : $50 * (2 * T(\mathbf{T}_1) + T(\mathbf{T}_1^T) + 2 * T(\mathbf{T}_2) + T(\mathbf{T}_2^T))$, soit environ 15 heures 45 pour la plateforme 1, respectivement 8 heures pour la plateforme 2. Ceci nous amène à environ 453 jours, respectivement 232 jours, pour l'ensemble de la base.

Pour l'indexation d'une base comme DynTex, les temps de calcul de la stratégie SSMoM ne sont pas acceptables. Pourtant, il est toujours possible, d'une part, de répartir la charge de calcul sur plusieurs unités, et d'autre part, dans le cadre de la recherche d'une texture particulière à l'aide d'une séquence requête, ces calculs peuvent s'effectuer à partir de séquences de durée limitée et de résolution faible.

Un des objectifs est de diminuer ces contraintes en proposant de nouvelles stratégies de seuillage permettant d'accélérer les temps de calcul.

4.2.2 Nouvelles stratégies de seuillage

4.2.2.1 Théorie

Comme vu précédemment, la qualité des résultats de la décomposition d'un signal à l'aide de l'algorithme MCA dépend fortement de l'évolution du seuil $\lambda^{(k)}$ au cours d'une itération de la

boucle principale. Nous montrons sur la figure 4.5 deux évolutions différentes de $\lambda^{(k)}$ correspondant à deux stratégies (S1) et (S2). L'évolution de $\lambda^{(k)}$ est plus lente dans le cas (S1) que dans celui de (S2). Dans cet exemple, l'évolution (S1), respectivement (S2), conduit à répartir 5% de la plage des coefficients, respectivement 25%, sur les deux bases. Si on considère que l'évolution (S1) est, dans cet exemple, optimale en terme de seuillage, une évolution non maîtrisée de la valeur de $\lambda^{(k)}$ (cas (S2)) amènera à répartir trop rapidement un grand nombre de coefficients dans les bases, dégradant ainsi la décomposition.

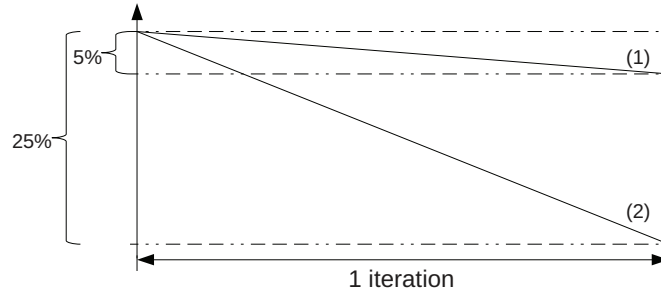


FIG. 4.5 – Deux stratégies de seuillage conduisant à des évolutions différentes de la valeur de seuil durant une itération de la boucle principale de l'algorithme des MCA.

La stratégie de seuillage linéaire, SSL, conduit à une évolution optimale de $\lambda^{(k)}$ lorsqu'une certaine d'itérations est fixée [Bobin 07]. Dans un grand nombre de textures naturelles, nous constatons que le nombre d'itérations peut être fortement réduit. Ce nombre d'itérations dépendant de la texture elle-même, la stratégie SSL n'est alors plus optimum. Cependant, l'évolution selon SSL peut être considérée comme une pente minimum en dessous de laquelle l'évolution de $\lambda^{(k)}$ sera sous-optimum. Une bonne stratégie pour le calcul de $\lambda^{(k)}$ doit donc conduire à une pente supérieure ou égale à celle engendrée par SSL.

La stratégie *Mean of Max*, SSMoM, est très intéressante car elle permet de modifier l'évolution de $\lambda^{(k)}$ lorsque cela s'avère nécessaire. Sur les séquences de textures naturelles, cette stratégie a cependant souvent tendance à diminuer fortement la pente, voire presque à l'annuler. En effet, si le résidu projeté dans les bases du dictionnaire est bien corrélé à plusieurs atomes de celles-ci, les valeurs de m_1 et m_2 sont très proches. Ainsi le seuil ne va pas décroître rapidement et l'algorithme va avoir besoin de beaucoup d'itérations pour décomposer le signal.

Stratégie de seuillage adaptative à correction linéaire Nous proposons de rassembler ces deux stratégies en une nouvelle dite adaptative à correction linéaire, SSaCL, qui définit $\lambda^{(k)}$ comme le minimum des valeurs de $\lambda^{(k)}$ calculées par les stratégies SSL (avec comme paramètre $N_{iter} = 100$) et SSMoM. SSaCL se formalise donc ainsi :

$$\lambda^{(k+1)} = \min \left(\frac{1}{2}(m_1 + m_2), \lambda^{(k)} - \frac{\lambda^{(1)} - \lambda_{\min}}{100} \right) \quad (4.14)$$

avec :

$$m_1 = \max_i \left\| \mathbf{T}_i r^{(k)} \right\|_{\infty}, \quad m_2 = \max_{j, j \neq i_0} \left\| \mathbf{T}_j r^{(k)} \right\|_{\infty} \quad \text{avec } i_0 = \operatorname{argmax}_i \left\| \mathbf{T}_i r^{(k)} \right\|_{\infty} \quad \text{et } r^{(k)} = y - \sum_{j=1}^K \hat{y}_j^{(k)}$$

Ainsi, avec cette stratégie, nous sommes assurés de modifier la valeur de $\lambda^{(k)}$ selon la plus grande pente.

Autrement dit, lorsque SSMoM conduit à des valeurs de $\lambda^{(k)}$ évoluant faiblement, les valeurs de $\lambda^{(k)}$ suivent la stratégie SSL, $\lambda^{(k+1)} = \lambda^{(k)} - \frac{\lambda^{(1)} - \lambda_{\min}}{100}$. Sinon, $\lambda^{(k)}$ suit la stratégie SSMoM, $\lambda^{(k+1)} = \frac{1}{2}(m_1 + m_2)$, et permet ainsi de diminuer le nombre de boucles principales de l'algorithme 5.

Stratégie de seuillage adaptative à correction exponentielle Nous proposons une seconde stratégie de seuillage, appelée stratégie de seuillage adaptative à correction exponentielle, SSACe, combinant l'approche SSE et SSMoM. Elle se formalise de la manière suivante :

$$\lambda^{(k+1)} = \min \left(\frac{1}{2}(m_1 + m_2), \lambda^{(k)} * \left(\lambda^{(1)} - \lambda_{\min} \right)^{-1/99} \right) \quad (4.15)$$

Autrement dit, lorsque SSMoM conduit à des valeurs de $\lambda^{(k)}$ évoluant faiblement, les valeurs de $\lambda^{(k)}$ suivent la stratégie SSE, $\lambda^{(k+1)} = \lambda^{(k)} * \left(\lambda^{(1)} - \lambda_{\min} \right)^{-1/99}$. Sinon, $\lambda^{(k)}$ suit la stratégie SSMoM, $\lambda^{(k+1)} = \frac{1}{2}(m_1 + m_2)$, et permet, tout en diminuant de manière importante la pente lors de l'approche de l'origine, de diminuer également le nombre d'itérations de la boucle principale de l'algorithme 5.

Les deux nouvelles stratégies de seuillage, SSACl et SSACe, ont été implémentées dans l'algorithme de l'analyse en composantes morphologiques étendu aux séquences d'images. Nous proposons dans la section suivante d'étudier la qualité de la décomposition obtenue selon les stratégies de seuillage. Pour cela, nous proposons plusieurs critères.

4.2.2.2 Évaluation

Dans un premier temps, nous prendrons comme critère le gain en temps de calcul, puis dans un second temps, nous étudions plusieurs critères permettant d'évaluer la qualité de la décomposition.

Temps de calcul

Dans les deux cas, le nombre d'itérations nécessaire à la décomposition d'une vidéo issue de DynTex diminue fortement. En effet, il faut en moyenne 12 itérations pour SSACl et 17 itérations pour SSACe pour réaliser la décomposition.

Le gain en terme de temps de calcul n'est cependant pas proportionnel au nombre d'itérations supprimées. En effet, comme dans le cas de la stratégie SSMoM, les stratégies SSACl et SSACe ont besoin d'une projection supplémentaire sur l'ensemble des bases afin de calculer m_1 et m_2 .

Le temps de calcul pour une séquence d'images correspond donc à la relation : (Nombre d'itérations) * ($2 * T(\mathbf{T}_1) + T(\mathbf{T}_1^T) + 2 * T(\mathbf{T}_2) + T(\mathbf{T}_2^T)$). Les performances moyennes des deux stratégies SSACl et SSACe sont estimées dans le tableau 4.2 sur l'ensemble de la base DynTex. Les temps de calcul des stratégies de la littérature y sont également résumés.

	SSL	SSE	SSMoM	SSAcL	SSAcE	
Plateforme 1	≈ 21h	≈ 21h	≈ 15h45	≈ 3h45	≈ 5h20	Temps estimé pour une vidéo
Plateforme 2	≈ 10h45	≈ 10h45	≈ 8h	≈ 1h55	≈ 2h45	
Plateforme 1	≈ 605j	≈ 605j	≈ 453j	≈ 108j	≈ 154j	Temps estimé pour l'ensemble de DynTex
Plateforme 2	≈ 309j	≈ 309j	≈ 232j	≈ 56j	≈ 79j	

TAB. 4.2 – Temps de calcul estimés nécessaires pour effectuer la décomposition à l'aide de l'algorithme des MCA selon les différentes stratégies de seuillage. Il s'agit du temps moyen estimé sur la base DynTex pour une vidéo. De plus, nous indiquons le nombre de jours pour la décomposition complète de la base, ceci pour les deux plateformes utilisées.

Dans le cas de l'utilisation de la plateforme 2, environ 2h sont nécessaires pour réaliser la décomposition d'une vidéo d'une taille de $720 \times 576 \times 128$ voxels à l'aide de la stratégie SSAcL. Ceci réduit d'un facteur 5 les temps de calcul par rapport aux stratégies originelles SSL et SSE.

Si nous étendons ce résultat à l'ensemble de la base de données de DynTex, il faut environ 60 jours pour effectuer une décomposition à l'aide des stratégies SSAcL et SSAcE. De plus, ce temps de calcul peut être divisé par le nombre de cœurs du serveur que l'on utilise. Dans notre cas, la décomposition de l'ensemble des séquences d'images de textures dynamiques, soit environ 700, a été effectuée en une semaine.

En terme de temps de calcul, le gain réalisé à l'aide des stratégies SSAcL et SSAcE paraît très intéressant. Nous présentons maintenant une évaluation de la qualité.

Critères d'évaluation en norme ℓ_0

Dans un premier temps, on étudie la qualité de la décomposition à l'aide des stratégies SSAcL et SSAcE au travers de la norme ℓ_0 par rapport aux stratégies originelles SSL et SSE.

Le nombre de coefficients sélectionnés par l'algorithme MCA au cours des itérations k après l'étape de seuillage est :

$$N_{coef}^{(k)} = \sum_{i=1}^N \left\| \alpha_i^{(k)} \right\|_0 \quad (4.16)$$

Cet indicateur permet de juger de la qualité de la stratégie de seuillage. En effet, pour une stratégie peu ou mal adaptée, le nombre de coefficients retenus sera irrégulier au cours des itérations. Ce phénomène s'observe donc par une courbe à croissance non régulière. Autrement dit, au cours des itérations, plus $N_{coef}^{(k)}$ évolue régulièrement, plus la stratégie de seuillage peut être considérée comme performante.

Sur les courbes de la figure 4.6, l'évolution de $N_{coef}^{(k)}$ a été calculée pour les stratégies SSL, SSE, SSAcL et SSAcE pour quatre séquences différentes ; elles sont affichées ici en échelle logarithmique. Les observations effectuées sur ces quatre vidéos peuvent être également constatées sur l'ensemble des séquences que nous avons testées (environ 200 séquences d'images) :

- Les deux stratégies SSAcL et SSAcE ont un comportement similaire sur les premières itérations et divergent par la suite. Lors des premières itérations, les deux stratégies choisissent la pente maximale qui est $\frac{1}{2}(m_1 + m_2)$, ensuite chacune d'elle se comporte différemment selon leur correction respective.

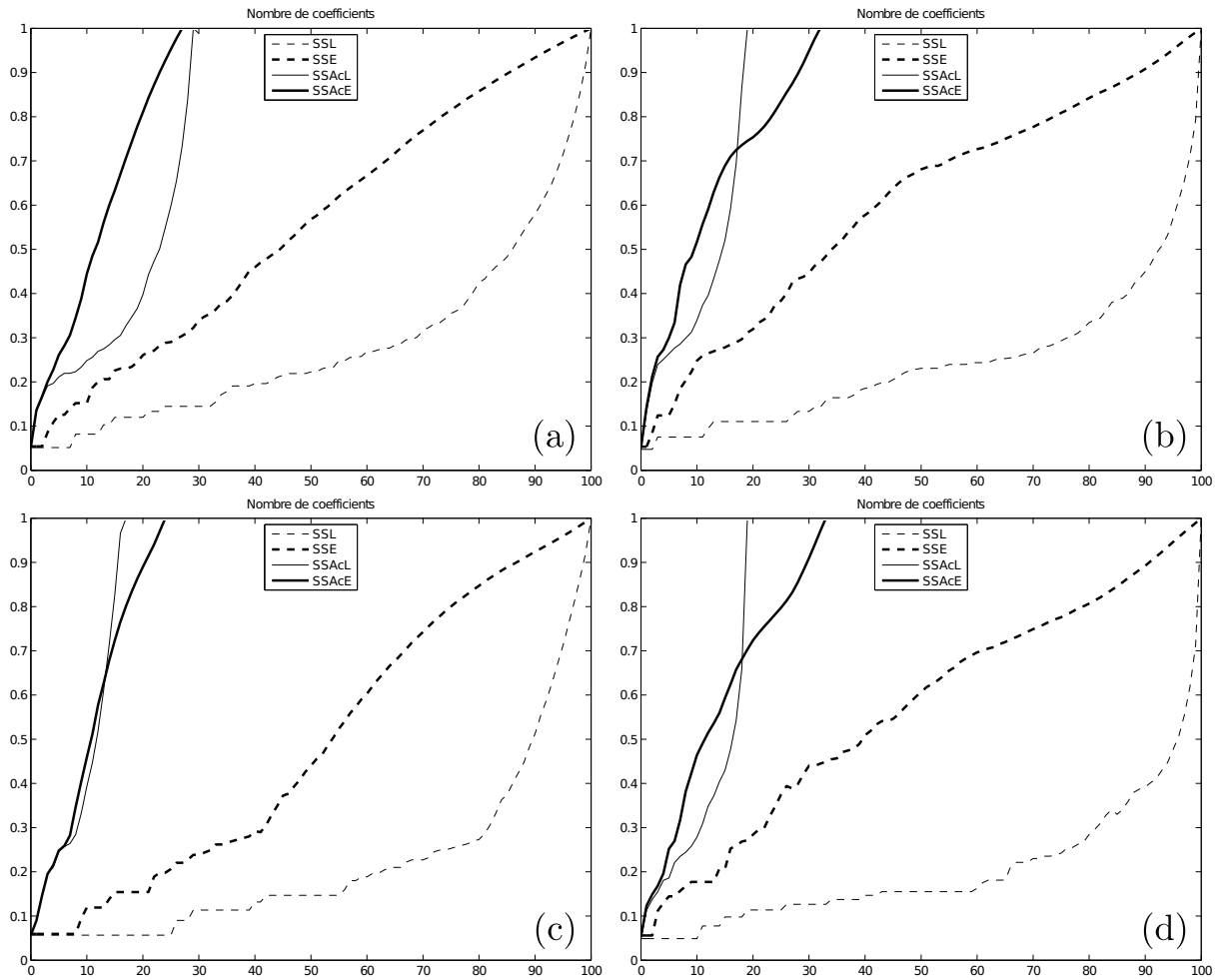


FIG. 4.6 – Étude des stratégies SSL, SSE, SSaL et SSaE. Les courbes représentent le nombre de coefficients seuillés (en échelle logarithmique) au cours des itérations sur quatre séquences d'images différentes.

- Dans le cadre de la stratégie SSL, la plupart des coefficients sont sélectionnés dans les dernières itérations de l'algorithme (les 20 dernières environ). Ceci s'observe par la croissance rapide de la courbe de $N_{coef}^{(k)}$ sur les dernières étapes. Ainsi cette stratégie peut conduire à effectuer beaucoup d'itérations sans sélection de coefficients, puis à une sélection trop rapide à la fin.
- En observant la croissance de $N_{coef}^{(k)}$ pour les stratégies SSaL et SSaE, nous constatons que celle de SSaE est plus régulière. En effet, le nombre de coefficients sélectionnés par la stratégie SSaL augmente rapidement sur les dernières itérations contrairement à SSaE.

Observons maintenant, pour chaque base Φ_i , l'erreur commise sur le choix des coefficients suite au processus de décomposition. Pour cela, les coefficients sélectionnés pour deux stratégies de seuillage différentes sont étudiés et les différences comptabilisées.

Ce critère se formalise de la manière suivante :

$$\forall i, \zeta_i = \frac{1}{M} \left\| \Gamma \left(\alpha_i^{(S1)} \right) - \Gamma \left(\alpha_i^{(S2)} \right) \right\|_0 \quad (4.17)$$

avec $S1$ et $S2$ les deux stratégies en compétition, $\alpha_i^{(S1)}$ (respectivement $\alpha_i^{(S2)}$) les coefficients de la base Φ_i retenus par l'algorithme MCA en utilisant la stratégie $S1$ (respectivement $S2$), M le nombre total de coefficients α_i , et $\Gamma(A)$ l'opérateur mettant la valeur des coefficients à 1 s'ils sont différents de 0.

Le tableau 4.3 présente les moyennes et écarts-types des erreurs sur les coefficients mal sélectionnés entre deux stratégies en compétition (par exemple SSL et SSAcL) pour chacune des bases de notre dictionnaire. Ces données ont été obtenues en utilisant 200 vidéos (soit 1/3 de la base DynTex). ζ_1 (respectivement ζ_2) représente l'erreur commise sur la composante de la transformée en curvelets 2D+T (respectivement la composante de la transformée en cosinus locale 2D+T).

	SSL				SSE			
	ζ_1		ζ_2		ζ_1		ζ_2	
	μ	σ	μ	σ	μ	σ	μ	σ
SSAcL	6.01%	3.12	0.24%	0.16	15.93%	11.12	1.66%	1.48
SSAcE	16.66%	11.56	0.90%	0.70	2.46%	1.67	0.98%	1.03

TAB. 4.3 – Moyennes et écarts-types de l'erreur sur les coefficients mal sélectionnés d'une stratégie $S1$ (par exemple SSAcL) par rapport à une stratégie de référence $S2$ (par exemple SSL) pour chacune des bases de notre dictionnaire (ζ_1 pour la transformée en curvelets 2D+T et ζ_2 pour la transformée en cosinus locale 2D+T).

On remarque, selon le critère (4.17), que :

- Les deux stratégies SSAcL et SSAcE approchent les solutions obtenues par les stratégies SSL et SSE. En effet, la moyenne la plus grande des erreurs est de 16.66%. Toutefois, ceci correspond à la comparaison entre les stratégies SSAcE et SSL, qui n'est pas la plus aisée à effectuer. En effet, même si les composantes finales issues des deux stratégies sont similaires (ceci sera abordé par la suite), la position des coefficients sélectionnés peut différer.
- La stratégie de seuillage adaptative approchant le plus une des stratégies originelles est la stratégie SSAcE. Ceci s'observe par les moyennes et les écart-types des erreurs ζ_1 et ζ_2 qui sont faibles.
- Comme le montrait l'observation du critère (4.16), la stratégie SSAcL a tendance à sélectionner rapidement de nombreux coefficients pouvant ainsi conduire à des erreurs de décomposition. Ceci s'observe par une moyenne des erreurs plus élevée que dans le cas de l'utilisation de SSAcE. Cette dernière passent, quant à elle, beaucoup plus d'itérations à répartir les coefficients des plages autour de l'origine, conduisant ainsi à une erreur minime. On retrouve ces constatations pour d'autres critères.

Critères d'évaluation en norme ℓ_2

Après avoir étudié l'influence des stratégies sur la sélection des coefficients, nous calculons l'erreur de reconstruction en norme ℓ_2 de l'algorithme au cours des itérations. Celle-ci s'écrit :

$$\xi_T^{(k)} = \left\| y - \sum_{i=1}^N \tilde{y}_i^{(k)} \right\|_2 \quad (4.18)$$

Comme pour le critère (4.16), c'est la régularité de la courbe qui traduit la performance de la stratégie de seuillage : plus l'erreur de reconstruction diminue rapidement et régulièrement au cours des itérations, plus les coefficients sélectionnés sont pertinents et représentatifs du signal d'origine. Concernant les courbes de la figure 4.7, l'évolution de $\xi_T^{(k)}$ a été calculée pour les stratégies SSL, SSE, SSaCL et SSaCE sur quatre séquences d'images. Comme pour le critère (4.17), les résultats obtenus sur ces quatre vidéos sont représentatifs de l'ensemble des tests effectués (200 vidéos).

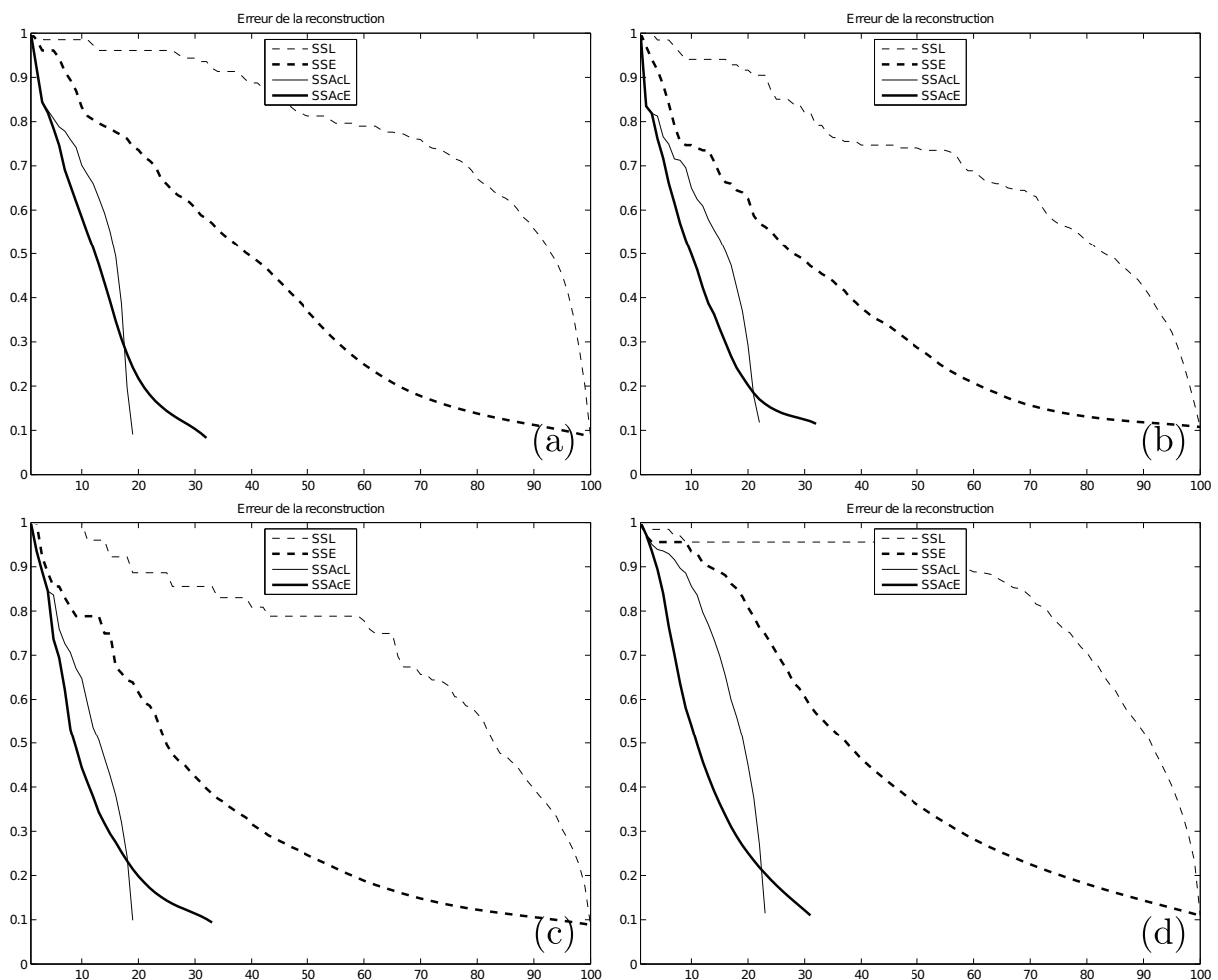


FIG. 4.7 – Étude des stratégies *SSL*, *SSE*, *SSaCL* et *SSaCE* : les courbes représentent l'erreur de reconstruction en norme ℓ_2 en fonction du nombre d'itérations.

Plusieurs constatations peuvent être effectuées :

- dans le cadre de la stratégie SSL, la plupart des coefficients représentatifs du signal d’origine sont sélectionnés sur les dernières itérations de l’algorithme. Ceci s’observe par la décroissance rapide de la courbe de $\xi_T^{(k)}$.
- la stratégie SSE sélectionne dans les premières itérations de nombreux coefficients pertinents. Une décroissance rapide de $\xi_T^{(k)}$ est donc observée. Ensuite, celle-ci va diminuer pour les derniers coefficients à répartir. Ce phénomène s’explique par le fait que la taille des plages de coefficients sélectionnés diminue au fur et à mesure des itérations de l’algorithme (cf. section 4.1.2.3 et figure 4.2).
- concernant les deux stratégies adaptatives SSAcL et SSAcE, l’erreur de reconstruction diminue très rapidement. Les deux stratégies sélectionnent donc de façon régulière des coefficients pertinents s’approchant ainsi rapidement du signal d’origine.
- la stratégie adaptative SSAcE paraît plus performante que SSAcL. En effet, nous observons une chute brutale de l’erreur de reconstruction dans le cas SSAcL. Ceci vient du fait que l’algorithme sélectionne beaucoup de coefficients pertinents sur quelques itérations ; la répartition des coefficients peut alors s’avérer problématique contrairement à la stratégie SSAcE pour laquelle la répartition s’effectue sur un nombre d’itérations plus important.

L’étude précédente a porté sur l’erreur en norme ℓ_2 entre la reconstruction et l’originale. Toutefois, nous observons qu’à la dernière itération de l’algorithme (cf. courbes de la figure 4.7) celle-ci est identique pour toutes les stratégies. Ceci s’explique par la construction des différentes stratégies : chacune conduit le seuil $\lambda^{(k)}$ à une valeur minimale λ_{min} fixé par l’utilisateur.

Le critère (4.18) permet donc d’observer la performance des stratégies au cours des itérations de l’algorithme, il ne permet cependant pas de juger de la qualité des différentes composantes extraites au terme des itérations.

Dans ce but, nous proposons un critère qui compare les composantes morphologiques obtenues selon les stratégies concurrentes. Celui-ci s’écrit :

$$\forall i, \quad \xi_i = \frac{1}{\xi_{max}} \left\| \tilde{y}_i^{(S1)} - \tilde{y}_i^{(S2)} \right\|_2 \quad (4.19)$$

avec $\tilde{y}_i^{(S1)}$ (respectivement $\tilde{y}_i^{(S2)}$) la composante morphologique i estimée à l’aide de la stratégie $S1$ (respectivement $S2$). ξ_{max} est calculée pour chaque composante $\tilde{y}_i^{(S2)}$ comme étant la plus grande erreur que l’on peut commettre.

Le tableau 4.4 présente les moyennes et écarts-types des erreurs en norme ℓ_2 sur chacune des bases de notre dictionnaire en fonction des stratégies en compétition. Ces résultats ont été obtenus à partir de 200 séquences d’images. ξ_1 (respectivement ξ_2 et ξ_T) représente l’erreur commise sur la composante de la transformée en curvelets 2D+T (respectivement sur la composante de la transformée en cosinus locale 2D+T et sur la reconstruction).

Nous synthétisons ici les observations effectuées :

- le maximum des moyennes des erreurs commises entre les reconstructions obtenues à partir de deux stratégies différentes est seulement de 0.48%. Ceci indique que, quelle que soit la stratégie employée, la reconstruction sera quasiment identique.

	SSL					
	ξ_T		ξ_1		ξ_2	
	μ	σ	μ	σ	μ	σ
SSAcL	0.27%	0.10	1.56%	1.08	1.56%	0.92
SSAcE	0.48%	0.20	2.30%	1.82	2.18%	1.50

	SSE					
	ξ_T		ξ_1		ξ_2	
	μ	σ	μ	σ	μ	σ
SSAcL	0.39%	0.15	2.67%	1.90	2.43%	1.58
SSAcE	0.38%	0.17	1.55%	1.51	1.50%	1.22

TAB. 4.4 – Moyennes et écart-types de l’erreur en norme ℓ_2 entre deux stratégies $S1$ (par exemple SSAcL) et $S2$ (par exemple SSL) pour chacune des bases de notre dictionnaire (ξ_1 pour la transformée en curvelets $2D+T$, ξ_2 pour la transformée en cosinus locale $2D+T$) et pour sa reconstruction ξ_T .

- Les deux stratégies adaptatives SSAcL et SSAcE approchent assez bien les décompositions obtenues à l’aide des stratégies originelles. En effet, pour chaque composante, l’erreur moyenne maximum observée entre deux stratégies différentes est en effet seulement de 2.67%.

Au vu des différents critères étudiés ainsi que du gain en terme de temps de calcul, nous pouvons établir que les stratégies proposées s’avèrent être pertinentes pour le traitement de séquences de textures dynamiques. Dans la section suivante, nous présentons des résultats de décomposition de textures dynamiques à l’aide de l’algorithme MCA et de la stratégie de seuillage adaptative à correction exponentielle.

4.2.3 Expérimentations

4.2.3.1 Paramètres utilisés

La mise en place des stratégies SSAcL et SSAcE, a permis l’obtention de plusieurs résultats satisfaisants. Dans cette partie, cinq d’entre eux sont détaillés précisément.

Les décompositions des textures dynamiques présentées ici ont été obtenues en utilisant comme dictionnaire la transformée en curvelets $2D+T$ et la transformée en cosinus locale $2D+T$.

La transformée en curvelets a été calculée en utilisant 5 échelles de décomposition, et 4 subdivisions angulaires à chaque changement d’échelle. La transformée en cosinus locale a été effectuée sur des fenêtres $32 \times 32 \times 32$ voxels.

Ces décompositions ont été calculées en utilisant la stratégie de seuillage adaptative à correction exponentielle. Comme dans de nombreuses expérimentations [Fadili 10b], nous utilisons $\lambda_{min} = \tau\sigma_\epsilon$ avec $\tau = 3$.

4.2.3.2 Résultats

Vidéo 644ce10 La première vidéo est celle d'un lac sur lequel dérive lentement un canard (figure 4.8). Des reflets d'arbres dans l'eau ondulante et un fond statique texturé peuvent être observés. La figure présente le résultat de la décomposition obtenue sur cette séquence à l'aide de l'algorithme MCA et selon la stratégie SSACeE proposée.

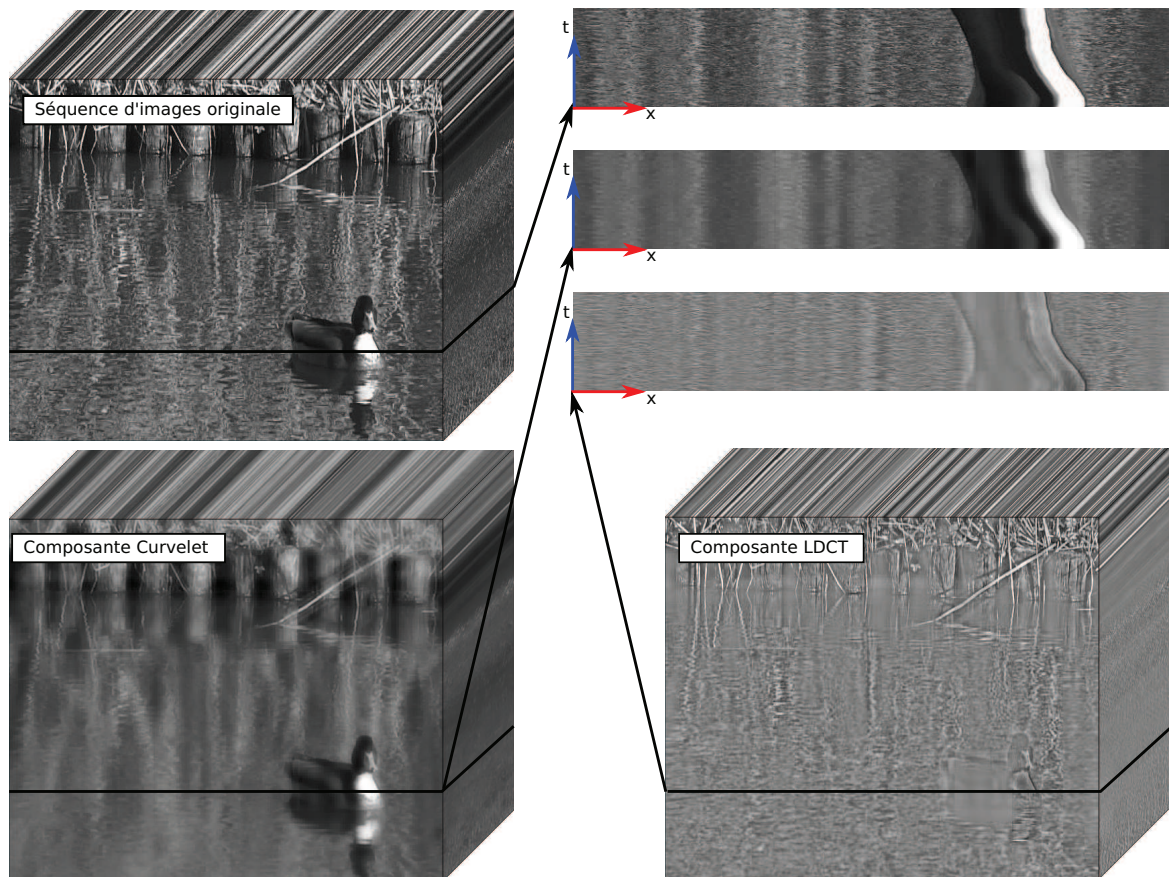


FIG. 4.8 – ♠ Résultat de la décomposition d'une séquence d'images à l'aide de l'algorithme MCA et selon la stratégie SSACeE. Une coupe spatio-temporelle xt est réalisée sur chacune des vidéos afin d'observer le comportement temporel.

Nous retrouvons la composante géométrique apportée par la transformée en curvelets et la composante texture obtenue par la transformée en cosinus locale. Les vaguelettes, qui sont des phénomènes locaux, sont bien capturées par la composante texture, alors que la surface de l'eau et les reflets sont présents dans la composante géométrique.

Les coupes spatio-temporelles selon l'axe xt permettent de visualiser l'impact de la décomposition. Elles montrent que les différents objets de la scène (le canard, les reflets d'arbres, ...) sont considérés comme étant de nature géométrique. Les reflets des arbres ne sont pas présents dans la composante texture. On peut par contre les observer dans la composante géométrique. Les oscillations qui se superposaient à ces reflets se retrouvent eux dans la composante texture.

La décomposition apporte également des informations qui n'étaient pas visibles sur la séquence d'images originale. Par exemple, dans la composante texture, on peut maintenant observer sous le cou du canard, la granularité du plumage, invisible dans la séquence initiale.

Vidéo 646a510 La deuxième séquence d'images utilisée est une scène comportant cinq drapeaux agités par un vent violent. Le battement des drapeaux est très rapide et à la limite de la fréquence d'échantillonnage. Les résultats de la décomposition de cette séquence d'images sont visibles sur la figure 4.9.

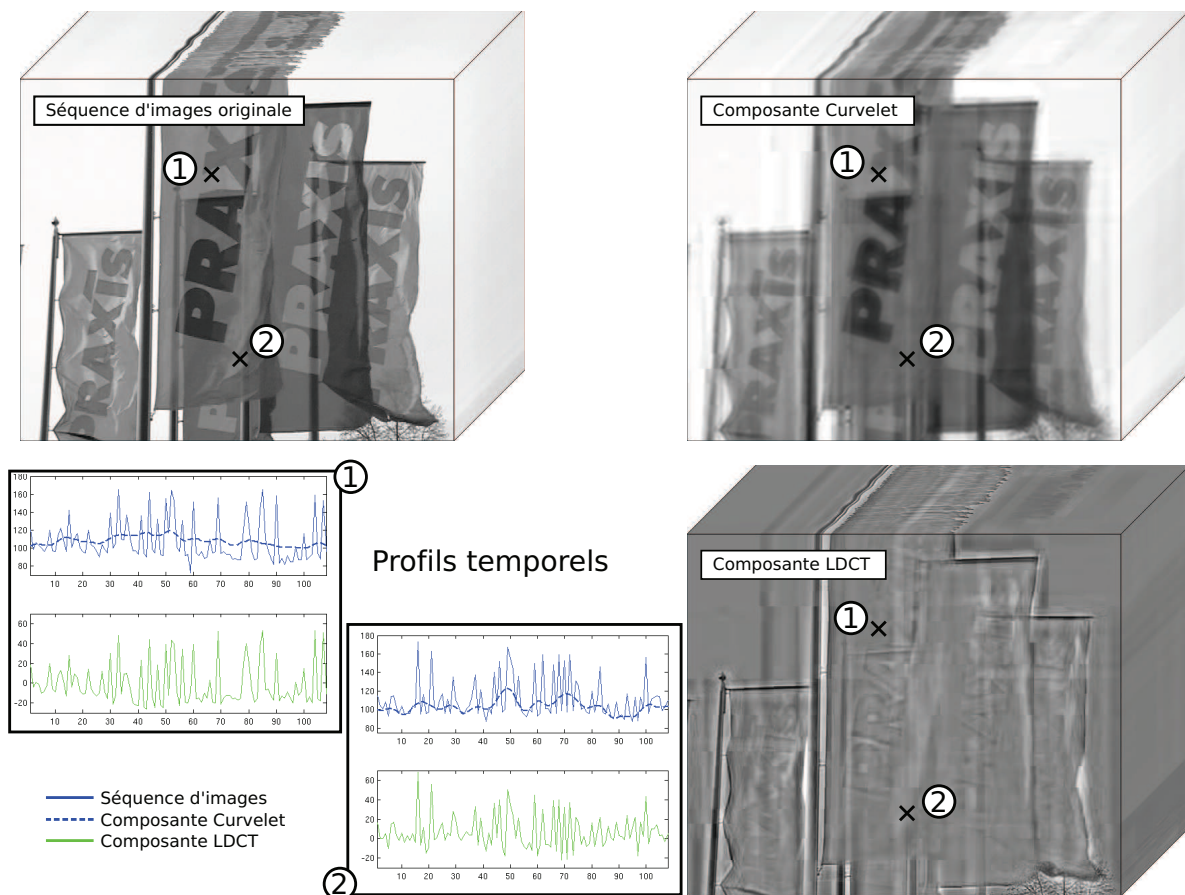


FIG. 4.9 – ♠ Résultat de la décomposition d'une séquence d'images à l'aide de l'algorithme MCA et selon la stratégie SSACe. Des profils temporels ont été extraits afin de visualiser les résultats sur l'axe temporel.

Les oscillations des drapeaux étant visibles sur l'axe temporel uniquement, des profils selon cet axe ont donc été extraits. Nous observons ainsi l'évolution dans le temps de la luminance d'un pixel (x, y) .

L'étude des différents profils temporels montre que la séquence originale est très perturbée. Après décomposition par MCA, les profils temporels témoignent bien de la séparation entre la partie géométrique, représentante du mouvement global des drapeaux, et la partie texture,

représentante des différentes vibrations des drapeaux.

Comme pour la séquence précédente, les objets géométriques sont bien classés puisque les caractères présents sur les drapeaux sont beaucoup moins visible dans la partie texture.

Vidéo 6487310 La séquence d'images suivante représente une fontaine. Celle-ci consiste en un jet d'eau qui, en retombant, agite la surface de l'eau en créant des vaguelettes. Les résultats de la décomposition MCA 2D+t sont affichés sur la figure 4.10.

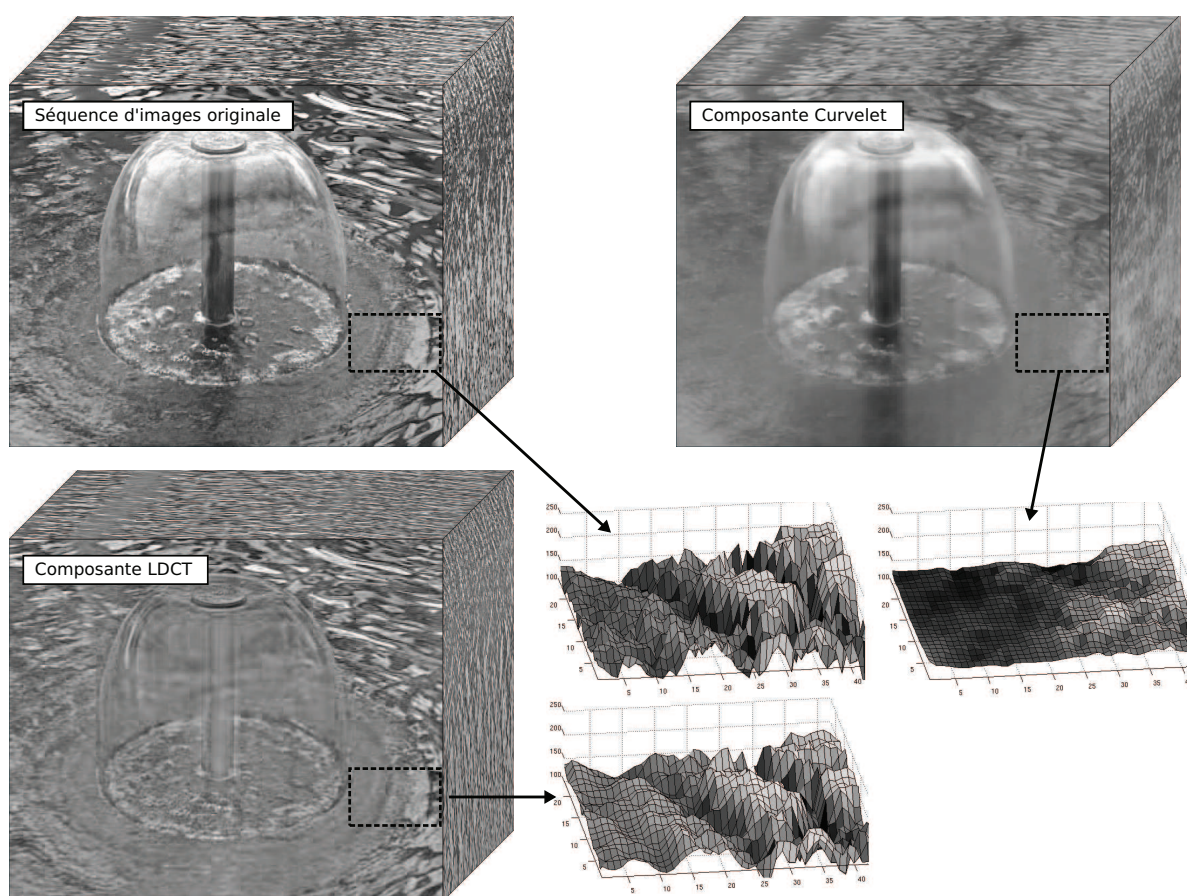


FIG. 4.10 – ♠ *Résultat de la décomposition d'une séquence à l'aide de l'algorithme MCA et selon la stratégie SSACe. Des zones d'intérêt sont mises en évidence sous la forme de surfaces afin de mieux visualiser les effets de l'algorithme.*

Les deux composantes obtenues semblent pertinentes : dans la partie géométrique, la colonne centrale du jet et la forme en cloche engendrée par le jet sont visibles, alors qu'elles sont quasiment absentes de la composante texture.

Nous remarquons également que toute la zone située autour du jet est comme figée et dépourvue de vaguelettes, visibles cependant dans l'autre composante. La représentation surfacique des zones d'intérêt met bien en évidence ce phénomène. En effet, la partie géométrique est privée des vaguelettes ; et on observe seulement une légère ondulation.

Vidéo 54pc210 : L'avant-dernière séquence présentée est celle d'un escalator. De part sa structure, cet exemple est très intéressant : les propriétés sous-jacentes sont très proches du modèle proposé. En effet, un escalator est constitué d'un onde porteuse (les marches) et de phénomènes locaux (les stries des marches). La figure 4.11 montre les résultats obtenus à l'aide de l'algorithme MCA.

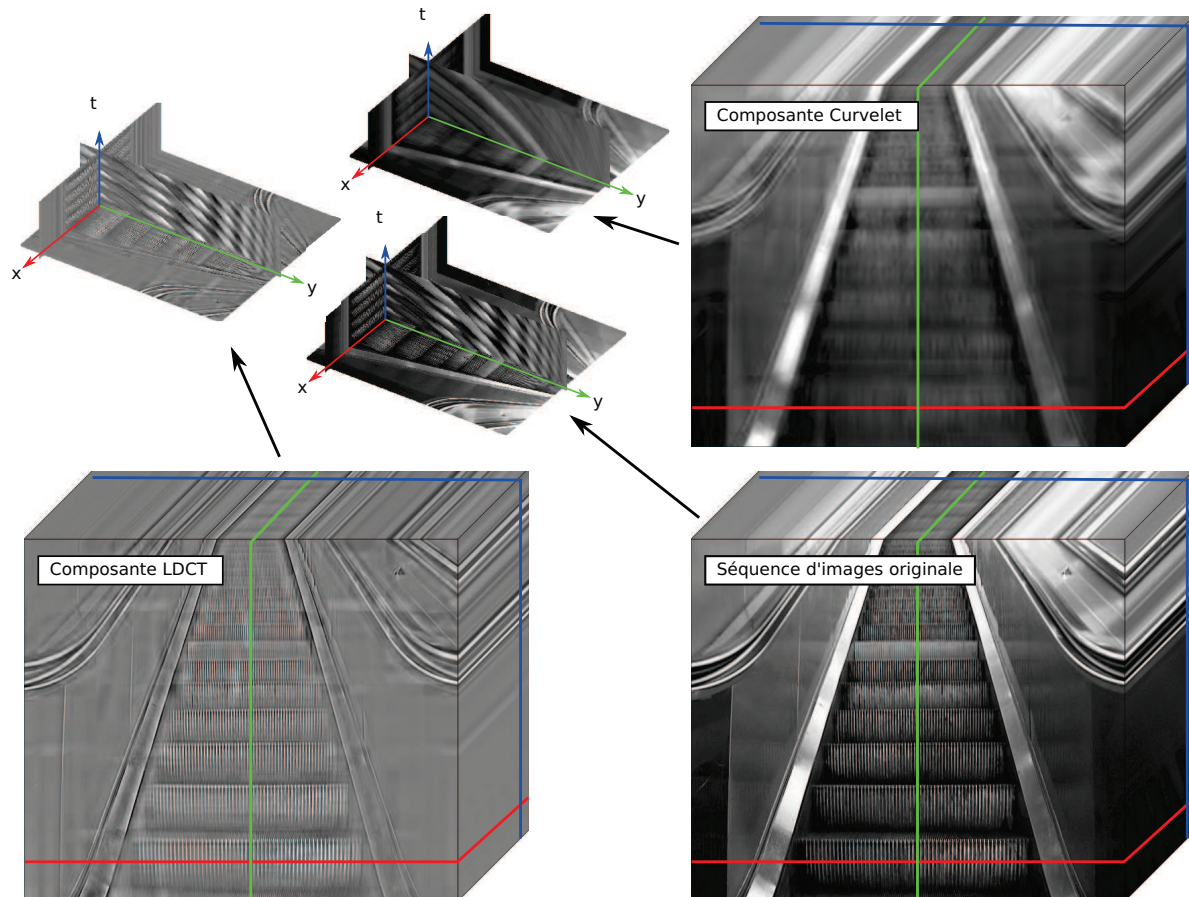


FIG. 4.11 – ♠ Résultat de la décomposition d'une séquence à l'aide de l'algorithme MCA et selon la stratégie SSAC. Des coupes spatio-temporelles dans différentes directions ont été effectuées afin de mieux visualiser les phénomènes.

Sur la séquence originale, les marches de l'escalator dessinent au cours du temps une forme géométrique très caractéristique. Celle-ci est cependant fortement perturbée par les stries de chaque marche.

Une fois la décomposition effectuée, la forme géométrique est clairement visible dans la composante géométrique et plus facilement caractérisable. On retrouve les stries des marches, phénomènes localisés, dans la composante texture.

Vidéo 649dc10 La dernière séquence présentée correspond à la surface de la mer au bord d'une plage. Encore une fois, il s'agit d'une texture dynamique dont les propriétés sont proches

de celles de notre modèle : elle est constituée d'une onde porteuse (les rouleaux des vagues) et de phénomènes locaux (l'écume). La figure 4.12 illustre les résultats de la décomposition obtenue.

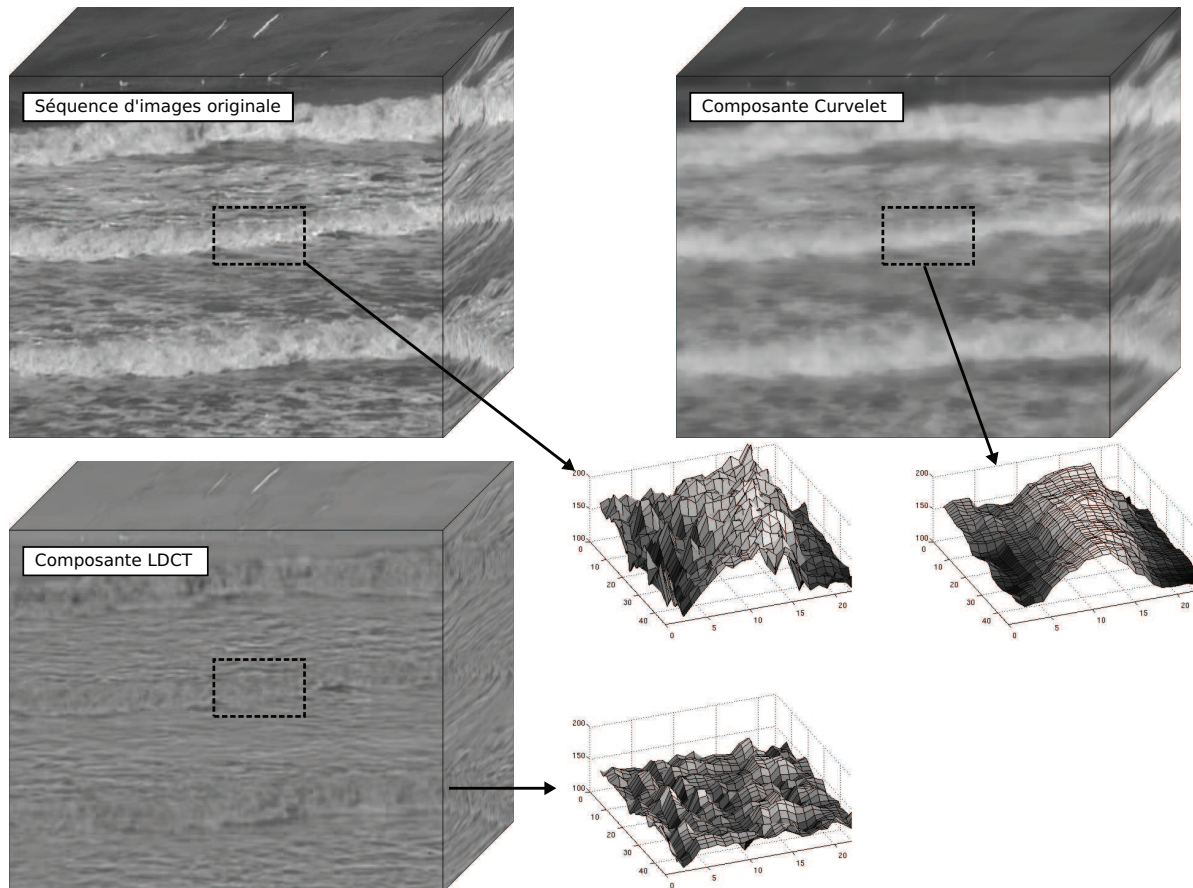


FIG. 4.12 – ♠ Résultat de la décomposition d'une séquence à l'aide de l'algorithme MCA et selon la stratégie SSACe. Des zones d'intérêt sont mises en évidence sous la forme de surfaces afin de mieux comprendre les effets de l'algorithme.

On retrouve les caractéristiques de notre modèle dans chacune des composantes. La partie correspondant à la transformée en curvelets est constituée des rouleaux des vagues alors que celle correspondant à la transformée en cosinus locale contient l'écume et les petites vaguelettes. Dans les zones mises en évidence, la composante texture ne permet pas d'observer les rouleaux des vagues, complètement présents dans la partie géométrique.

Ces différentes observations sont effectuées sur les représentations surfaciques où l'on voit clairement la séparation des rouleaux et de l'écume.

Conclusion du chapitre Dans ce chapitre, en s'appuyant sur le modèle proposé dans la section 3.1, nous avons étudié la décomposition de séquences de textures dynamiques.

Après une introduction des méthodes de décomposition de la littérature, nous nous sommes focalisés sur l'analyse en composantes morphologiques. Cette approche est particulièrement

intéressante de part la richesse du dictionnaire et la souplesse de son algorithme.

L'extension au cadre temporel de l'algorithme MCA est extrêmement coûteux en temps de calcul. Nous proposons donc deux stratégies de seuillage adaptatives afin d'accélérer la convergence de l'algorithme. Ces deux stratégies ont été validées au travers de plusieurs critères fondés sur les normes ℓ_0 et ℓ_2 .

La dernière partie de ce chapitre est consacrée aux résultats de la décomposition obtenue à l'aide de notre stratégie de seuillage adaptative à correction exponentielle.

Afin de mettre en évidence l'intérêt pratique de nos contributions, nous proposons de terminer cette étude par la présentation de plusieurs applications.

CHAPITRE 5

Applications

Applications

Sommaire

5.1	Indexation de textures dynamiques	145
5.1.1	Construction des vecteurs caractéristiques	146
5.1.2	Méthode de validation	148
5.1.3	Bases de données utilisées	149
5.1.4	Résultats et discussion	149
5.2	Recherche par le contenu	153
5.3	Estimation du mouvement global	155

Dans les chapitres précédents, plusieurs outils d'analyse de textures dynamiques ont été mis en place. Dans le chapitre 3, une première application à la segmentation spatio-temporelle de textures dynamiques a été proposée. Elle s'appuie sur les informations apportées par la transformée en curvelets 2D+T. Une deuxième application a été présentée dans le chapitre 4. Elle porte sur la décomposition de séquences d'images de textures dynamiques en composantes géométrique et texture. Ces outils ont permis de mieux analyser les propriétés des textures dynamiques.

Ce chapitre se focalise sur l'indexation. Dans la première partie, nous étudions plusieurs descripteurs issus des transformées multi-échelles classiquement utilisés. Ils seront utilisés sur trois jeux de données : un cas simple mettant en jeu trois classes avec des textures dynamiques bien différenciées, et deux cas plus complexes avec dix classes et des textures dynamiques difficiles.

Nous proposerons ensuite une application de recherche par le contenu : les requêtes se présenteront sous forme de séquences d'images.

Pour finir, nous présentons l'apport de la décomposition de l'analyse en composantes morphologiques pour l'analyse du mouvement global présent sur une séquence d'images.

5.1 Indexation de textures dynamiques

La première application que nous présentons porte sur l'indexation de textures dynamiques. Nous comparons plusieurs approches ainsi que différents descripteurs. L'objectif principal est d'évaluer ces approches et d'identifier les descripteurs les plus pertinents.

Chacune de nos expérimentations d'indexation se déroule de la manière suivante :

- analyse des séquences d'images à l'aide d'une transformée spatio-temporelle étudiée précédemment,
- calcul des descripteurs et construction d'un vecteur caractéristique,
- classement des signatures à l'aide de la méthode *leave-one-out*.

5.1.1 Construction des vecteurs caractéristiques

Nous rappelons ici les différentes approches multi-échelles qui ont été présentées dans notre manuscrit :

- la transformée en ondelettes frame par frame (cf. section 2.1.2.1),
- la transformée en ondelettes temporelles (cf. section 2.1.2.2),
- la transformée en ondelettes 2D+T (cf. section 2.1.2.3),
- l’approche de Smith *et al.* [Smith 02] (cf section 2.1.2.4),
- la transformée en curvelets 2D+T (cf. section 2.2),
- les composantes morphologiques issues de la décomposition à l’aide de la transformée en curvelets 2D+T et de la transformée en cosinus locale 2D+T (cf. section 4.2).

Chacune de ces méthodes est repérée à l’aide d’un indice : $m = \{fpf, t, xyt, xy2t, curv, mca\}$ (les indices suivent l’ordre de la liste ci-dessus).

Dans le cas de la caractérisation de textures 2D pour l’analyse multi-résolutions, les descripteurs sont construits à partir des sous-bandes de détails. Ainsi, $d_m^{(j,\ell)}$ représente la sous-bande de détails de dimensions $(t_1^{(j,\ell)} \times t_2^{(j,\ell)} \times t_3^{(j,\ell)})$ de la transformée multi-résolutions m à l’échelle j pour l’orientation ℓ (horizontal, vertical, ...).

Les descripteurs suivants sont calculés :

- Moyenne des sous-bandes de détails :

$$\forall m, \forall j, \forall \ell, \quad \mu_m^{(j,\ell)} = \frac{1}{t_1^{(j,\ell)} t_2^{(j,\ell)} t_3^{(j,\ell)}} \sum_{k_1=1}^{t_1^{(j,\ell)}} \sum_{k_2=1}^{t_2^{(j,\ell)}} \sum_{k_3=1}^{t_3^{(j,\ell)}} d_m^{(j,\ell)}(k_1, k_2, k_3) \quad (5.1)$$

- Écart-type des sous-bandes de détails :

$$\forall m, \forall j, \forall \ell, \quad \sigma_m^{(j,\ell)} = \sqrt{\frac{1}{t_1^{(j,\ell)} t_2^{(j,\ell)} t_3^{(j,\ell)}} \sum_{k_1=1}^{t_1^{(j,\ell)}} \sum_{k_2=1}^{t_2^{(j,\ell)}} \sum_{k_3=1}^{t_3^{(j,\ell)}} \left(d_m^{(j,\ell)}(k_1, k_2, k_3) - \mu_m^{(j,\ell)} \right)^2} \quad (5.2)$$

- Energie des sous-bandes de détails :

$$\forall m, \forall j, \forall \ell \quad E_m^{(j,\ell)} = \frac{1}{t_1^{(j,\ell)} t_2^{(j,\ell)} t_3^{(j,\ell)}} \sum_{k_1=1}^{t_1^{(j,\ell)}} \sum_{k_2=1}^{t_2^{(j,\ell)}} \sum_{k_3=1}^{t_3^{(j,\ell)}} \left(d_m^{(j,\ell)}(k_1, k_2, k_3) \right)^2 \quad (5.3)$$

- Entropie des sous-bandes de détails

$$\forall m, \forall j, \forall \ell, \quad H_m^{(j,\ell)} = - \sum_{k_1=1}^{t_1^{(j,\ell)}} \sum_{k_2=1}^{t_2^{(j,\ell)}} \sum_{k_3=1}^{t_3^{(j,\ell)}} p(d_m^{(j,\ell)}(k_1, k_2, k_3)) \ln \left(p(d_m^{(j,\ell)}(k_1, k_2, k_3)) \right) \quad (5.4)$$

avec $p(d_m^{(j,\ell)}(k_1, k_2, k_3))$ la densité de probabilité de $d_m^{(j,k)}$ dans la sous-bande de détails ($d_m^{(j,k)}$ est considéré comme une variable aléatoire).

Dans le cadre de la transformée en cosinus locale utilisée dans l’approche MCA, nous ne possédons pas de sous-bandes de détails. Ainsi, comme l’illustre la figure 5.1, nous découpons les coefficients de cette transformée en plusieurs sous-bandes. Chaque sous-bande, est représentative

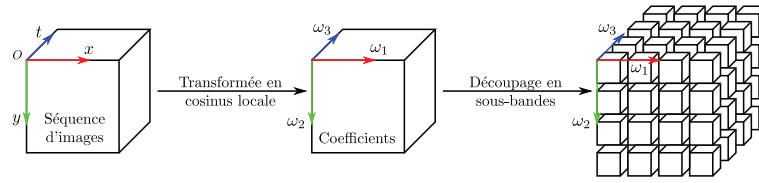


FIG. 5.1 – Découpage des coefficients de la transformée en cosinus locale pour obtenir comme dans une analyse multi-résolutions différentes sous-bandes de détails.

d'un ensemble de fréquences orientées et d'échelles similaires. Les descripteurs sont calculés de la même manière que dans le cadre des transformées multi-échelles.

Dans notre objectif d'indexation, nous observons cinq signatures différentes, quatre construites directement à partir d'un seul descripteur, et une construite à partir de la concaténation des quatre descripteurs :

- Vecteur de descripteurs s'appuyant sur la moyenne des sous-bandes de détails :

$$\mathbf{S}_m^\mu = \left(\mu_m^{(1,1)}, \dots, \mu_m^{(1,N_\ell^1)}, \dots, \mu_m^{(N_j,1)}, \dots, \mu_m^{(N_j,N_\ell^j)} \right) \quad (5.5)$$

avec N_ℓ^j le nombre d'orientations à l'échelle j et N_j le nombre d'échelles.

- Vecteur de descripteurs fondé sur l'écart-type des sous-bandes de détails :

$$\mathbf{S}_m^\sigma = \left(\sigma_m^{(1,1)}, \dots, \sigma_m^{(1,N_\ell^1)}, \dots, \sigma_m^{(N_j,1)}, \dots, \sigma_m^{(N_j,N_\ell^j)} \right) \quad (5.6)$$

- Vecteur de descripteurs s'appuyant sur l'énergie des sous-bandes de détails :

$$\mathbf{S}_m^E = \left(E_m^{(1,1)}, \dots, E_m^{(1,N_\ell^1)}, \dots, E_m^{(N_j,1)}, \dots, E_m^{(N_j,N_\ell^j)} \right) \quad (5.7)$$

- Vecteur de descripteurs calculé à partir de l'entropie des sous-bandes de détails :

$$\mathbf{S}_m^H = \left(H_m^{(1,1)}, \dots, H_m^{(1,N_\ell^1)}, \dots, H_m^{(N_j,1)}, \dots, H_m^{(N_j,N_\ell^j)} \right) \quad (5.8)$$

- Vecteur de descripteurs s'appuyant sur différentes caractéristiques des sous-bandes de détails :

$$\mathbf{S}_m^A = (\mathbf{S}_m^\mu, \mathbf{S}_m^\sigma, \mathbf{S}_m^E, \mathbf{S}_m^H) \quad (5.9)$$

Ainsi pour une base de données vidéos, nous obtenons un ensemble de signatures $\mathbf{S}_{m,c,i}^d$ avec m représentant la méthode spatio-temporelle utilisée, $d = \{\mu, \sigma, E, H, A\}$ étant le(s) descripteur(s) utilisé(s) et i le i -ème échantillon de la classe c de la base.

Dans notre cas, nous normalisons les vecteurs caractéristiques de la manière suivante :

$$\forall n, \forall r, \forall j, \mathbf{S}_{m,r,j}^d(n) = \frac{\mathbf{S}_{m,r,j}^d(n) - \min_{c,i} \mathbf{S}_{m,c,i}^d(n)}{\max_{c,i} \mathbf{S}_{m,c,i}^d(n) - \min_{c,i} \mathbf{S}_{m,c,i}^d(n)} \quad (5.10)$$

avec n représentant l'indice du paramètre dans le vecteur caractéristique.

D'autres normalisations ont été testées, celles-ci sont visibles en annexe D.

5.1.2 Méthode de validation

Pour chaque approche m et chaque type de vecteurs de descripteurs d , on étudie la pertinence de l'approche en calculant un taux de reconnaissance à partir de la matrice de confusion.

Nous utilisons la méthode *leave-one-out*, illustrée par la figure 5.2 et dont le mode opératoire est le suivant :

- Calcul des centres pour chacune des classes c :

$$\forall c, \boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{S}_{m,c,i}^d \quad (5.11)$$

- Pour chaque élément p de chaque classe c :
 - Recalcul du centre de la classe privée de celui-ci :

$$\boldsymbol{\mu}_c^{p-} = \frac{1}{N_c - 1} \sum_{\substack{i=1 \\ i \neq p}}^{N_c} \mathbf{S}_{m,c,i}^d \quad (5.12)$$

- Recherche du centre de la classe r se rapprochant le plus de l'élément p :

$$r = \underset{u}{\operatorname{argmin}} \left(\sum_n \left(\boldsymbol{\mu}_u(n) - \mathbf{S}_{m,c,p}^d(n) \right)^2 \right) \quad (5.13)$$

- Si $r \neq c_p$ une mauvaise classification est comptabilisée.

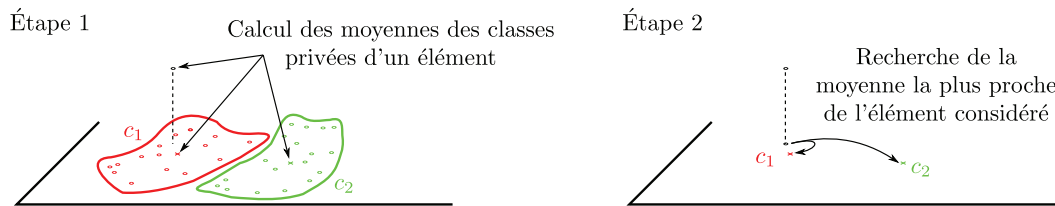


FIG. 5.2 – Illustration de la méthode *leave-one-out* pour un élément.

La méthode du *leave-one-out* conduit à l'obtention d'une matrice de confusion représentative de la pertinence de la signature.

Par exemple, la table 5.1 présente la matrice de confusion de trois classes A, B et C.

	A	B	C
A	10	0	0
B	3	2	5
C	9	1	0

TAB. 5.1 – Exemple d'une matrice de confusion pour trois classes.

Les éléments diagonaux correspondent aux cas de bonnes classifications. Nous remarquons que les signatures utilisées sont très représentatives de la classe A puisque tous les éléments de cette classe ont été redistribués dans la même classe. Par contre, les vecteurs de descripteurs sont moins performants pour la classe B et médiocre pour la classe C.

5.1.3 Bases de données utilisées

Nous présentons dans cette section les résultats d'indexation obtenus sur des textures dynamiques. L'ensemble des approches multi-échelles ont été effectuées sur 5 niveaux de résolution. Seule l'approche de Smith *et al.* [Smith 02] est évaluée sur 3 niveaux, sa construction ne permettant pas d'avoir plus de niveaux de résolution.

Chacune de ces méthodes est ensuite testée sur l'ensemble des descripteurs décrits précédemment et ceci sur 3 bases de données de complexité croissante.

Les expérimentations ont également été menées sur d'autres paramètres (selon le nombre d'échelles de décomposition, selon la normalisation des descripteurs) mais nous ne présentons ici qu'une partie des résultats, les autres sont disponibles en annexe (cf. Annexe D).

Les trois bases de données utilisées pour nos expérimentations sont constituées des vidéos de DynTex. Ces bases diffèrent par leur difficulté, le nombre de classes et le nombre d'éléments :

- **Base de données Alpha** : 60 séquences d'images de textures dynamiques regroupées en 3 classes relativement simples : Mer, Herbes et Arbres.
- **Base de données Beta** : 162 séquences d'images de textures dynamiques regroupées en 10 classes : Mer, Herbes, Arbres, Drapeaux, Eau calme, Fontaines, Fumée, Escalator, Trafic routier, Rotation. Ici, plus de phénomènes complexes sont présents que dans la base de donnée Alpha.
- **Base de données Gamma** : 275 séquences d'images de textures dynamiques regroupées en 11 classes : Fleurs, Mer, Arbres sans feuillage, Feuillage dense, Escalator, Eau calme, Drapeaux, Herbes, Trafic routier, Fontaines, Feu. Dans cette base de données, les classes sont dotées de nombreux échantillons pour couvrir de nombreux cas (changement d'échelles, d'orientation). Il s'agit d'un jeu de données très complexe.

L'ensemble des séquences d'images utilisées dans ces bases sont visibles en annexe C.

5.1.4 Résultats et discussion

Le tableau 5.2 ainsi que ceux présents en annexe D montrent les taux de reconnaissance obtenus à l'aide des descripteurs précédemment évoqués.

Ces résultats d'indexation conduisent, du point de vue des descripteurs, aux constatations suivantes :

- quelque soit la méthode d'analyse des séquences d'images et la base de données utilisées, le vecteur caractéristique S_m^μ discrimine le moins bien les échantillons. Sa combinaison avec les autres descripteurs pour créer S_m^A peut donc être remise en question. Ainsi, des expérimentations où le vecteur S_m^A était construit sans S_m^μ ont été menées, et n'ont pas montré un meilleur taux de reconnaissance. Au contraire, dans certains cas, celui-ci est même dégradé.
- à la vue des quatre vecteurs caractéristiques de base (S_m^μ , S_m^σ , S_m^E et S_m^H), celui construit à partir de l'écart-type des sous-bandes de détails possède le pouvoir de discrimination le plus fort.
- la concaténation des différents vecteurs caractéristiques pour construire S_m^A est très bénéfique. En effet, excepté dans un cas, les taux de reconnaissance obtenus avec celui-ci sont les meilleurs.

Méthodes	Base	S_m^μ	S_m^σ	S_m^E	S_m^H	S_m^A
Transformée en ondelettes frame par frame	Alpha	68	82	78	88 †*	88 †*
	Beta	41 †	50	51	57	66
	Gamma	38	60	60	55	65
Transformée en ondelettes temporelle	Alpha	37	75	67	67	73
	Beta	15	43	37	28	46
	Gamma	15	34	31	26	40
Transformée en ondelettes 2D+T	Alpha	72 †	85 †	85 †	87	85
	Beta	33	62	61	65	65
	Gamma	36	65 †	64 †	61 †	68 †*
Approche de Smith <i>et al.</i> [Smith 02]	Alpha	65	83	80	82	83
	Beta	35	65	59	65	67
	Gamma	43 †	63	56	59	65
Transformée en curvelets 2D+T	Alpha	47	85 †	83	85	85
	Beta	19	65	61	62	67
	Gamma	18	62	60	56	63
Analyse en composantes morphologiques	Alpha	37	83	83	83	85
	Beta	23	68 †	64 †	66 †	70 †*
	Gamma	19	61	62	59	63

TAB. 5.2 – Taux de reconnaissance (en %) de l’indexation de textures dynamiques selon les différents descripteurs et sur trois bases de données. † représente les meilleurs taux de reconnaissance pour un descripteur sur une base de données. * représente le meilleur taux de reconnaissance pour une base de données.

Si on observe les taux de reconnaissance pour chaque base de données :

- Alpha : la méthode la plus discriminante est celle de la transformée en ondelettes frame par frame. Au regard des éléments constituant la base de données (cf. annexe C), nous constatons de manière évidente que la distinction entre les classes peut s’effectuer uniquement à partir des propriétés spatiales de chaque séquence. Dans ce cas, la dimension temporelle n’ajoute que peu d’information intéressante. Les autres méthodes d’analyse (exceptée la transformée en ondelettes temporelle) restent performantes puisqu’elles obtiennent un taux de reconnaissance proche (5% d’écart).
- Beta : pour cette base, l’approche la plus discriminante est celle fondée sur l’analyse en composantes morphologiques. Celle-ci obtient le meilleur taux de reconnaissance pour 4 des vecteurs caractéristiques proposés (S_m^σ , S_m^E , S_m^H et S_m^A).
- Gamma : la transformée en ondelettes 2D+T est la méthode obtenant les taux de reconnaissance les plus élevés pour cette base. Comme pour les autres bases de données, les différentes méthodes se démarquent peu et les taux de reconnaissance restent relativement proches.

Les mêmes tests effectués sur les bases de données Beta et Gamma conduisent à un taux de reconnaissance de 15% moins élevé que la base Alpha ; ceci est du à la complexité des classes ainsi qu’au nombre d’échantillons.

Les différentes approches multi-résolutions, exceptée la transformée en ondelettes tempo-

relle, apportent des taux de reconnaissance acceptables sur l'ensemble des bases de données. Les résultats obtenus sont toujours dans une fourchette de valeurs relativement proches, alors qu'ils sont bien inférieurs lorsque seule l'information temporelle est utilisée.

Les taux de reconnaissance obtenus sont donc satisfaisants : ils sont de l'ordre de 70% pour des bases de données de dimension relativement importante (bases Beta et Gamma). Ces performances peuvent sûrement être améliorées. Nous proposons ici plusieurs pistes de réflexion.

Lorsque l'on regarde les dimensions des vecteurs caractéristiques, fonction de la méthode multi-résolutions employée (cf. tableau 5.3), on observe que pour certaines méthodes, le nombre de descripteurs est plus grand que le nombre d'échantillons à classer. Ceci pose plusieurs difficultés :

- l'information permettant de classer les échantillons est trop redondante et peut détériorer la classification.
- le jugement sur les approches multi-échelles peut être discuté puisqu'il ne s'établit pas dans les mêmes conditions. En effet, le classement de 10 classes dans un espace à 3 dimensions n'est pas de la même difficulté que dans un espace à 5508 dimensions, et l'approche de classement utilisée n'est pas nécessairement adaptée.

Méthodes d'analyse	Nombre d'échelles	Nombre de subdivisions	Taille des vecteurs caractéristiques				
			S_m^μ	S_m^σ	S_m^E	S_m^H	S_m^A
Transformée en ondelettes frame par frame	3	×	9			36	
	4	×	12			48	
	5	×	15			60	
Transformée en ondelettes temporelle	3	×	3			12	
	4	×	4			16	
	5	×	5			20	
Transformée en ondelettes 2D+T	3	×	21			84	
	4	×	28			112	
	5	×	35			140	
Approche de Smith <i>et al.</i>	3	×	45			180	
Transformée en curvelets 2D+T	3	2	25			100	
	3	4	97			388	
	4	2	121			484	
	4	4	481			1924	
	5	2	217			868	
	5	4	865			3460	
Analyse en composantes morphologiques	×	×	1377			5508	

TAB. 5.3 – Dimension des vecteurs caractéristiques en fonction des méthodes d'analyses.

Pour corriger ce problème, nous devons réduire la dimension des signatures. Plusieurs approches sont envisageables :

- réduire la dimension des signatures à l'aide de l'analyse en composantes principales. Deux tests ont été réalisés en ramenant la dimension des signatures à 50 et 15. Toutefois, les taux de reconnaissance ont grandement chuté (de l'ordre de 30% à 40%).
- effectuer une sélection de variables à l'aide, par exemple, de la méthode *Stepwise Discriminant Analysis*. Géométriquement, il s'agit de trouver le sous-espace de représentation qui permet un écartement maximal entre les centres de gravité des nuages de points.
- modifier la façon de construire les vecteurs descripteurs afin de les rendre plus compacts et plus représentatifs de l'information spatio-temporelle. Plutôt que de calculer l'énergie de toutes les sous-bandes de détails des approches multi-échelles, nous pourrions uniquement calculer celle de chaque échelle et utiliser d'autres descripteurs pour qualifier l'information directionnelle (comme l'homogénéité de direction).

Par exemple, dans le cadre de la transformée en curvelet 2D+T, à la place d'avoir un vecteur de 3460 éléments, on aurait un ensemble de 5 éléments représentatifs des différentes échelles additionné des éléments caractéristiques de l'information directionnelle.

Dans l'objectif d'améliorer les performances de l'indexation, nous pourrions également regarder plus en détails l'invariance de nos descripteurs. En effet, à l'heure actuelle, nos vecteurs de caractéristiques sont par construction invariants en translation, mais pas en rotation et échelle. C'est un problème classiquement rencontré lors de l'utilisation de l'énergie de sous-bandes d'ondelettes dans un vecteur caractéristique.

Pour résoudre ce problème d'invariance, nous envisageons deux pistes :

- la création d'un ordre de parcours des différentes orientations pour chaque approche multi-échelles, permettant à l'aide d'une permutation circulaire, de rendre invariant en rotation la signature. Toutefois, cette méthode ne permet pas la réduction du nombre de dimensions qui est un autre de nos problèmes.
- la construction de nouveaux descripteurs comme par exemple l'homogénéité des directions, ou l'amplitude maximale intégrant directement la notion d'invariance.

Cette deuxième approche est la piste que nous souhaitons privilégier dans nos travaux futurs.

Concernant l'analyse en composantes morphologiques, les descripteurs construits sont les mêmes pour les deux composantes. Il pourrait être pertinent de construire des descripteurs plus adaptés à la nature des composantes.

Dans cette première partie de chapitre, une application ayant pour objectif l'indexation, donnant des résultats encourageants, a été proposée. Celle-ci a été expérimentée sur trois bases de textures dynamiques issues de DynTex. Dans la prochaine application, les vecteurs caractéristiques vont être utilisés avec comme objectif la recherche par le contenu.

5.2 Recherche par le contenu

L'objectif est de présenter une requête sur la base DynTex sous la forme d'une séquence d'images et d'observer celles retenues par le moteur de recherche.

Pour réaliser cette expérimentation (illustrée par la figure 5.3), nous commençons par effectuer une analyse complète de la base de données DynTex à l'aide de l'algorithme de l'analyse en composantes morphologiques (cf. chapitre 4.2). Les résultats de cette décomposition sont ensuite utilisés dans la construction du vecteur caractéristique. Ici, la signature S_m^A , détaillée dans la section 5.1, est utilisée.

Nous procédons également à l'analyse en composantes morphologiques et au calcul de la signature pour la séquence d'images requête. Ensuite, nous recherchons les n plus proches voisins de celle-ci en effectuant un calcul de distance euclidienne.

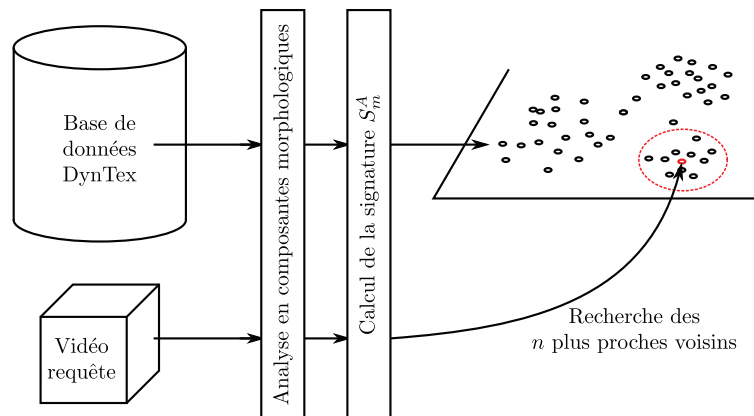


FIG. 5.3 – Approche adaptée pour la recherche de séquences d'images par le contenu.

La figure 5.4 présente les résultats de quatre requêtes sur la base DynTex. La première colonne présente la séquence d'images requête. Chaque requête est suivie des 9 séquences les plus proches.

Différentes observations peuvent être effectuées :

- Fleur : les 9 séquences d'images les plus proches sont pertinentes et représentatives de la requête. La sixième séquence peut paraître mal située, car on la placerait naturellement en troisième position. Toutefois, entre celle-ci et les premières séquences, on peut noter un changement d'échelle. Le changement de couleur ou d'arrière plan n'interfère pas dans le calcul du vecteur caractéristique.
- Herbe : pour cette requête, les meilleurs résultats sont proches de la vidéo requête, par la suite nous notons plusieurs aberrations dans les résultats. La présence des deux séquences d'images "Évier" traduit le fait que les descripteurs considérés sont relativement proches dans l'espace des caractéristiques.
- Mer : comme dans le cadre des fleurs, les 9 séquences résultats sont proches de la séquence requête. On constate en effet que l'ensemble des textures dynamiques sélectionné se res-

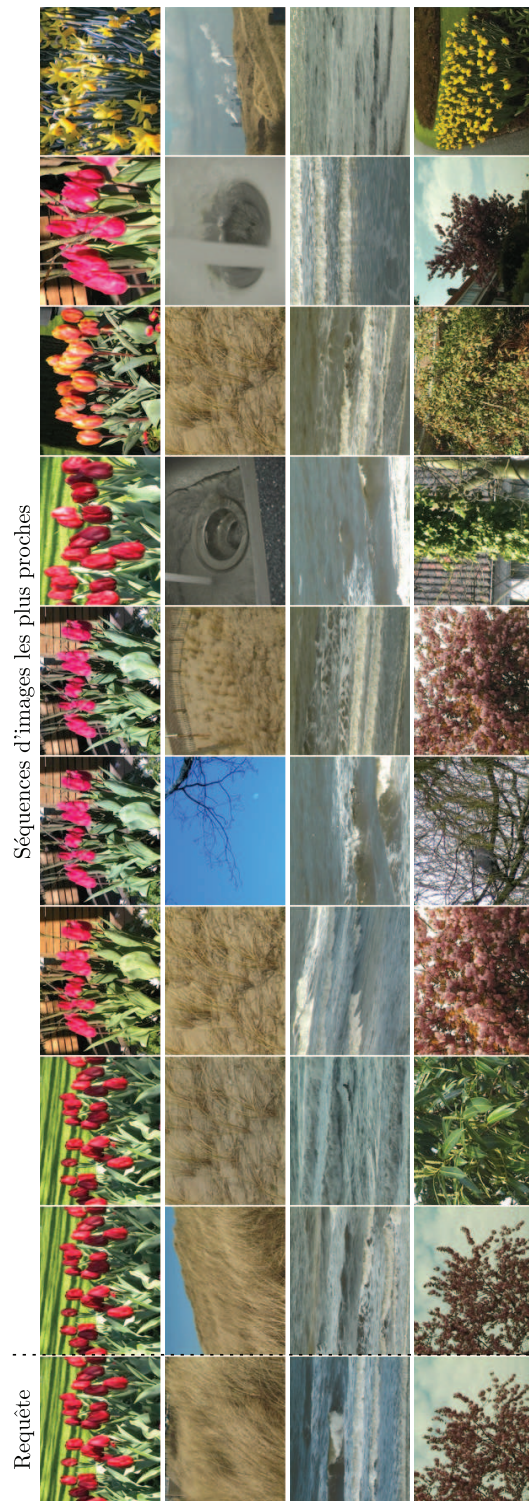


FIG. 5.4 – Résultats d'une recherche par le contenu à l'aide d'une requête sous forme de séquence d'images.

semble du point de vue des caractéristiques spatio-temporelles.

- Arbre : les résultats obtenus par la recherche correspondent à des séquences d’images de feuillage exceptées pour deux d’entre elles : la quatrième et la neuvième séquence. Cependant, la structure de branches sans feuillage, et le massif de fleurs possèdent des caractéristiques spatio-temporelles proches de la séquence requête. Nous observons également que l’arbre de la séquence d’images requête se retrouve présent à différentes échelles dans les séquences résultats.

Nous avons présenté ici la pertinence de l’utilisation des approches de décomposition dans le contexte de l’indexation de séquences d’images par le contenu. Nous nous appuyons uniquement sur une mesure de distance entre la signature de la séquence requête et celles des séquences de la base sans information de couleur, souvent discriminante. L’utilisation des approches du type bouclage de pertinence [Smeulders 00, Zhou 03] permettrait de prendre en compte le retour de l’utilisateur.

5.3 Estimation du mouvement global

La décomposition d’une texture dynamique en une composante géométrique et une composante texture via l’analyse en composantes morphologiques apporte une meilleure visualisation et une meilleure compréhension des phénomènes présents dans la séquence. Nous proposons une application mettant en avant cette propriété. L’objectif est l’estimation du mouvement global présent dans une séquence d’images.

Nous proposons l’approche suivante :

- décomposition de la texture dynamique à l’aide de l’algorithme de l’analyse en composantes morphologiques,
- calcul du flot optique sur la composante géométrique à l’aide de l’algorithme de Horn et Schunk [Horn 81] avec un paramètre de régularisation $\lambda = 10$.

Le mouvement global ainsi estimé est comparé (comme l’illustre la figure 5.5) à l’estimation du mouvement de la séquence originale à l’aide de l’algorithme de Horn et Schunk [Horn 81] avec deux paramètres de régularisation différents : $\lambda = 10$ (faible) et $\lambda = 50$ (fort). Pour cela, différents critères sont utilisés.

Les résultats du flot optique estimé (présentés sur la figure 5.6) sont illustrés sur une séquence de mer présentant des vagues et de l’écume. Les différents flots optiques sont représentés par un champ de vecteurs coloré où la teinte (respectivement la saturation) indique la direction (respectivement l’intensité) du flot optique.

A la vue des différents champs de vecteurs, nous remarquons que le mouvement principal de la texture dynamique est mieux détectée sur la composante géométrique de l’analyse en composante morphologique que sur la séquence d’images originale. Dans cette dernière, les directions et les amplitudes des vecteurs du mouvement sont plus en accord avec les observations effectuées sur la séquence d’images originale.

Nous remarquons également que le flot estimé à l’aide d’un paramètre de régularisation faible ($\lambda = 10$) directement sur la séquence d’images est très perturbé. En effet, aucune couleur ne

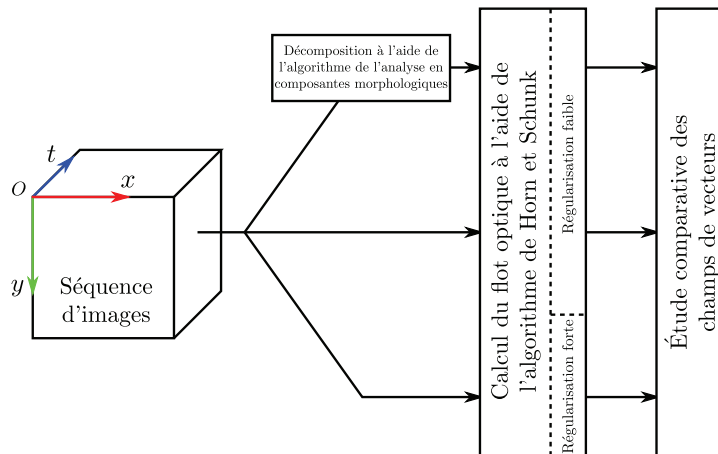


FIG. 5.5 – Méthode de comparaison entre l'estimation du mouvement global sur la composante géométrique et l'estimation du mouvement sur la séquence originale.

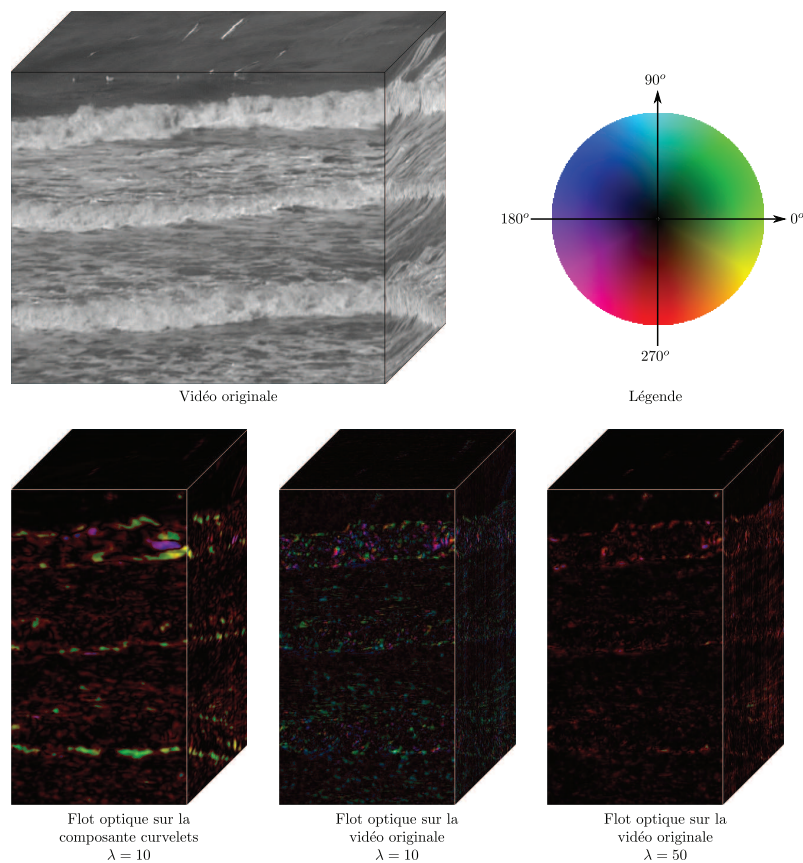


FIG. 5.6 – ♠ Flots optiques estimés sur la vidéo originale et sur la composante géométrique de l'analyse en composantes morphologiques.

se démarque des autres, les mouvements estimés sont chaotiques, dus principalement à l'écume turbulente et sans direction privilégiée.

Le renforcement de la régularisation ne permet pas de corriger ce défaut. Le flot optique estimé (en prenant $\lambda = 50$) est un peu meilleur que le précédent mais la détection du mouvement des vagues n'est pas encore satisfaisante. De plus, la régularisation du champ a pour effet de lisser les frontières de mouvement, ce qui n'est pas le cas avec l'approche proposée.

Afin de confirmer ces observations, nous proposons deux autres critères d'études :

- un critère d'homogénéité des directions (cf. figure 5.7). Pour un ensemble Ω de vecteurs \vec{v}_i , deux informations sont calculées :

- la direction principale

$$V = \sum_{i \in \Omega} \vec{v}_i \quad (5.14)$$

- l'homogénéité du champ de vecteur

$$\phi = \frac{\|\sum_{i \in \Omega} \vec{v}_i\|}{\sum_{i \in \Omega} \|\vec{v}_i\|} \quad (5.15)$$

Ce critère est ensuite représenté par un triangle isocèle pointant dans la direction V et dont la base est de longueur ϕ .

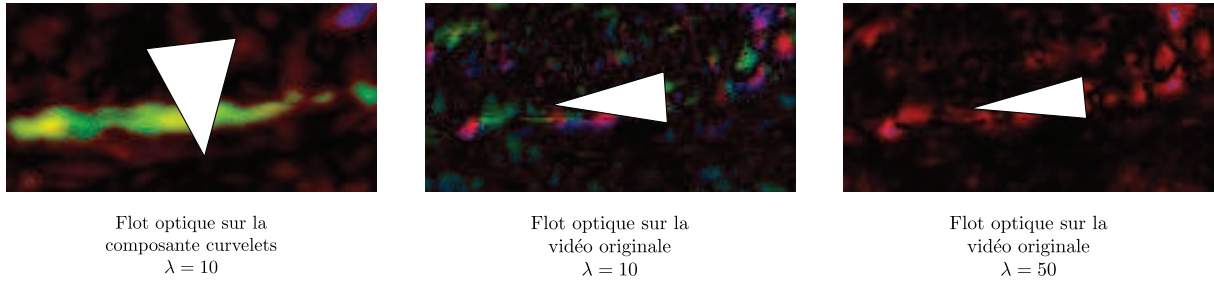


FIG. 5.7 – Critère d'homogénéité pour une zone d'intérêt du champ de vecteurs mouvement représenté dans la figure 5.6.

- un diagramme des orientations (cf. figure 5.8) où nous comptabilisons pour chaque direction le nombre de vecteurs.

Ces deux critères confirment les résultats précédents. Le mouvement détecté, en utilisant la composante géométrique de la décomposition, est plus pertinent. En effet, le champ est beaucoup plus homogène ($\phi = 0.79$) que les deux autres ($\phi = 0.37$ et $\phi = 0.45$). De plus la direction donnée par le diagramme des directions ou par le critère d'homogénéité est également plus représentatif du mouvement observé.

La composante géométrique de l'analyse en composantes morphologique permet ainsi une meilleure extraction des phénomènes sous-jacents à la texture dynamique.

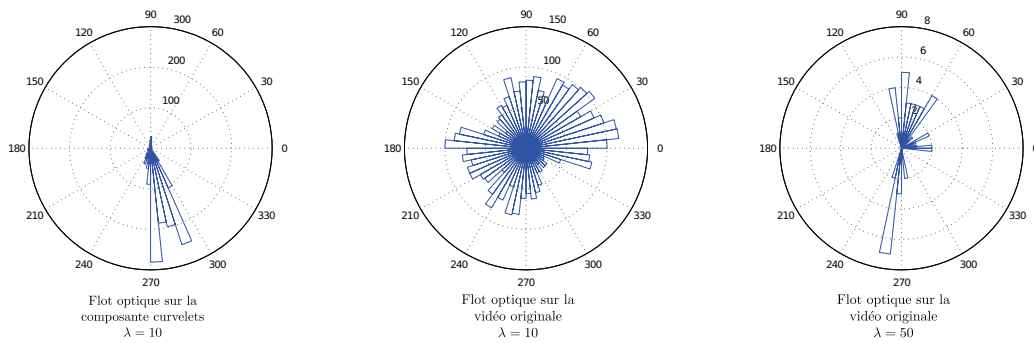


FIG. 5.8 – Diagramme d'orientations pour l'ensemble de la séquence d'images des vecteurs mouvement présentés sur la figure 5.6.

Conclusion du chapitre Dans la première partie de ce chapitre, une première application d'indexation a été proposée. Par celle-ci, nous avons montré que l'analyse en composantes morphologiques conduisait à des taux de reconnaissance intéressants et prometteurs. Cette méthode peut être améliorée en construisant d'autres descripteurs plus représentatifs de l'information extraite.

Dans une deuxième partie, une approche de recherche par le contenu a été proposée. L'objectif est de retrouver un ensemble de séquences d'images proche de la séquence requête. Cette approche offre de bon taux de réussite.

Enfin dans la dernière partie, la composante géométrique de l'analyse en composantes morphologiques est utilisée avec succès pour une estimation du mouvement global de la texture dynamique.

Conclusion générale et perspectives

Au cours de cette thèse, nos travaux de recherche ont porté sur l'analyse et la caractérisation des textures dynamiques à l'aide de décompositions multi-échelles spatio-temporelles.

Dans une première partie, après discussion sur les définitions des textures dynamiques existantes dans la littérature, nous avons établi une taxonomie de celles-ci. Nous avons ensuite proposé une définition des textures dynamiques. Ainsi, les séquences d'images (naturelles, artificielles ou synthétiques) sont constituées d'une composante texture statique et/ou d'une composante texture dynamique. Cette dernière est composée d'au moins une texture dynamique.

Une texture dynamique est une structure texturée (rigide ou déformable) possédant un mouvement (stochastique ou déterministe) induit par une force (interne ou externe) ou par le déplacement de la caméra. De plus, elle peut être constituée de plusieurs modes, pouvant se superposer, caractérisés par des phénomènes spatiaux et temporels répétitifs.

Afin de situer notre travail, une synthèse originale des méthodes de la littérature permettant la caractérisation des textures dynamiques a ensuite été effectuée. Nous avons établi qu'il existait principalement quatre types d'approches reposant sur : le mouvement ; les propriétés géométriques de l'espace spatio-temporel ; des transformées spatio-temporelles ; des modèles mathématiques.

La prise en compte du comportement multi-échelles des textures dynamiques nous a semblé être un élément crucial pour les caractériser. Nous nous sommes donc tournés vers les méthodes multi-échelles étendues au domaine temporel. Constatant que la littérature présentait une seule transformée spatio-temporelle multi-échelles (issue de l'extension du cas mono-dimensionnel de la transformée en ondelettes) pour caractériser les textures dynamiques, nous en avons présenté trois autres dans le chapitre 2 de ce manuscrit. Ces approches multi-échelles diffèrent par la manière dont sont considérées les variables spatiales et temporelles. Nous avons ensuite souligné les défauts rencontrés par ce type de transformée (mauvaise représentation des structures anisotropiques 2D+T) et proposé une approche géométrique permettant la représentation creuse de structures de co-dimension 1 : la transformée en curvelets 2D+T. A cette occasion, nous avons répondu au problème délicat de visualisation des coefficients de cette transformée à l'aide d'une méthode originale : les cartes de Mercator.

Le troisième chapitre formalise tout d'abord un modèle de textures dynamiques. Celui-ci définit une classe importante de textures dynamiques comme étant la somme d'une ou plusieurs ondes porteuses et de phénomènes locaux. Pour évaluer la pertinence de ce modèle, deux méthodes de synthèse ont été mises en place. La première consiste à paramétrer le modèle avec des coefficients arbitraires. Les résultats ainsi obtenus ont permis la validation du modèle. Une deuxième approche d'évaluation a été exposée. Elle repose sur l'estimation des paramètres du modèle à l'aide d'un algorithme d'optimisation par essaim particulaire. Toutefois, par manque de temps, cette méthode a permis l'obtention de résultats prometteurs mais pour des modèles encore trop restrictifs. L'aboutissement de cette approche pour évaluer quantitativement notre modèle constitue notre première perspective.

Nous avons poursuivi notre étude en présentant l'étude d'une des composantes du modèle : l'onde porteuse. L'utilisation conjointe de la transformée en curvelets 2D+T et d'une structure en octree a permis de :

- montrer que la transformée en curvelets 2D+T est adaptée à la représentation de la composante onde porteuse de notre modèle.
- mettre en place une méthode efficace de segmentation spatio-temporelle des textures dynamiques.

En s'appuyant sur le modèle proposé, le quatrième chapitre, étudie la décomposition de séquences de textures dynamiques. Après une introduction des méthodes de décomposition de la littérature, nous nous sommes focalisés sur l'analyse en composantes morphologiques. Cette approche permet l'utilisation d'un dictionnaire d'atomes riches ainsi qu'un algorithme modulaire. Toutefois, l'extension au domaine temporel de l'algorithme de l'analyse en composantes morphologiques s'avère extrêmement coûteuse en temps de calcul.

C'est pour palier à ce problème que nous avons élaboré deux nouvelles stratégies de seuillage adaptatives permettant l'accélération de la convergence de l'algorithme. Ces deux stratégies ont été validées par plusieurs critères quantitatifs.

Une extension naturelle de ce travail serait de reprendre le modèle proposé et d'y ajouter de nouvelles composantes permettant la caractérisation d'une autre classe de textures dynamiques. Ainsi, des bases représentatives de ces composantes seraient par la suite ajoutées au dictionnaire de l'analyse en composantes morphologiques.

Dans cette idée, une thèse a été lancée entre le laboratoire MIA de La Rochelle et le laboratoire MAPMO d'Orléans afin de décomposer des volumes 3D de données IRM. L'approche présentée dans ce manuscrit est privilégiée, en étudiant l'apport de la transformée en beamlets 3D qui permettra de représenter de manière creuse les structures filamenteuses.

Le dernier chapitre a été consacré à diverses applications :

- l'indexation vidéo : cette application met en place l'ensemble des outils d'analyse 2D+T étudié au cours de cette thèse dans un contexte d'indexation de trois bases de séquences de textures dynamiques. Ces bases sont de complexités différentes (60 échantillons pour 3 classes, 162 échantillons pour 10 classes et 275 échantillons pour 11 classes). Les vecteurs caractéristiques obtenus sur les composantes de l'analyse en composantes morphologiques conduisent à des taux de reconnaissance intéressants.

Cette application ouvre de nombreuses pistes de travail :

- réduction de la taille des signatures : les vecteurs caractéristiques de certaines transformées sont de trop grandes dimensions ce qui peut pénaliser l'indexation. Pour répondre à ce problème, nous proposons de nouveaux descripteurs permettant de caractériser différemment l'information issus des transformées multi-échelles. En effet, plutôt que de calculer l'énergie de toutes les sous-bandes de détails des approches multi-échelles, nous pourrions uniquement calculer celle de chaque échelle et utiliser d'autres descripteurs permettant de qualifier l'information directionnelle (comme l'homogénéité de direction).
- la construction de vecteurs caractéristiques invariants : que ce soit dans le cadre de l'indexation ou d'autres applications utilisant des vecteurs caractéristiques, il est important de rendre ces derniers invariant par rotation et échelle.
- la construction de nouveaux descripteurs : dans le cadre de l'analyse en composantes

morphologiques, les vecteurs caractéristiques calculés sont les mêmes pour les deux composantes. Cependant l'information extraite par chacune d'elle n'est pas de même nature. Il peut donc être intéressant de concevoir des descripteurs basés sur les caractéristiques de chaque composante, et par ce biais les propriétés propres de notre modèle.

- le recherche par le contenu dans des séquences : l'application présentée à la fin de cette thèse avait pour but d'extraire d'une base de données les séquences les plus similaires par rapport à une séquence requête. Cette recherche est basée sur les vecteurs caractéristiques issus de l'analyse en composantes morphologiques et utilise la norme euclidienne. Les exemples que nous avons développés montrent la pertinence de notre approche. Une perspective pour cette application est la mise en place d'approches de type bouclage de pertinence. Ceci peut permettre d'analyser les descripteurs afin de comprendre et d'apprendre ce qui du point de vue de l'utilisateur caractérise les textures dynamiques. Une autre perspective de ce bouclage de pertinence est aussi de pouvoir effectuer des requêtes plus sémantiques, par exemple rechercher des séquences de mer "calme" ou d'arbres "fortement agités".
- estimation du mouvement global d'une texture dynamique : la dernière application présentée utilise le résultat de l'analyse en composantes morphologiques pour mieux interpréter les textures dynamiques. Une estimation du mouvement a été effectuée sur la composante caractérisant l'onde porteuse et sur la séquence originale. Nous avons remarqué que le mouvement global de la texture dynamique était beaucoup mieux estimé dans le premier cas que dans le deuxième. Des applications assez proches de type débruitage ou compression peuvent être envisagées à court terme.

Des vecteurs caractéristiques robustes peuvent aussi être utilisés dans l'objectif de construire des résumés vidéo, permettant ainsi d'étiqueter la séquence d'images de manière spatio-temporelle (début d'un feu à l'instant t , ...).

L'ensemble des outils d'analyse spatio-temporelle proposés dans ce manuscrit ne se restreignent pas qu'aux textures dynamiques. Ils peuvent s'appliquer à toutes les séquences d'images et également aux données tri-dimensionnelles (imagerie médicales 3D, échographie pour l'étude des sols, ...).

L'étude des textures dynamiques est encore un domaine de recherche jeune, et nous pensons qu'il continuera à croître fortement dans les prochaines années. Prendre en compte à la fois la complexité et la grande variabilité de ces textures dynamiques est en effet un défi difficile. Le développement d'outils avancés permettant de progresser dans la compréhension et l'analyse de ces textures dynamiques est un enjeu important : les domaines applicatifs où interviennent les textures dynamiques étant en effet abondants et sont amenés à croître avec le futur.

Littérature des textures dynamiques

Chen et al.	A Novel Frame Error Concealment Algorithm Based on Dynamic Texture Synthesis	2010
Chetverikov et al.	Dynamic Texture as Foreground and Background	2010
Derpanis and Wildes	Dynamic Texture Recognition based on Distributions of Spacetime Oriented Structure	2010
Péteri	Tracking Dynamic Textures using a Particle Filter Driven by Intrinsic Motion Information	2010
Péteri et al.	DynTex: A Comprehensive Database of Dynamic Textures	2010
Traver et al.	Fast Dynamic Texture Detection	2010
Ghanem and Ahuja	Sparse Coding of Linear Dynamical Systems with an Application to Dynamic Texture Recognition	2010
Dubois et al.	Decomposition of Dynamic Textures using Morphological Component Analysis: A New Adaptive Strategy	2010
Chan and Vasconcelos	Variational Layered Dynamic Textures	2009
Crivelli et al.	Learning Mixed-State Markov Models for Statistical Motion Texture Tracking	2009
Dubois et al.	A Comparison of Wavelet Based Spatio-temporal Decomposition Methods for Dynamic Texture Recognition	2009
Dubois et al.	A 3D Discrete Curvelet based Method for Segmenting Dynamic Textures	2009
Fazekas et al.	Dynamic Texture Detection Based on Motion Analysis	2009
Kim et al.	A Model Change Detection Approach to Dynamic Scene Modeling	2009
Kobayashi et al.	Recognition of Dynamic Texture Patterns Using CHLAC Features	2009
Li et al.	Dynamic Texture Segmentation Using Fourier Transform	2009
Li et al.	Dynamic Texture Segmentation Using 3-D Fourier Transform	2009
Ma and Cisar	Event Detection using Local Binary Pattern based Dynamic Textures	2009
Rahman and Murshed	Detection of Multiple Dynamic Textures Using Feature Space Mapping	2009
Ravichandran et al.	View-Invariant Dynamic Texture Recognition using a Bag of Dynamical Systems	2009
Zhang and Sun	Dynamic Textures Retrieval with Surfacelet Transform	2009
Zhou et al.	Non-Linear Dynamic Texture Analysis and Synthesis Using Constrained Gaussian Process Latent Variable Model	2009
Chan and Vasconcelos	Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures	2008
Costantini et al.	Higher Order SVD Analysis for Dynamic Texture Synthesis	2008
Doretto and Soatto	From Dynamic Texture to Dynamic Shape and Appearance Models: An Overview	2008
Gao and Xu	Extended Statistical Landscape Features for Dynamic Texture Recognition	2008
Ghanem and Ahuja	Extracting a Fluid Dynamic Texture and the Background from Video	2008
Li and Hall	Learning A Stable Structure To Describe Dynamic Texture	2008
Peyré	Dynamic Texture Synthesis with Grouplets	2008
Rahman and Murshed	Temporal Texture Characterization: A Review	2008
Ravichandran and Vidal	Video Registration Using Dynamic Textures	2008
Zhu et al.	Video Coding with Spatio-Temporal Texture Synthesis and Edge-based Inpainting	2008
Amiaz et al.	Detecting Regions of Dynamic Texture	2007
Chan and Vasconcelos	Classifying Video with Kernel Dynamic Textures	2007
Fazekas and Chetverikov	Analysis and Performance Evaluation of Optical Flow Features for Dynamic Texture Recognition	2007
Fazekas and Chetverikov	Dynamic Texture Recognition Using Optical Flow Features and Temporal Periodicity	2007
Ferrari et al.	Real-Time Detection of Steam in Video Images	2007
Ghanem and Ahuja	Phase Based Modelling of Dynamic Textures	2007
Ghanem and Ahuja	Phase PCA for Dynamic Texture Video Compression	2007
Huang et al.	Optimization and Learning for Registration of Moving Dynamic Textures	2007
Hyndman et al.	Higher-Order Autoregressive Models for Dynamic Textures	2007
Rahman and Murshed	Multiple Temporal Texture Detection using Feature Space Mapping	2007
Rahman and Murshed	A Temporal Texture Characterization Technique Using Block-Based Approximated Motion Measure	2007
Toreyin et al.	Dynamic Texture Detection, Segmentation and Analysis	2007
Vishwanathan et al.	Binet-Cauchy Kernels on Dynamical Systems and its Application to the Analysis of Dynamic Scenes	2007
Zhao and Pietikäinen	Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions	2007
Zhao and Pietikäinen	Dynamic Texture Recognition Using Volume Local Binary Patterns	2007
Zhu et al.	Video Coding with Spatio-Temporal Texture Synthesis	2007
Zhu et al.	Spatial and Temporal Models for Texture-based Video Coding	2007
Bouthemy et al.	Mixed-State Auto-Models and Motion Texture Modeling	2006
Chan and Vasconcelos	Layered Dynamic Textures	2006
Chetverikov and Fazekas	On Motion Periodicity of Dynamic Textures	2006
Cooper et al.	Spatial Segmentation of Temporal Texture Using Mixture Linear Models	2006
Costantini et al.	Dynamic Texture Analysis and Synthesis Using Tensor Decomposition	2006
Doretto and Soatto	Modeling Dynamic Scenes: An Overview of Dynamic Textures	2006
Doretto and Soatto	Dynamic Shape and Appearance Models	2006
Filip et al.	Fast Synthesis of Dynamic Colour Textures	2006
Ghoreyshi and Vidal	Segmenting Dynamic Textures with Ising Descriptors, ARX Models and Level Sets	2006
Liu et al.	Dynamic Textures Synthesis as Nonlinear Manifold Learning and Traversing	2006
Woolfe and Fitzgibbon	Shift-Invariant Dynamic Texture Recognition	2006
Zhao and Pietikäinen	Local Binary Pattern Descriptors for Dynamic Texture Recognition	2006
Abraham et al.	Dynamic Texture with Fourier Descriptors	2005
Chan and Vasconcelos	Mixtures of Dynamic Textures	2005
Chan and Vasconcelos	The EM Algorithm for Layered Dynamic Textures	2005
Chan and Vasconcelos	Probabilistic Kernels for the Classification of Auto-Regressive Visual Processes	2005
Chetverikov and Peteri	A Brief Survey of Dynamic Texture Description And Recognition	2005
Dedeoglu et al.	Real-Time Fire and Flame Detection in Video	2005
Doretto	Modeling Dynamic Scenes with Active Appearance	2005
Fazekas and Chetverikov	Normal Versus Complete Flow in Dynamic Texture Recognition: a Comparative Study	2005
Kumar et al.	Spatio-Temporal Texture Synthesis and Image Inpainting for Video Applications	2005
Liu et al.	Modeling Dynamic Textures Using Subspace Mixtures	2005

Lu et al.	Dynamic Texture Recognition by Spatio-Temporal Multiresolution Histograms	2005
Pashley et al.	Motion Texture	2005
Péteri and Chetverikov	Dynamic Texture Recognition Using Normal Flow and Texture Regularity	2005
Rahman and Murshed	A Robust Optical Flow Estimation Algorithm for Temporal Textures	2005
Vidal and a. Ravichandran	Optical Flow Estimation and Segmentation of Multiple Moving Dynamic Textures	2005
Doretto et al.	Spatially Homogeneous Dynamic Textures	2004
Péteri and Chetverikov	Qualitative Characterization of Dynamic Textures for Video Retrieval	2004
Rahman et al.	Feature Weighting Methods for Abstract Features Applicable to Motion based Video Indexing	2004
Yuan et al.	Synthesizing Dynamic Texture with Closed-Loop Linear Dynamic System	2004
Doretto et al.	Dynamic Textures	2003
Doretto et al.	Dynamic Texture Segmentation	2003
Doretto and Soatto	Editable Dynamic Textures	2003
Doretto and Soatto	Towards Plenoptic Dynamic Textures	2003
Edwards et al.	Motion Field Estimation for Temporal Textures	2003
Fablet and Boutheymy	Motion Recognition Using Nonparametric Image Motion Models Estimated from Temporal and Multiscale Cooccurrence Statistics	2003
Fujita and Nayar	Recognition of Dynamic Textures using Impulse Responses of State Variables	2003
Kwatra et al.	Graphcut Textures: Image and Video Synthesis Using Graph Cuts	2003
Wang and Zhu	Modeling Textured Motion: Particle, wave and Sketch	2003
Doretto	Dynamic Texture Modeling	2002
Fablet et al.	Nonparametric Motion Characterization using Causal Probabilistic Models for Video Indexing and Retrieval	2002
Peh and Cheong	Synergizing Spatial and Temporal Texture	2002
Phillips et al.	Flame Recognition in Video	2002
Smith et al.	Video Texture Indexing using Spatio-Temporal Wavelets	2002
Y.Ma and Zhang	Motion Texture: A New Motion Based Video Representation	2002
Zhong and Scarlaroff	Temporal Texture Recognition Model Using 3D Features	2002
Joseph et al.	Texture Mixing and Texture Movie Synthesis using Statistical Learning	2001
Fablet and Boutheymy	Motion Recognition Using Spatio-temporal Random Walks in Sequence of 2D Motion-related Measurements	2001
Fablet and Boutheymy	Non Parametric Motion Recognition using Temporal Multiscale Gibbs Models	2001
Saisan et al.	Dynamic Texture Recognition	2001
Soatto et al.	Dynamic Textures	2001
Ardizzone et al.	Using Temporal Texture for Content-Based Video Retrieval	2000
Chetverikov	Pattern Regularity as a Visual Key	2000
Fablet et al.	Statistical Motion-based Video Indexing and Retrieval	2000
Schodl et al.	Video Textures	2000
C.H Peh and Cheong	Exploring Video Content in Extended Spatio-Temporal Textures	1999
Fablet and Boutheymy	Motion-Based Feature Extraction and Ascendant Hierarchical Classification for Video Indexing and Retrieval	1999
Boutheymy and Fablet	Motion characterization from temporal cooccurrences of local motion-based measures for video indexing	1998
Otsuka et al.	Feature Extraction of Temporal Texture Based on Spatiotemporal Motion Trajectory	1998
Liu	Modeling Spatial and Temporal Textures	1997
Polana and Nelson	Temporal Texture and Activity Recognition	1997
Szumner and Picard	Temporal Texture Modeling	1996
Zhu et al.	Qualitative Estimations of Range and Motion using Spatio-Temporal Textural Images	1994
Rao and Lohse	Identifying High Level Features of Texture Perception	1993
Nelson and Polana	Qualitative Recognition of Motion using Temporal Texture	1992
Polana and Nelson	Recognition of Motion from Temporal Texture	1992

Évaluation quantitative du modèle de textures dynamiques

Pour juger de la qualité de la séquence d'images synthétisée par rapport à la vidéo originale, nous proposons d'estimer l'erreur en norme ℓ_2 . La difficulté vient de l'identification des composants du modèle : les paramètres des exponentielles complexes constituant l'onde porteuse, la position de chaque phénomène local, les paramètres des exponentielles complexes le constituant. Pour l'estimation de l'ensemble de ces paramètres, nous proposons l'utilisation d'un algorithme d'Optimisation par Essaims Particulaires (OEP).

L'Optimisation par Essaims Particulaires (OEP) est une méta-heuristique proposée par James Kennedy et Russel Eberhart [Kennedy 95]. Cette méthode possède des similitudes avec l'algorithme des colonies de fourmis, notamment dans l'intervention d'agents communiquant entre eux de manière simple et conduisant à l'émergence de comportements complexes.

La méthode en elle-même met en jeu de larges groupes de particules (appelés essaims) se présentant sous forme de vecteurs se déplaçant dans l'espace de recherche. Dans notre cas, celui-ci est constitué des paramètres du modèle (amplitude, pulsation et déphasage de la porteuse \mathcal{P} , amplitude, pulsation, déphasage et paramètres du noyau gaussien spatio-temporel des phénomènes locaux \mathcal{L}). Pour éviter que l'espace de recherche ne soit trop grand, le nombre de phénomènes locaux recherché est fixé.

Un essaim est composé de plusieurs particules, sa taille peut être dynamique : si aucune particule de l'essaim n'améliore la solution, on peut en générer d'autres et on peut également en supprimer. Chaque particule i est caractérisée par :

- sa position \mathbf{X}_i dans l'espace d'état
- un vecteur de déplacement \mathbf{D}_i

A chaque itération k , la particule se déplace selon l'équation :

$$\mathbf{X}_i(k) = \mathbf{X}_i(k-1) + \mathbf{D}_i(k) \tag{B.1}$$

Le coeur de la méthode consiste donc à choisir comment définir $\mathbf{D}_i(k)$ à chaque itération. Il faut tenir compte, pour chaque particule, de sa position \mathbf{X}_i , de la direction de son mouvement, de sa meilleure position précédente \mathbf{P}_1 et de la meilleure position parmi l'essaim \mathbf{P}_2 . Par meilleure position, nous entendons que la reconstruction, engendrée par les paramètres du modèle, désignés par les particules de l'essaim, est meilleure.

Pour réaliser son déplacement, chaque particule combine trois tendances :

- suivre son propre déplacement : $a.\mathbf{D}_i(k-1)$ avec a un coefficient d'inertie.
- revenir vers sa meilleure performance : $b_1.r_1.(\mathbf{P}_1 - \mathbf{X}_i(k-1))$ avec b_1 un coefficient d'attraction et r_1 une valeur aléatoire de distribution uniforme sur $[0; 1]$.
- aller vers la meilleure performance des autres particules de l'essaim : $b_2.r_2.(\mathbf{P}_2 - \mathbf{X}_i(k-1))$ avec b_2 un coefficient d'attraction et r_2 une valeur aléatoire de distribution uniforme sur $[0; 1]$.

Ce déplacement se formalise donc par :

$$\mathbf{D}_i(k) = a.\mathbf{D}_i(k-1) + b_1.r_1.(\mathbf{P}_1 - \mathbf{X}_i(k-1)) + b_2.r_2.(\mathbf{P}_2 - \mathbf{X}_i(k-1)) \quad (\text{B.2})$$

L'évolution de l'essaim s'effectuant selon des critères de meilleures positions, il est primordiale de définir ce concept. La position \mathbf{X}_i d'une particule représentant l'ensemble des paramètres de notre modèle (amplitudes, pulsations et déphasage de l'onde porteuse, et les caractéristiques du noyau gaussien spatio-temporel, amplitudes, pulsations et déphasage des phénomènes locaux), nous l'utilisons pour générer la séquence de synthèse. Ainsi une particule sera jugée meilleure qu'une autre si elle minimise l'erreur en norme ℓ_2 entre la séquence d'images synthétisée et l'originale.

Nous proposons l'algorithme 6 pour trouver la reconstruction de la séquence d'images au sens de la norme ℓ_2 .

Cet algorithme a été mis en place avec succès, mais les temps de calcul sont très longs. De plus, les paramètres propres à l'algorithme (le nombre de particules, le coefficient d'inertie et les coefficients d'attraction) doivent être choisis judicieusement pour une bonne évolution de l'essaim.

A l'heure actuelle, les premiers résultats obtenus sont prometteurs, malheureusement, ceux-ci ont été calculés sur des petits formats de vidéo et donc peu exploitable.

Algorithme 6 Algorithme d'optimisation par essaim particulaire pour déterminer les paramètres du modèle de texture dynamique

Tache : Identification des paramètres de synthèse par rapport à une séquence originale selon le modèle de textures dynamiques de la définition 8.

Paramètres :

- f séquence originale
- σ écart-type du bruit estimé de f
- N nombre de particules de l'essaim
- a coefficient d'inertie
- b_1 et b_2 coefficient d'attraction des meilleures particules

Initialisation :

// Pour chaque particule

pour $i = 1$ à N **faire**

// Position et déplacement aléatoires dans l'espace de recherche

$\mathbf{X}_i(0) = \text{rand};$

$\mathbf{D}_i(0) = \text{rand};$

// Indice de sa meilleure position

$\mathbf{K}_i = 0;$

// Génération de la synthèse

$\mathcal{S}_i(0) = \text{synthese.texture.dynamique}(\mathbf{X}_i(0));$

fin pour

// Indice de la particule offrant la meilleure synthèse

$M = \underset{i}{\text{argmin}}(\|f - \mathcal{S}_i(0)\|_2);$

// Itérateur permettant de suivre l'évolution de l'essaim

$k = 1;$

Boucle principale :

// Tant que la meilleure synthèse n'est pas ressemblante à la séquence originale

tant que $\|f - \mathcal{S}_M(\mathbf{K}_M)\|_2 < \sigma^2$ **faire**

// Pour chaque particule

pour $i = 1$ à N **faire**

// Tirage de deux valeurs aléatoires entre 0 et 1

$r_1 = \text{rand}(0, 1); r_2 = \text{rand}(0, 1);$

// Calcul du déplacement

$\mathbf{D}_i(k) = a \cdot \mathbf{D}_i(k-1) + b_1 \cdot r_1 \cdot (\mathbf{X}_i(\mathbf{K}_i) - \mathbf{X}_i(k-1)) + b_2 \cdot r_2 \cdot (\mathbf{X}_M(\mathbf{K}_M) - \mathbf{X}_i(k-1));$

// Mise à jour de la position

$\mathbf{X}_i(k) = \mathbf{X}_i(k-1) + \mathbf{D}_i(k);$

// Génération de la synthèse

$\mathcal{S}_i(k) = \text{synthese.texture.dynamique}(\mathbf{X}_i(k));$

// Mise à jour de la meilleure position de cette particule

$\mathbf{K}_i = \underset{k}{\text{argmin}}(\|f - \mathcal{S}_i(k)\|_2)$

fin pour

// Mise à jour de l'indice de la particule offrant la meilleure synthèse

$M = \underset{i}{\text{argmin}}(\|f - \mathcal{S}_i(\mathbf{K}_i)\|_2);$

// Passage à l'itération suivante

$k = k + 1;$

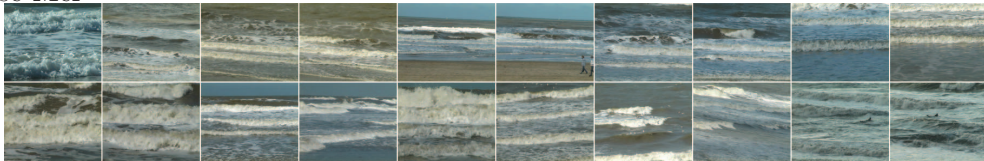
fin tant que

Bases de données expérimentales

Les trois bases de données utilisées dans nos expérimentations sont détaillées ici.

– **Base de données Alpha** : 60 textures dynamiques réparties sur 3 classes.

– Classe Mer



– Classe Herbes



– Classe Arbres

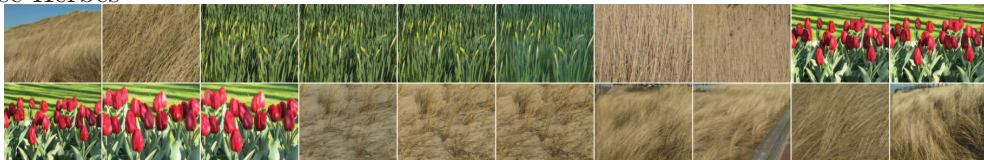


– **Base de données Beta** : 162 textures dynamiques réparties sur 10 classes.

– Classe Mer



– Classe Herbes



– Classe Arbres



– Classe Drapeaux



– Classe Eau calme



– Classe Fontaines



– Classe Fumée



– Classe Escalator



– Classe Trafic routier



– Classe Rotation



– **Base de données Gamma** : 275 textures dynamiques réparties sur 10 classes.

– Classe Fleurs



– Classe Mer



– Classe Arbres sans feuillage



– Classe Feuillage



– Classe Escalator



– Classe Eau calme



– Classe Drapeaux



- Classe Herbes



- Classe Trafic routier



- Classe Fontaines



Compléments résultats indexation

Le calcul des taux de reconnaissance sur les bases Alpha, Beta et Gamma (cf. annexe C) a été calculé selon plusieurs paramètres. Nous présentons l'ensemble des résultats dans cette annexe.

Quatre normalisations différentes ont été étudiées :

- sans normalisation.
- normalisation centrée et réduite :

$$\forall n, \forall r, \forall j, \mathbf{S}_{m,r,j}^d(n) = \frac{\mathbf{S}_{m,r,j}^d(n) - \mu_{c,i}(\mathbf{S}_{m,c,i}^d(n))}{\sigma_{c,i}(\mathbf{S}_{m,c,i}^d(n))} \quad (\text{D.1})$$

- normalisation linéaire pour chaque descripteur :

$$\forall n, \forall r, \forall j, \mathbf{S}_{m,r,j}^d(n) = \frac{\mathbf{S}_{m,r,j}^d(n) - \min_{c,i} \mathbf{S}_{m,c,i}^d(n)}{\max_{c,i} \mathbf{S}_{m,c,i}^d(n) - \min_{c,i} \mathbf{S}_{m,c,i}^d(n)} \quad (\text{D.2})$$

- normalisation linéaire pour chaque type de descripteur :

$$\forall n, \forall r, \forall j, \mathbf{S}_{m,r,j}^d(n) = \frac{\mathbf{S}_{m,r,j}^d(n) - \min_{d,c,i} \mathbf{S}_{m,c,i}^d(n)}{\max_{d,c,i} \mathbf{S}_{m,c,i}^d(n) - \min_{d,c,i} \mathbf{S}_{m,c,i}^d(n)} \quad (\text{D.3})$$

Les taux de reconnaissance obtenus, à l'aide des différents paramètres, pour les bases de données Alpha, Beta et Gamma sont reportés respectivement dans les tableaux D.1, D.2 et D.3.

Vecteur de caractéristiques	Sans normalisation					Centrés/Réduits					Normalisation linéaire chaque descripteur					Normalisation linéaire chaque type de descripteur				
	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5		
Transformée en ondelettes frame par frame	Energie	72%	65%	73%	73%	73%	78%	77%	78%	72%	65%	73%	72%	65%	73%	72%	65%	73%		
	Entropie	88%	88%	88%	92%	90%	93%	88%	88%	88%	88%	88%	88%	88%	88%	88%	88%	88%		
	Moyenne	65%	48%	58%	18%	18%	18%	62%	68%	65%	48%	58%	65%	48%	58%	65%	48%	58%		
	Ecart type	83%	77%	85%	77%	77%	77%	80%	82%	83%	77%	85%	80%	82%	85%	83%	77%	85%		
	Tous	72%	65%	73%	18%	18%	18%	88%	88%	90%	87%	87%	90%	87%	87%	90%	87%	87%		
Transformée en ondelettes temporelle	Energie	55%	68%	68%	48%	48%	55%	60%	67%	55%	68%	67%	55%	68%	68%	55%	68%	68%		
	Entropie	57%	62%	68%	53%	60%	65%	57%	62%	62%	67%	67%	57%	62%	68%	62%	62%	68%		
	Moyenne	42%	37%	42%	40%	40%	40%	47%	42%	42%	37%	37%	42%	37%	42%	42%	37%	42%		
	Ecart type	57%	73%	77%	45%	47%	48%	52%	75%	75%	75%	75%	52%	75%	77%	57%	73%	77%		
	Tous	55%	68%	68%	47%	47%	47%	53%	68%	62%	67%	73%	62%	67%	75%	62%	67%	75%		
Transformée en ondelettes 2D+T	Energie	68%	67%	75%	65%	65%	73%	75%	85%	68%	67%	85%	68%	67%	85%	68%	67%	85%		
	Entropie	83%	82%	83%	83%	85%	83%	83%	87%	83%	87%	87%	83%	82%	83%	83%	82%	83%		
	Moyenne	68%	43%	57%	27%	27%	27%	68%	73%	72%	72%	68%	73%	72%	68%	73%	72%	72%		
	Ecart type	82%	75%	80%	75%	75%	75%	82%	83%	85%	85%	82%	83%	85%	80%	82%	75%	80%		
	Tous	68%	67%	75%	27%	27%	27%	83%	83%	85%	85%	83%	83%	85%	80%	85%	85%	87%		
Approche de Smith et al.	Energie	75%	X	X	65%	X	80%	X	X	75%	X	80%	X	X	75%	X	X	X		
	Entropie	80%	X	X	83%	X	82%	X	X	80%	X	82%	X	X	80%	X	X	X		
	Moyenne	42%	X	X	43%	X	43%	X	X	42%	X	42%	X	X	42%	X	X	X		
	Ecart type	78%	X	X	75%	X	83%	X	X	78%	X	78%	X	X	78%	X	X	X		
	Tous	75%	X	X	43%	X	43%	X	X	82%	X	82%	X	X	82%	X	X	X		
Nombre d'échelles / de subdivision	Energie	77%	75%	75%	67%	75%	67%	75%	73%	75%	75%	75%	73%	73%	75%	77%	75%	75%		
	Entropie	82%	82%	83%	89%	85%	89%	85%	85%	85%	85%	85%	83%	83%	85%	82%	82%	83%		
	Moyenne	45%	48%	48%	40%	40%	42%	42%	45%	42%	42%	47%	45%	48%	48%	45%	48%	42%		
	Ecart type	80%	80%	82%	77%	85%	85%	77%	85%	82%	83%	85%	80%	80%	82%	80%	80%	82%		
	Tous	77%	75%	75%	32%	40%	40%	32%	40%	42%	42%	42%	85%	83%	85%	85%	80%	83%		
Curv : Nombre d'échelles / subdivision Dct : taille fenêtre locale	Energie	68%	68%	68%	50%	50%	83%	83%	83%	85%	85%	85%	85%	85%	85%	85%	85%	85%		
	Entropie	82%	82%	82%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%		
	Moyenne	50%	50%	50%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%		
	Ecart type	82%	82%	82%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%		
	Tous	68%	68%	68%	35%	35%	35%	85%	85%	85%	85%	85%	85%	85%	85%	85%	85%	85%		
Analyse en Composantes Morphologiques	Energie	68%	68%	68%	50%	50%	83%	83%	83%	85%	85%	85%	85%	85%	85%	85%	85%	85%		
	Entropie	82%	82%	82%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%	83%		
	Moyenne	50%	50%	50%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%	37%		
	Ecart type	82%	82%	82%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%	67%		
	Tous	68%	68%	68%	35%	35%	35%	85%	85%	85%	85%	85%	85%	85%	85%	85%	85%	85%		

TAB. D.1 – Taux de reconnaissance (en %) de l'indexation de textures dynamiques selon différents descripteurs et normalisation sur la base de données Alpha.

	Vecteur de caractéristiques	Sans normalisation					Centrés/Réduits					Normalisation linéaire chaque descripteur					Normalisation linéaire chaque type de descripteur						
		3	4	5	3	4	5	3	4	5	3	4	5	3	4	5							
Transformée en ondelettes frame par frame	Nombre d'échelles	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5							
	Energie	50%	41%	35%	31%	31%	31%	52%	49%	51%	50%	49%	51%	50%	41%	35%							
	Entropie	54%	56%	57%	55%	57%	56%	42%	38%	41%	42%	38%	41%	41%	56%	57%							
	Moyenne	23%	19%	20%	27%	27%	27%	42%	38%	41%	42%	38%	41%	23%	19%	20%							
Transformée en ondelettes temporelle	Ecart type	51%	49%	44%	44%	44%	44%	51%	52%	50%	51%	52%	50%	51%	49%	44%							
	Tous	50%	41%	35%	28%	28%	28%	62%	61%	60%	62%	61%	60%	60%	56%	62%							
	Energie	28%	35%	38%	23%	28%	31%	30%	34%	37%	30%	34%	37%	28%	35%	38%							
	Entropie	23%	25%	27%	22%	25%	27%	22%	25%	28%	22%	25%	28%	23%	25%	27%							
Transformée en ondelettes 2D+T	Moyenne	17%	9%	11%	12%	12%	12%	15%	14%	15%	14%	15%	17%	9%	11%	11%							
	Ecart type	31%	35%	43%	28%	32%	36%	30%	38%	43%	39%	43%	46%	31%	35%	43%							
	Tous	28%	35%	38%	22%	26%	27%	33%	43%	46%	33%	43%	46%	28%	35%	38%							
	Energie	48%	38%	40%	30%	30%	30%	57%	59%	61%	62%	59%	61%	48%	38%	40%							
Approche de Smith et al.	Moyenne	62%	62%	59%	57%	60%	61%	62%	62%	65%	62%	65%	62%	62%	62%	59%							
	Ecart type	23%	19%	17%	19%	19%	19%	40%	35%	33%	40%	35%	33%	23%	19%	17%							
	Tous	48%	38%	40%	19%	19%	19%	62%	62%	65%	62%	65%	62%	48%	38%	40%							
	Energie	40%	X	X	28%	X	X	59%	X	X	65%	X	65%	40%	X	X							
Nombre d'échelles / de subdivisions	Entropie	63%	X	X	61%	X	X	65%	X	X	65%	X	65%	63%	X	X							
	Moyenne	10%	X	X	15%	X	X	35%	X	X	35%	X	10%	X	X								
	Ecart type	51%	X	X	44%	X	X	65%	X	X	65%	X	51%	X	X								
	Tous	40%	X	X	15%	X	X	67%	X	X	67%	X	66%	X	X								
Curv : Nombre d'échelles / subdivision	3 / 2	3 / 2	4 / 2	4 / 4	5 / 2	5 / 4	3 / 2	3 / 4	4 / 2	4 / 4	5 / 2	5 / 4	3 / 2	3 / 4	4 / 2	4 / 4	5 / 2	5 / 4					
	46%	48%	48%	47%	49%	46%	52%	53%	53%	53%	54%	44%	54%	57%	57%	61%	61%	48%	48%	51%	47%	49%	
	52%	57%	57%	62%	64%	52%	54%	56%	59%	61%	52%	58%	56%	60%	60%	62%	62%	52%	57%	57%	62%	61%	64%
	19%	17%	22%	27%	22%	6%	15%	15%	15%	14%	15%	23%	25%	24%	24%	28%	19%	17%	22%	27%	25%	22%	
Dct : taille fenêtre locale	Moyenne	53%	56%	52%	49%	59%	58%	56%	62%	60%	57%	60%	67%	53%	56%	52%	56%	52%	56%	52%	56%	51%	53%
	Ecart type	46%	48%	45%	47%	49%	6%	15%	15%	15%	14%	15%	15%	14%	15%	15%	15%	14%	15%	14%	15%	15%	14%
	Tous	46%	48%	45%	47%	49%	6%	15%	15%	15%	14%	15%	15%	14%	15%	15%	15%	14%	15%	14%	15%	15%	14%
	5 / 4	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32
Analyse en Composantes Morphologiques	Energie	46%			38%			64%			46%			46%									
	Entropie	67%			63%			66%			67%			67%									
	Moyenne	24%			14%			23%			24%			24%									
	Ecart type	64%			44%			68%			64%			64%									
Tous	46%			14%			70%			46%			54%										

TAB. D.2 – Taux de reconnaissance (en %) de l'indexation de textures dynamiques selon différents descripteurs et normalisation sur la base de données Beta.

Vecteur de caractéristiques	Sans normalisation					Centrés/Réduits					Normalisation linéaire chaque descripteur					Normalisation linéaire chaque type de descripteur				
	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5		
Transformée en ondelettes frame par frame	Energie	45%	50%	41%	35%	35%	49%	56%	60%	49%	56%	60%	49%	56%	60%	49%	56%	60%		
	Entropie	49%	52%	55%	48%	52%	49%	52%	55%	49%	52%	55%	49%	52%	55%	49%	52%	55%		
	Moyenne	21%	11%	15%	25%	25%	21%	11%	15%	25%	25%	21%	11%	15%	25%	25%	21%	11%		
	Ecart type	49%	55%	53%	43%	43%	49%	55%	53%	43%	43%	49%	55%	53%	43%	43%	49%	55%		
	Tous	45%	50%	41%	26%	26%	45%	50%	41%	26%	26%	45%	50%	41%	26%	26%	45%	50%		
Transformée en ondelettes temporelle	Energie	17%	22%	34%	23%	23%	23%	28%	31%	23%	28%	31%	23%	28%	31%	23%	28%	31%		
	Entropie	23%	23%	25%	23%	24%	23%	24%	26%	23%	24%	26%	23%	24%	26%	23%	24%	26%		
	Moyenne	16%	12%	10%	14%	15%	16%	12%	10%	14%	15%	16%	12%	10%	14%	15%	16%	12%		
	Ecart type	20%	29%	34%	21%	24%	26%	21%	24%	28%	20%	28%	34%	21%	24%	26%	20%	28%		
	Tous	17%	22%	34%	21%	22%	23%	21%	22%	23%	22%	23%	24%	21%	22%	23%	22%	23%		
Transformée en ondelettes 2D+T	Energie	49%	45%	41%	33%	33%	53%	59%	64%	53%	59%	64%	53%	59%	64%	53%	59%	64%		
	Entropie	54%	56%	59%	56%	56%	56%	59%	61%	56%	59%	61%	56%	59%	61%	56%	59%	61%		
	Moyenne	21%	9%	13%	8%	8%	21%	9%	13%	8%	8%	21%	9%	13%	8%	8%	21%	9%		
	Ecart type	54%	52%	50%	41%	41%	49%	45%	41%	33%	33%	49%	45%	41%	33%	33%	49%	45%		
	Tous	49%	45%	41%	8%	8%	49%	45%	41%	8%	8%	49%	45%	41%	8%	8%	49%	45%		
Approche de Smith et al.	Energie	44%	X	X	31%	X	56%	X	X	56%	X	X	56%	X	X	56%	X	X		
	Entropie	60%	X	X	57%	X	59%	X	X	59%	X	X	59%	X	X	59%	X	X		
	Moyenne	13%	X	X	11%	X	11%	X	X	11%	X	X	11%	X	X	11%	X	X		
	Ecart type	52%	X	X	41%	X	41%	X	X	41%	X	X	41%	X	X	41%	X	X		
	Tous	44%	X	X	11%	X	11%	X	X	11%	X	X	11%	X	X	11%	X	X		
Nombre d'échelles / de subdivision	Energie	43%	45%	48%	37%	50%	39%	53%	52%	39%	53%	52%	39%	53%	52%	39%	53%	52%		
	Entropie	51%	52%	55%	52%	52%	52%	52%	53%	52%	52%	53%	52%	52%	53%	52%	52%	53%		
	Moyenne	18%	8%	11%	9%	12%	18%	9%	12%	14%	13%	15%	16%	15%	14%	16%	15%	14%		
	Ecart type	47%	57%	55%	45%	53%	45%	53%	54%	45%	53%	55%	44%	60%	59%	44%	60%	59%		
	Tous	43%	46%	48%	9%	12%	9%	12%	14%	13%	15%	15%	53%	60%	60%	53%	60%	60%		
Curv : Nombre d'échelles / subdivision		5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32	5 / 4	32			
Dict : taille fenêtre locale		44%		33%		33%		33%		33%		33%		33%		33%				
Analyse en Composantes Morphologiques	Energie																	62%		
	Entropie																	59%		
	Moyenne																	19%		
	Ecart type																	61%		
Tous																	63%			

TAB. D.3 – Taux de reconnaissance (en %) de l'indexation de textures dynamiques selon différents descripteurs et normalisation sur la base de données Gamma.

Publications

Articles de revues internationales

1. *Decomposition of Dynamic Textures using Morphological Component Analysis*, Sloven Dubois, Renaud Péteri, Michel Ménard, IEEE Transactions on Circuits and Systems for Video Technology. (Soumis en juillet 2010)

Chapitre de livre

1. *Morphological Component Analysis for decomposing Dynamic Textures*, Sloven Dubois, Renaud Péteri, Michel Ménard, Springer-Verlab. (Soumis en octobre 2010)

Conférences internationales avec comité de lecture

1. *Decomposition of Dynamic Textures using Morphological Component Analysis : a New Adaptative Strategy*, Sloven Dubois, Renaud Péteri, Michel Ménard, 20th International Conference on Pattern Recognition (ICPR 2010), 23-26 août 2010, Istanbul, Turquie.
2. *Description of New Image Processing Tools Improving Echo Contrast Detection for ECHO-PIV Method*, Mathieu Lugiez, Damien Coisne, Michel Ménard, Sloven Dubois, C. Cuvier, Valérie Deplano, EUROECHO 11 Congress, ESC Congress and Events Division, 8-11 décembre 2010, Copenhagen, Danemark.
3. *A 3D Discrete Curvelet Based Method for Segmenting Dynamic Textures*, Sloven Dubois, Renaud Péteri, Michel Ménard, 16th International Conference on Image Processing (ICIP 2009), 7-11 novembre 2009, Caire, Egypte.
4. *Segmenting and Indexing Old Documents using a Letter Extraction*, Mickael Coustaty, Sloven Dubois, Jean-Marc Ogier, Michel Ménard, 8th International Workshop on Graphics Recognition (GREC 2009), 22-23 juillet 2009, La Rochelle, France.
5. *A Comparison of Wavelet Based Spatio-temporal Decomposition Methods for Dynamic Texture Recognition*, Sloven Dubois, Renaud Péteri, Michel Ménard, 4th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2009), 10-12 juin 2009, Póvoa do Varzim, Portugal.
6. *Adding a Noise Component To A Color Decomposition Model For Improving Color Texture Extraction*, Sloven Dubois, Mathieu Lugiez, Renaud Péteri, Michel Ménard, 4th European Conference on Colour in Graphics, Imaging, and Vision (CGIV 2008), 9-13 juin 2008, Barcelone, Espagne.
7. *Spatio-temporal Extension Of Color Decomposition Model And Dynamic Color Structure-Texture Extraction*, Mathieu Lugiez, Sloven Dubois, Michel Ménard, Abdallah El-Hamidi, 4th European Conference on Colour in Graphics, Imaging, and Vision (CGIV 2008), 9-13 juin 2008, Barcelone, Espagne.

8. *Assessment of 2D Velocity Field using Contrast Echo. Comparison of Weiner Filtering and Cross-correlation Method*, Sloven Dubois, Damien Coisne, Michel Ménard, EURO ECHO 10 Congress, ESC Congress and Events Division, 6-9 décembre 2006, Prague, République Tchèque.

Conférences nationales avec comité de lecture

1. *Analyse de Textures Dynamiques par Décompositions Spatio-Temporelles : Application à l'Estimation du Mouvement Global*, Sloven Dubois, Renaud Péteri., Michel Ménard, COMpression et REprésentation des Signaux Audiovisuels (CORESA 2010), 26-27 octobre 2010, Lyon, France.
2. *Segmentation de Textures Dynamiques à l'aide d'une Méthode utilisant la Transformée en Curvelets 3D*, Sloven Dubois, Renaud Péteri, Michel Ménard, XXIIe Colloque GRETSI - Traitement du Signal et des Images, 8-11 septembre 2009, Dijon, France.
3. *ECHO-PIV : Une Nouvelle Méthode d'Estimation de Champ de Vitesse 2D par l'Échographie*, Damien Coisne, Nicolas Bernard, Luc Christiaens, Robert Perrault, Sloven Dubois et Michel Ménard, 1st International Meeting of the French Society of Hypertension, 10-14 décembre 2007, Paris, France.

Liste des algorithmes

1	Segmentation de textures dynamiques au sens des directions spatio-temporelles à l'aide de l'énergie de la transformée en curvelets 2D+T	98
2	Construction d'une structure en octree pour une vidéo	102
3	Premier critère d'homogénéité pour la segmentation de textures dynamiques à l'aide d'une structure en octree	105
4	Deuxième critère d'homogénéité dans le cadre de segmentation de textures dynamiques à l'aide d'une structure en octree	106
5	Analyse en composantes morphologiques d'un signal nD	122
6	Algorithme d'optimisation par essaim particulaire pour déterminer les paramètres du modèle de texture dynamique	167

Liste des tableaux

1.1	Ensemble des descripteurs utilisés par l'approche de [Polana 92].	34
1.2	Ensemble des descripteurs utilisés par l'approche Synergizing Spatial and Temporal Features (SSTF).	36
1.3	Ensemble des descripteurs utilisés par l'approche de Peteri et Chetverikov [Péteri 05, Péteri 04].	37
1.4	Ensemble des descripteurs utilisés par l'approche de Otsuka et Horikoshi [Otsuka 98].	45
3.1	Tableaux indiquant la taille du plus petit sous-cube de coefficients en fonction du nombre d'échelles et d'orientations choisit dans la transformée en curvelet $2D+T$ selon la taille de la séquence.	102
4.1	Temps de calcul nécessaire pour effectuer une analyse T_i ou une synthèse T_i^T avec le dictionnaire (Φ_1 : transformée en curvelets, Φ_2 : transformée en cosinus locale) choisi, selon deux configurations matérielles différentes.	127
4.2	Temps de calcul estimés nécessaires pour effectuer la décomposition à l'aide de l'algorithme des MCA selon les différentes stratégies de seuillage. Il s'agit du temps moyen estimé sur la base DynTex pour une vidéo. De plus, nous indiquons le nombre de jours pour la décomposition complète de la base, ceci pour les deux plateformes utilisées.	131
4.3	Moyennes et écarts-types de l'erreur sur les coefficients mal sélectionnés d'une stratégie $S1$ (par exemple SSACL) par rapport à une stratégie de référence $S2$ (par exemple SSL) pour chacune des bases de notre dictionnaire (ζ_1 pour la transformée en curvelets $2D+T$ et ζ_2 pour la transformée en cosinus locale $2D+T$).	133
4.4	Moyennes et écart-types de l'erreur en norme ℓ_2 entre deux stratégies $S1$ (par exemple SSACL) et $S2$ (par exemple SSL) pour chacune des bases de notre dictionnaire (ξ_1 pour la transformée en curvelets $2D+T$, ξ_2 pour la transformée en cosinus locale $2D+T$) et pour sa reconstruction ξ_T	136
5.1	Exemple d'une matrice de confusion pour trois classes.	148
5.2	Taux de reconnaissance (en %) de l'indexation de textures dynamiques selon les différents descripteurs et sur trois bases de données. † représente les meilleurs taux de reconnaissance pour un descripteur sur une base de données. ★ représente le meilleur taux de reconnaissance pour une base de données.	150
5.3	Dimension des vecteurs caractéristiques en fonction des méthodes d'analyses. . .	151
D.1	Taux de reconnaissance (en %) de l'indexation de textures dynamiques selon différents descripteurs et normalisation sur la base de données Alpha.	174
D.2	Taux de reconnaissance (en %) de l'indexation de textures dynamiques selon différents descripteurs et normalisation sur la base de données Beta.	175
D.3	Taux de reconnaissance (en %) de l'indexation de textures dynamiques selon différents descripteurs et normalisation sur la base de données Gamma.	176

Table des figures

1	♠ Exemples de textures dynamiques. Les séquences d'images sont visualisées ici comme des cubes de données tri-dimensionnelles.	12
2	Quelques exemples applicatifs des textures dynamiques : (a) synthèse de la surface d'un lac selon plusieurs paramètres (d'après [Doretto 02]), (b) résultats de segmentation spatio-temporelle (d'après [Li 09b]) et (c) détection de changement dans une application de vidéo surveillance (d'après [Phillips 02]).	13
3	Ensemble des contributions de ce manuscrit.	15
1.1	Exemples de textures dynamiques. On peut observer les séquences d'images d'un escalator (a), de la surface d'un lac (b) et (c), d'une hélice sans fin en rotation due au vent (d) et d'une mer calme (e). Ici chaque séquence d'images est vue comme un cube de données 3D dans lequel on réalise des coupes afin d'observer les différents comportement spatio-temporels.	22
1.2	Principales publications sur le thème des textures dynamiques au cours des dernières années.	23
1.3	Répartition des principales publications sur le thème des textures dynamiques selon les principaux domaines applicatifs.	24
1.4	Répartition des principales publications selon leur définition de la texture dynamique.	26
1.5	Taxonomie des textures dynamiques.	27
1.6	Ensemble des textures dynamiques de la base de données MIT [Szummer 96b].	29
1.7	Ensemble des textures dynamiques de la base de données UCLA [Saisan 01].	30
1.8	Ensemble des textures dynamiques de la base de données DynTex [Péteri 10b].	32
1.9	Répartition des principales publications vis-à-vis de la base de données utilisée dans les expérimentations.	33
1.10	Répartition des principales publications selon les différentes approches de caractérisation.	33
1.11	Illustration de la méthode de [Polana 92].	35
1.12	Illustration de la méthode de [Peh 99].	36
1.13	Illustration de la méthode proposée par Péteri et Chetverikov [Péteri 05, Péteri 04].	37
1.14	Illustration de la méthode de [Rahman 04a].	38
1.15	Illustration de l'approche de Lu et Xie [Lu 05].	39
1.16	Illustration de la méthode utilisant un modèle auto-régressif à moyenne mobile (ARMA).	41
1.17	Illustration de la méthode de [Chan 05a].	42
1.18	Illustration de la méthode utilisant la réponse impulsionnelle des variables d'état [Fujita 03].	44
1.19	Plans tangents à une trajectoire définissant la direction spatio-temporelle du mouvement par l'approche de Otsuka et Horikoshi [Otsuka 98].	44
1.20	Illustration de la méthode de [Schodl 00].	45
1.21	Illustration de l'opérateur Volume Local Binary Patterns (VLBP) de [Zhao 07b].	46

1.22	<i>Illustration de la méthode de [Gao 08].</i>	47
2.1	<i>Pavages temps-fréquence sur la base des distributions de Dirac (a) et sur la base de Fourier (b).</i>	55
2.2	<i>Pavages temps-fréquence de deux transformées de Fourier à fenêtre glissante (fenêtre étroite (a) et fenêtre plus large (b)) illustrant le principe d'incertitude d'Heisenberg.</i>	55
2.3	<i>Pavage temps-fréquence obtenu lors d'une transformée en ondelettes. La résolution temps-fréquence dépend de l'échelle d'analyse.</i>	56
2.4	<i>Schéma du banc de filtres de la transformée en ondelettes permettant de passer du niveau de résolution j à $j + 1$.</i>	58
2.5	<i>Schéma du banc de filtres de la transformée en ondelettes rapide pour $j_{max} = 3$.</i>	59
2.6	<i>Banc de filtres de la décomposition en ondelettes d'une image. Avec a^{j+1} l'approximation de niveau de résolution $j + 1$, d_v^{j+1} les détails verticaux, d_h^{j+1} les détails horizontaux et d_d^{j+1} les détails diagonaux.</i>	59
2.7	<i>Transformée en ondelettes 2D appliquée sur une séquence vidéo image par image.</i>	60
2.8	<i>Transformée en ondelettes 1D appliquée sur les profils temporels d'une séquence d'images.</i>	61
2.9	<i>Décomposition en ondelettes 2D+T appliquée sur une séquence d'images.</i>	62
2.10	<i>Transformée en ondelettes de J. R. Smith et al. [Smith 02] appliquée sur une séquence d'images.</i>	63
2.11	<i>Découpage du domaine fréquentiel par la transformée en curvelets 2D continue. Visualisation des supports de $U_j(r, \theta)$, de $W_j(2^{-j}r)$ et de $V_j\left(\frac{2^{\lfloor j/2 \rfloor} \theta}{2\pi}\right)$</i>	67
2.12	<i>Pavage du domaine fréquentiel par la transformée en curvelets 2D discrète (c) obtenue à l'aide des fenêtres $\tilde{U}_{j,\ell}$. (a), respectivement (b), représente le pavage radiale \tilde{W}_j respectivement angulaire $\tilde{V}_{j,\ell}$.</i>	69
2.13	<i>Les différentes étapes de l'implémentation USFFT de la transformée en curvelets.</i>	70
2.14	<i>Les différentes étapes de l'implémentation par recalage de la transformée en curvelets par wrapping.</i>	71
2.15	<i>Opération de wrapping</i>	71
2.16	<i>Atomes 2D de curvelets $\varphi_{j,\ell,\mathbf{k}}$ à différentes échelles j et orientations ℓ.</i>	72
2.17	<i>Résultat de la transformée en curvelets sur une image en utilisant 2 échelles de décomposition.</i>	73
2.18	<i>Comparaison de la parcimonie de la transformée en curvelets et de la transformée en ondelettes à l'aide du PSNR de la reconstruction partielle.</i>	74
2.19	<i>Pavage du domaine fréquentiel 3D de la transformée en curvelets 3D discrète (c) obtenu à l'aide des fenêtres $\tilde{U}_{j,\ell}$. La partie (a), respectivement la partie (b), représente le découpage radiale \tilde{W}_j, respectivement angulaire $\tilde{V}_{j,\ell}$.</i>	75
2.20	<i>Atomes 3D de curvelets $\varphi_{j,\ell,\mathbf{k}}$ à différentes échelles j et orientations ℓ.</i>	76
2.21	<i>Comparaison de la parcimonie de la transformée en curvelets et de la transformée en ondelettes à l'aide du PSNR de la reconstruction partielle.</i>	77
2.22	<i>Illustration du problème d'homogénéité des variables spatiales et temporelles.</i>	78
2.23	<i>Principe de la représentation d'une caractéristique dans un espace 2D construit sur l'échelle et l'orientation</i>	79

2.24	<i>Projection de Mercator</i>	80
2.25	<i>Cartographie d'une projection de Mercator</i>	80
2.26	<i>Projection de Mercator des énergies des sous-bandes de détails de la transformée en curvelets de séquences d'images synthétiques.</i>	81
2.27	<i>Cartes de Mercator des énergies des sous-bandes de détails de la transformée en curvelets d'une séquence d'images fixes, ne présentant donc aucun mouvement.</i>	82
2.28	♠ <i>Cartes de Mercator des énergies des sous-bandes de détails de la transformée en curvelets sur la séquence d'images d'un escalator.</i>	83
2.29	♠ <i>Cartes de Mercator des énergies des sous-bandes de détails de la transformée en curvelets sur la séquence d'images de la surface d'un lac.</i>	83
2.30	♠ <i>Cartes de Mercator des énergies des sous-bandes de détails de la transformée en curvelets sur la séquence d'images d'un champ d'herbes.</i>	84
3.1	♠ <i>Exemples de textures dynamiques avec leur support respectif. Les colorations bleue et rouge correspondent à des supports spatio-temporels de textures dynamiques différents ; la coloration violette correspond à l'intersection de deux supports.</i>	90
3.2	♠ <i>Coupes spatio-temporelles de textures dynamiques. On peut observer l'onde porteuse (1), les phénomènes locaux (2) et l'interférence entre les deux (3) sur les séquences "Surface de la mer" et "Arbre dans le vent".</i>	91
3.3	♠ <i>Textures dynamiques synthétiques (ligne (1)) obtenues à l'aide du modèle décrit précédemment (Définition 8). (a) Vidéo représentant l'onde porteuse, (b) phénomènes localement oscillants, (c) texture dynamique composée de (a) et (b). La seconde ligne correspond à des textures dynamiques issues de DynTex (645b410, 649i810 et 64ac510).</i>	94
3.4	<i>Illustration de trajectoires spatio-temporelles d'objets évoluant selon la direction de l'axe \vec{x} à différentes vitesses.</i>	95
3.5	♠ <i>Illustration du lien entre les directions spatio-temporelles et les directions de propagation des ondes porteuse. La vitesse de propagation de \mathcal{P}_1, respectivement \mathcal{P}_2 et \mathcal{P}_3 est nulle, respectivement faible et un peu plus grande.</i>	96
3.6	♠ <i>Projection de Mercator de l'énergie de la transformée en curvelets d'une séquence d'images de synthèse.</i>	97
3.7	<i>Illustration de la méthode permettant de rechercher et de connecter les maxima locaux des énergies de la transformée en curvelets $2D+T$.</i>	99
3.8	<i>Illustration du support d'un atome de curvelet dans le cadre $2D$.</i>	100
3.9	♠ <i>Les deux régions, (b) et (c), résultant de la segmentation de la vidéo de synthèse (a) à l'aide des atomes de la transformée en curvelets $2D+T$.</i>	100
3.10	<i>Principe général du découpage à l'aide de la structure en octree.</i>	103
3.11	<i>Fonction de normalisation des énergies de la transformée en curvelets $2D+T$ définie pour pénaliser les orientations purement spatiales. A gauche un profil $2D$, et à droite une vue $3D$.</i>	107

3.12	♠	<i>Résultats de la segmentation en régions spatio-temporelles, en utilisant une structure en octree avec l'énergie de la transformée en curvelets 2D+T comme critère d'homogénéité. A gauche, les images de synthèse originales, et à droite le résultat de la segmentation pour chacune d'elles. Chaque couleur (bleue et rouge respectivement labelisées par 1 et 2) représente une région différente. Une région non colorée correspond à une zone d'ambiguïté. Les traits noirs représentent les frontières des sous-cubes de la structure en octree.</i>	108
3.13	♠	<i>(a) Séquence originale 645c310. Les principales directions spatio-temporelles sont symbolisées par des flèches. (b) Résultat de la segmentation de la vidéo. La couleur rouge, labelisée par le nombre 1, représente la région détectée. Une région dépourvue de couleur correspond à une région considérée comme ambiguë. Les traits noirs représentent les frontières des sous-cubes de la structure en octree.</i>	110
3.14	♠	<i>(a) Séquence originale 645b310. Les principales directions spatio-temporelles sont symbolisées ici par des flèches. (b) Résultat de la segmentation de la vidéo. Chaque couleur (rouge, verte et bleu, respectivement labelisée par les nombres 1, 2 et 3) représente une région différente. Une région dépourvue de couleur correspond à une région considérée comme ambiguë. Les traits noirs représentent les frontières des sous-cubes de la structure en octree.</i>	111
3.15	♠	<i>(a) Séquence originale 6482210. Les principales directions spatio-temporelles sont symbolisées ici par des flèches. (b) Résultat de la segmentation de la vidéo. Chaque couleur (rouge, verte et bleu, respectivement labelisée par les nombres 1, 2 et 3) représente une région différente. Une région dépourvue de couleur correspond à une région considérée comme ambiguë. Les traits noirs représentent les frontières des sous-cubes de la structure en octree.</i>	112
4.1		<i>Exemples de décomposition en composantes morphologiques issus de [Starck 04, Fadili 09b]. (a) L'observation, (b) ses structures isotropiques (extraites à l'aide de la transformée par ondelettes à trous) et (c) ses lignes (projetées sur une base des ridgelets). (d) L'image originale, (e) ses structures isotropiques et son fond (extraites par la transformée en ondelettes) et (f) ses courbes (projetées sur une base de curvelets).</i>	119
4.2		<i>Représentation schématique de plages déterminées selon les stratégies SSL et SSE sur une distribution d'illustration.</i>	124
4.3		<i>Évolution de $\lambda^{(k)}$ pour les stratégies SSL et SSE en fonction du nombre d'itérations k pour différentes valeurs de N_{iter}</i>	124
4.4		<i>Illustration, dans le cadre de deux bases, du principe de la stratégie de seuillage Mean of Max.</i>	125
4.5		<i>Deux stratégies de seuillage conduisant à des évolutions différentes de la valeur de seuil durant une itération de la boucle principale de l'agorithme des MCA. . .</i>	129
4.6		<i>Étude des stratégies SSL, SSE, SSACl et SSACe. Les courbes représentent le nombre de coefficients seuillés (en échelle logarithmique) au cours des itérations sur quatre séquences d'images différentes.</i>	132
4.7		<i>Étude des stratégies SSL, SSE, SSACl et SSACe : les courbes représentent l'erreur de reconstruction en norme ℓ_2 en fonction du nombre d'itérations.</i>	134

4.8	♠	Résultat de la décomposition d'une séquence d'images à l'aide de l'algorithme MCA et selon la stratégie SSACe. Une coupe spatio-temporelle xt est réalisée sur chacune des vidéos afin d'observer le comportement temporel.	137
4.9	♠	Résultat de la décomposition d'une séquence d'images à l'aide de l'algorithme MCA et selon la stratégie SSACe. Des profils temporels ont été extraits afin de visualiser les résultats sur l'axe temporel.	138
4.10	♠	Résultat de la décomposition d'une séquence à l'aide de l'algorithme MCA et selon la stratégie SSACe. Des zones d'intérêt sont mises en évidence sous la forme de surfaces afin de mieux visualiser les effets de l'algorithme.	139
4.11	♠	Résultat de la décomposition d'une séquence à l'aide de l'algorithme MCA et selon la stratégie SSACe. Des coupes spatio-temporelles dans différentes directions ont été effectuées afin de mieux visualiser les phénomènes.	140
4.12	♠	Résultat de la décomposition d'une séquence à l'aide de l'algorithme MCA et selon la stratégie SSACe. Des zones d'intérêt sont mises en évidence sous la forme de surfaces afin de mieux comprendre les effets de l'algorithme.	141
5.1		Découpage des coefficients de la transformée en cosinus locale pour obtenir comme dans une analyse multi-résolutions différentes sous-bandes de détails.	147
5.2		Illustration de la méthode leave-one-out pour un élément.	148
5.3		Approche adaptée pour la recherche de séquences d'images par le contenu.	153
5.4		Résultats d'une recherche par le contenu à l'aide d'une requête sous forme de séquence d'images.	154
5.5		Méthode de comparaison entre l'estimation du mouvement global sur la composante géométrique et l'estimation du mouvement sur la séquence originale.	156
5.6	♠	Flots optiques estimés sur la vidéo originale et sur la composante géométrique de l'analyse en composantes morphologiques.	156
5.7		Critère d'homogénéité pour une zone d'intérêt du champ de vecteurs mouvement représenté dans la figure 5.6.	157
5.8		Diagramme d'orientations pour l'ensemble de la séquence d'images des vecteurs mouvement présentés sur la figure 5.6.	158

Bibliographie

- [Amiaz 07] T. Amiaz, S. Fazeakas, D. Chetverikov & N. Kiryati. *Detecting Regions of Dynamic Texture*. In International Conference on Scale Space and Variational Methods in Computer Vision (SSVM 07), pages 848–859, 2007. 40
- [Aujol 05a] J.F. Aujol, G. Aubert, L. Blanc-Féraud & A. Chambolle. *Image Decomposition into a Bounded Variation Component and an Oscillating Component*. Mathematical Imaging and Vision, vol. 22, pages 71–88, 2005. 118
- [Aujol 05b] J.F. Aujol & A. Chambolle. *Dual Norms and Image Decomposition Models*. Computer Vision, vol. 63, pages 85–104, 2005. 117, 118
- [Aujol 06] J.F. Aujol & G. Gilboa. *Constrained and SNR-Based Solutions for TV-Hilbert Space Image Denoising*. Mathematical Imaging and Vision, vol. 26, pages 217–237, 2006. 118
- [Backer 10] A. Backer, B. Rolfs & F. Yu. *Video Restoration using Multichannel-Morphological Component Analysis Inpainting*, 2010. 126
- [Bamberger 92] R.H. Bamberger & M.J.T. Smith. *A Filter Bank for the Directional Decomposition of Images : Theory and Design*. IEEE Transactions on Signal Processing, vol. 40, pages 882–893, 1992. 64
- [Bar-Joseph 01] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski & M. Werman. *Texture Mixing and Texture Movie Synthesis using Statistical Learning*. IEEE Transactions on Visualization and Computer Graphics, vol. 7, pages 120–135, 2001. 24
- [Benedetto 00] J.L. Benedetto. *Mécanique quantique*. Ecole Polytechnique, 2000. 54
- [Bobin 07] J. Bobin, J.L. Starck, J.M. Fadili, Y. Moudden & D.L. Donoho. *Morphological Component Analysis : An Adaptive Thresholding Strategy*. IEEE Transactions on Image Processing, pages 2675–2681, 2007. 125, 128, 129
- [Bobin 08] J. Bobin, J.L. Starck, Y. Moudden & J.M. Fadili. *Blind Source Separation : the Sparsity Revolution*. Advances in Imaging and Electron Physics, pages 221–298, 2008. 119
- [Bouthemy 98] Patrick Bouthemy & Ronan Fablet. *Motion Characterization from Temporal Cooccurrences of Local Motion-Based Measures for Video Indexing*. In International Conference on Pattern Recognition (ICPR 98), pages 905–908, 1998. 35, 37
- [Candès 98] E. Candès. *Ridgelets : Theory and Applications*. PhD thesis, University of Stanford, 1998. 64
- [Candès 00a] E. Candès & D.L. Donoho. *Curvelets - A Surprisingly Effective Nonadaptive Representation For Objects with Edges*, 2000. 74
- [Candès 00b] E. Candès & D.L. Donoho. *Curvelets and Curvilinear Integrals*, 2000. 64, 65

- [Candès 03] E. Candès & L. Demanet. *Curvelets and Fourier Integral Operators*. Comptes Rendus Mathématique, vol. 336, pages 395 – 398, 2003. 67
- [Candès 04] E. Candès & D.L Donoho. *New Tight Frames of Curvelets and Optimal Representations of Objects with Piecewise C^2 Singularities*. Communications on Pure and Applied Mathematics, vol. 57, pages 219–266, 2004. 14, 64, 65, 67, 74
- [Candès 05a] E. Candès & L. Demanet. *The curvelet Representation of Wave Propagators is Optimally Sparse*. Communications on Pure and Applied Mathematics, vol. 58, pages 1472–1528, 2005. 14, 67
- [Candès 05b] E. Candès, L. Demanet, D.L. Donoho & L. Ying. *Fast Discrete Curvelet Transforms*. Rapport technique, California Institute of Technology, 2005. 64, 65, 69, 74, 75
- [Candès 05c] E. Candès & D.L. Donoho. *Continuous Curvelet Transform : I. Resolution of the Wavefront Set*. Applied and Computational Harmonic Analysis, vol. 19, pages 162 – 197, 2005. 67, 74
- [Candès 05d] E. Candès & D.L. Donoho. *Continuous Curvelet Transform : II. Discretization and Frames*. Applied and Computational Harmonic Analysis, vol. 19, pages 198 – 222, 2005. 67, 74
- [Chan 01] T.F. Chan, S. Osher & J. Shen. *The Digital TV Filter and Nonlinear Denoising*. IEEE Transactions on Image Processing, vol. 10, pages 231–241, 2001. 117
- [Chan 05a] A.B. Chan & N. Vasconcelos. *Mixtures of Dynamic Textures*. In IEEE International Conference on Computer Vision (ICCV 05), pages 641–647, 2005. 42, 183
- [Chan 05b] A.B. Chan & N. Vasconcelos. *Probabilistic Kernels for the Classification of Auto-Regressive Visual Processes*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05), pages 846–851, 2005. 42
- [Chan 06] A.B. Chan & N. Vasconcelos. *Layered Dynamic Textures*. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006. 42
- [Chan 07] A.B. Chan & N. Vasconcelos. *Classifying Video with Kernel Dynamic Textures*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 07), pages 1–6, 2007. 42
- [Chan 08] A.B. Chan & N. Vasconcelos. *Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, pages 909–926, 2008. 42
- [Chan 09] A.B. Chan & N. Vasconcelos. *Variational Layered Dynamic Textures*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 09), pages 1062–1069, 2009. 42
- [Chandrasekaran 04] V. Chandrasekaran, M.B. Wakin, D. Baron & R.G. Baraniuk. *Compression of Higher Dimensional Functions Containing Smooth Discontinuities*. In Annual Conference on Information Science and Systems, 2004. 64

- [Chetverikov 00] D. Chetverikov. *Pattern Regularity as a Visual Key*. Image and Vision Computing, vol. 18, pages 975–986, 2000. 37
- [Chetverikov 06] D. Chetverikov & S. Fazekas. *On Motion Periodicity of Dynamic Textures*. In British Machine Vision Conference (BMVC 06), pages 167–176, 2006. 25, 37
- [Combettes 04] P.L. Combettes. *Solving Monotone inclusions via Compositions of Nonexpansive Averaged Operators*. Optimization, vol. 53, pages 475–504, 2004. 121
- [Combettes 05] P.L. Combettes & V.R. Wajs. *Signal Recovery by Proximal Forward-Backward Splitting*. Multiscale Modeling and Simulation, vol. 4, pages 1168–1200, 2005. 121
- [Combettes 07] P.L. Combettes & J.P. Pesquet. *A Douglas-Rachford Splitting Approach to Nonsmooth Convex Variational Signal Recovery*. IEEE Journal of Selected Topics in Signal Processing, vol. 1, pages 564–574, 2007. 121
- [Cooper 06] L. Cooper, J. Liu & K. Huang. *Spatial Segmentation of Temporal Texture Using Mixture Linear Models*. In International Conference of Computer Vision (ICCV 06), pages 142–150, 2006. 42
- [Davies 04] M. Davies. *Identifiability Issues in Noisy ICA*. IEEE Signal Processing Letters, vol. 11, pages 470–473, 2004. 120
- [Derpanis 10] K.G. Derpanis & R.P. Wildes. *Dynamic Texture Recognition based on Distributions of Spacetime Oriented Structure*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 10), 2010. 48
- [Do 05] M.N. Do & M. Vetterli. *The Contourlet Transform : an Efficient Directional Multiresolution Image Representation*. IEEE Transactions on Image Processing, vol. 14, pages 2091–2106, 2005. 64
- [Donoho 98] D.L. Donoho. *Orthonormal Ridgelets and Linear Singularities*. SIAM Journal on Mathematical Analysis, vol. 31, pages 1062–1099, 1998. 64
- [Donoho 99a] D.L. Donoho. *Wedgelets : Nearly Minimax Estimation of Edges*. The Annals of Statistics, vol. 27, pages 859–897, 1999. 64
- [Donoho 99b] D.L. Donoho & M.R. Duncan. *Digital Curvelet Transform : Strategy, Implementation and Experiments*. In Wavelet Applications VII, pages 12–29. SPIE, 1999. 64, 65
- [Donoho 01] D.L. Donoho, X. Huo, I. Jermyn, P. Jones, G. Lerman, O. Levi & F. Natterer. *Beamlets and Multiscale Image Analysis*. In Multiscale and Multi-resolution Methods, pages 149–196, 2001. 64
- [Doretto 02] G. Doretto. *Dynamic texture modeling*. Master’s thesis, University of California, Los Angeles, CA, June 2002. 13, 183
- [Doretto 03a] G. Doretto, A. Chiuso, Y.N. Wu & S. Soatto. *Dynamic Textures*. International Journal of Computer Vision, vol. 51, pages 91–109, 2003. 41
- [Doretto 03b] G. Doretto, D. Cremers, P. Favaro & S. Soatto. *Dynamic Texture Segmentation*. In IEEE International Conference on Computer Vision (ICCV 03), pages 1236–1242, 2003. 41, 49

- [Doretto 03c] G. Doretto & S. Soatto. *Editable Dynamic Textures*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 03), pages 137–142, 2003. 41
- [Doretto 03d] G. Doretto & S. Soatto. *Towards Plenoptic Dynamic Textures*. In International Workshop on Texture Analysis and Synthesis, pages 25–30, 2003. 41
- [Doretto 04] G. Doretto, E. Jones & S. Soatto. *Spatially Homogeneous Dynamic Textures*. In IEEE European Conference on Computer Vision (ECCV 04), pages 591–602, 2004. 41
- [Doretto 05a] G. Doretto. *Dynamic Textures : modeling, learning, synthesis, animation, segmentation, and recognition*. PhD thesis, University of California, 2005. 41
- [Doretto 05b] G. Doretto. *Modeling Dynamic Scenes with Active Appearance*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05), pages 66–73, 2005. 42
- [Doretto 06] G. Doretto & S. Soatto. *Dynamic Shape and Appearance Models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, pages 0162–8828, 2006. 42
- [El-Hamidi 10] A. El-Hamidi, M. Ménard, M. Lugiez & C. Ghannam. *Weighted and Extended Total Variation for Image Restoration and Decomposition*. Pattern Recognition, vol. 43, pages 1564–1576, 2010. 118
- [Fablet 99] R. Fablet & P. Bouthemy. *Motion-Based Feature Extraction and Ascendant Hierarchical Classification for Video Indexing and Retrieval*. In Visual Information and Information Systems, pages 658–658. Springer Berlin / Heidelberg, 1999. 35
- [Fablet 00] R. Fablet, P. Bouthemy & P. Pérez. *Statistical Motion-based Video Indexing and Retrieval*. In International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information, pages 602–619, 2000. 35
- [Fablet 01a] R. Fablet & P. Bouthemy. *Motion Recognition Using Spatio-temporal Random Walks in Sequence of 2D Motion-related Measurements*. In IEEE International Conference on Image Processing (ICIP 01), pages 652–655, 2001. 35
- [Fablet 01b] R. Fablet & P. Bouthemy. *Non Parametric Motion Recognition using Temporal Multiscale Gibbs Models*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 01), vol. 1, page 501, 2001. 35
- [Fablet 02] R. Fablet, P. Bouthemy & P. Pérez. *Nonparametric Motion Characterization using Causal Probabilistic Models for Video Indexing and Retrieval*. IEEE Transactions on Image Processing, vol. 11(4), pages 393–407, 2002. 35
- [Fablet 03] R. Fablet & P. Bouthemy. *Motion Recognition Using Nonparametric Image Motion Models Estimated from Temporal and Multiscale Cooccurrence Sta-*

- tistics*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 12, pages 1619–1624, 2003. 35
- [Fadili 09a] J.M. Fadili & J.L. Starck. *Curvelets and Ridgelets*. In Robert A. Meyers, editeur, Encyclopedia of Complexity and Systems Science, pages 1718–1738. Springer, 2009. 74
- [Fadili 09b] J.M. Fadili, J.L. Starck, J. Bobin & Y. Moudden. *Image Decomposition and Separation using Sparse Representations : An Overview*. In IEEE Special Issue : Applications of Sparse Representation, 2009. 119, 186
- [Fadili 10a] J.M. Fadili. *Une exploration des problèmes inverses par la représentations parcimonieuses et l’optimisation non lisse*, 2010. 123
- [Fadili 10b] J.M. Fadili, J.L. Starck, M. Elad & D.L. Donoho. *MCALab : Reproducible Research in Signal and Image Decomposition and Inpainting*. IEEE Computing in Science and Engineering, vol. 12, pages 44–63, 2010. 136
- [Fazekas 05] S. Fazekas & D. Chetverikov. *Normal Versus Complete Flow in Dynamic Texture Recognition : a Comparative Study*. In International Workshop on Texture Analysis and Synthesis, pages 37–42, 2005. 38, 39
- [Fazekas 07a] S. Fazekas & D. Chetverikov. *Analysis and Performance Evaluation of Optical Flow Features for Dynamic Texture Recognition*. Image Communication, vol. 22, pages 680–691, 2007. 40
- [Fazekas 07b] S. Fazekas & D. Chetverikov. *Dynamic Texture Recognition Using Optical Flow Features and Temporal Periodicity*. In International Workshop on Content-Based Multimedia Indexing (CBMI 07), pages 25 –32, 2007. 40
- [Fazekas 09] S. Fazekas, T. Amiaz, D. Chetverikov & N. Kiryati. *Dynamic Texture Detection Based on Motion Analysis*. Computer Vision, vol. 82, pages 48–63, 2009. 40
- [Finch 04] M. Finch. GPU Gems : Programming Techniques, Tips, and Tricks for Real-Time Graphics, Chap.1. Randima Fernando, 2004. 92, 93
- [Finkel 74] R.A. Finkel & J.L. Bentley. *Quad Trees a Data Structure for Retrieval on Composite Keys*. Acta Informatica, vol. 4, pages 1–9, 1974. 101
- [Francos 93] J.M. Francos, A.Z. Meiri & B. Porat. *A Unified Texture Model based on a 2-D Wold-like Decomposition*. IEEE Transactions on Signal Processing, vol. 41, pages 2665 –2678, 1993. 119
- [Fujita 03] K. Fujita & S.K. Nayar. *Recognition of Dynamic Textures using Impulse Responses of State Variables*. In International Workshop on Texture Analysis and Synthesis, 2003. 43, 44, 183
- [Gabor 46] D. Gabor. *Theory of Communication*. Institution of Electrical Engineers, vol. 93, pages 429–459, 1946. 55
- [Gao 08] P. Gao & C.L. Xu. *Extended Statistical Landscape Features for Dynamic Texture Recognition*. In International Conference on Computer Science and Software Engineering (CSSE 08), pages 548–551, 2008. 47, 184

- [Gervautz 90] M. Gervautz & W. Purgathofer. *A Simple Method for Color Quantization : Octree Quantization*. In *New Trends in Computer Graphics*, pages 287–293. Springer Verlag, Berlin, 1990. 101, 104
- [Ghanem 07a] B. Ghanem & N. Ahuja. *Phase Based Modelling of Dynamic Textures*. In *International Conference on Computer Vision (ICCV 07)*, pages 1–8, 2007. 43
- [Ghanem 07b] B. Ghanem & N. Ahuja. *Phase PCA for Dynamic Texture Video Compression*. In *International Conference on Image Processing (ICIP 07)*, pages 425–428, 2007. 43
- [Ghanem 08] B. Ghanem & N. Ahuja. *Extracting a Fluid Dynamic Texture and the Background from Video*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 09)*, pages 1–8, 2008. 43
- [Horn 81] B.K.P. Horn & B.G. Schunck. *Determining Optical Flow*. *Artificial Intelligence*, vol. 17, pages 185–203, 1981. 34, 155
- [Kennedy 95] J. Kennedy & R. Eberhart. *Particle Swarm Optimization*. In *IEEE International Conference on Neural Networks*, pages 1942–1948, 1995. 165
- [Kobayashi 09a] T. Kobayashi, T. Higuchi, T. Miyajima & N. Otsu. *Recognition of Dynamic Texture Patterns Using CHLAC Features*. In *ECSIS Symposium on Bio-inspired Learning and Intelligent Systems for Security (BLISS 09)*, pages 58–60, 2009. 48
- [Kobayashi 09b] T. Kobayashi & N. Otsu. *Three-Way Auto-Correlation Approach to Motion Recognition*. *Pattern Recognition Letters*, vol. 30, pages 212 – 221, 2009. 48
- [Koldovsky 06] Z. Koldovsky & P. Tichavsky. *Methods of Fair Comparison of Performance of Linear ICA Techniques in Presence of Additive Noise*. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 873–876, May 2006. 120
- [Kung 88] T. Kung & W. Richards. *Inferring "Water" from Images*. *Natural Computation*, pages 224–233, 1988. 40
- [Li 08] C. Li & P.M. Hall. *Learning A Stable Structure To Describe Dynamic Texture*. In *British Machine Vision Conference (BMVC 08)*, 2008. 48
- [Li 09a] J. Li, L. Chen & Y. Cai. *Dynamic Texture Segmentation Using 3-D Fourier Transform*. In *International Conference on Image and Graphics (ICIG 09)*, pages 293–298, 2009. 49
- [Li 09b] J. Li, L. Chen & Y. Cai. *Dynamic Texture Segmentation Using Fourier Transform*. *Modern Applied Science*, vol. 3, pages 29–36, 2009. 13, 49, 183
- [Lions 79] P.L. Lions & B. Mercier. *Splitting Algorithms for the Sum of Two Non-linear Operators*. *SIAM Journal on Numerical Analysis*, vol. 16, pages 964–979, 1979. 121
- [Lu 05] Z. Lu, W. Xie, J. Pei & J. Huang. *Dynamic Texture Recognition by Spatio-Temporal Multiresolution Histograms*. In *IEEE Workshop on Motion and Video Computing (WACV 05)*, pages 241–246, 2005. 39, 183

- [Ma 09] Y. Ma & P. Cisar. *Event Detection using Local Binary Pattern based Dynamic Textures*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 09), pages 38–44, 2009. 47
- [Mallat 89] S. Mallat. *A Theory for Multiresolution Signal Decomposition : The Wavelet Representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence journal, vol. 11, pages 674–693, 1989. 54, 57
- [Mallat 98] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1998. 54, 57, 58, 59
- [Meyer 90] Y. Meyer. *Ondelettes et Opérateurs*, tome 1. Hermann, 1990. 54
- [Meyer 01] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*. American Mathematical Society, Boston, MA, USA, 2001. 14, 118
- [Meyer 07] Y. Meyer. *Workshop sur "An interdisciplinary approach to Textures and Natural Images Processing"*. Institut Henri Poincaré, Janvier 2007. 26
- [Nelson 92] R.C. Nelson & R. Polana. *Qualitative Recognition of Motion using Temporal Texture*. Computer Vision and Image Understanding, vol. 56, pages 78–89, 1992. 22, 24, 34, 37
- [Ojala 96] T. Ojala, M. Pietikäinen & D. Harwood. *A Comparative Study of Texture Measures with Classification based on Featured Distributions*. Pattern Recognition, vol. 29, pages 51 – 59, 1996. 46
- [Ojala 02] T. Ojala, M. Pietikäinen & T. Mäenpää. *Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pages 971–987, 2002. 46
- [Otsuka 98] K. Otsuka, T. Horikoshi, S. Suzuki & M. Fujii. *Feature Extraction of Temporal Texture Based on Spatiotemporal Motion Trajectory*. In International Conference on Pattern Recognition (ICPR 98), page 1047, 1998. 44, 45, 181, 183
- [Peh 99] C.H. Peh & L.F. Cheong. *Exploring Video Content in Extended Spatio-Temporal Textures*. In European workshop on Content-Based Multimedia Indexing, 1999. 24, 36, 183
- [Peh 02] C.H. Peh & L.F. Cheong. *Synergizing Spatial and Temporal Texture*. IEEE Transactions on Image Processing, vol. 11, pages 1179–1191, 2002. 34, 36
- [Phillips 02] W. Phillips, M. Shah & N.V. Lobo. *Flame Recognition in Video*. Pattern Recognition Letters, vol. 23, pages 319–327, 2002. 13, 183
- [Polana 92] R. Polana & R.C. Nelson. *Recognition of Motion from Temporal Texture*. In Conference on Computer Vision and Pattern Recognition (CVPR 92), 1992. 22, 24, 34, 35, 181, 183
- [Polana 97] R. Polana & R.C. Nelson. *Temporal Texture and Activity Recognition*. Computational Imaging and Vision, vol. 9, pages 87–124, 1997. 34

- [Péteri 04] R. Péteri & D. Chetverikov. *Qualitative Characterization of Dynamic Textures for Video Retrieval*. In International Conference on Computer Vision and Graphics (ICCVG 04), pages 33–38, 2004. 37, 38, 181, 183
- [Péteri 05] R. Péteri & D. Chetverikov. *Dynamic Texture Recognition Using Normal Flow and Texture Regularity*. In Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 05), pages 223–230, 2005. 37, 181, 183
- [Péteri 10a] R. Péteri. *Tracking Dynamic Textures using a Particle Filter Driven by Intrinsic Motion Information*. Machine Vision and Applications, pages 1–9, 2010. 38
- [Péteri 10b] R. Péteri, S. Fazekas & M.J. Huiskes. *DynTex : A Comprehensive Database of Dynamic Textures*. Pattern Recognition Letters, vol. 31, pages 1627–1632, 2010. 16, 26, 28, 31, 32, 183
- [Rahman 04a] A. Rahman & M. Murshed. *Real-Time Temporal Texture Characterisation using Block based Motion Co-occurrence Statistics*. In International Conference on Image Processing (ICIP 04), pages 1593–1596, 2004. 38, 183
- [Rahman 04b] A. Rahman, M. Murshed & L. Dooley. *Feature Weighting Methods for Abstract Features Applicable to Motion based Video Indexing*. In International Conference on Information Technology : Coding and Computing (ITCC 04), pages 676–, 2004. 38
- [Rahman 05] A. Rahman & M. Murshed. *A Robust Optical Flow Estimation Algorithm for Temporal Textures*. In International Conference on Information Technology : Coding and Computing (ITCC 05), pages 72–76, 2005. 38
- [Rahman 07a] A. Rahman & M. Murshed. *Multiple Temporal Texture Detection using Feature Space Mapping*. In ACM International Conference on Image and Video Retrieval (CIVR 07), pages 417–424, 2007. 39
- [Rahman 07b] A. Rahman & M. Murshed. *A Temporal Texture Characterization Technique Using Block-Based Approximated Motion Measure*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, pages 1370–1382, 2007. 38
- [Ravichandran 09] A. Ravichandran, R. Chaudhry & R. Vidal. *View-Invariant Dynamic Texture Recognition using a Bag of Dynamical Systems*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 09), pages 1651–1657, 2009. 43
- [Saisan 01] P. Saisan, G. Doretto, Y.N. Wu & S. Soatto. *Dynamic Texture Recognition*. In Conference on Computer Vision and Pattern Recognition (CVPR 01), pages 58–63, 2001. 24, 25, 30, 41, 43, 183
- [Schodl 00] A. Schodl, R. Szeliski, D.H. Salesin & I. Essa. *Video Textures*. In Annual Conference on Computer Graphics and Interactive Techniques, pages 489–498, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 45, 46, 183

- [Smeulders 00] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta & R. Jain. *Content-based Image Retrieval at the end of the Early Years*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pages 1349–1380, 2000. 155
- [Smith 02] J.R. Smith, C.Y. Lin & M. Naphade. *Video Texture Indexing using Spatio-Temporal Wavelets*. In IEEE International Conference on Image Processing (ICIP 02), pages 437–440, 2002. 14, 15, 48, 60, 62, 63, 146, 149, 150, 184
- [Soatto 01] S. Soatto, G. Doretto & Y.N. Wu. *Dynamic Textures*. In IEEE International Conference on Computer Vision (ICCV 01), pages 439–446, 2001. 25, 41, 43
- [Starck 00] J.L. Starck, E. Candès & D.L. Donoho. *The Curvelet Transform for Image Denoising*. IEEE Transactions on Image Processing, vol. 11, pages 670–684, 2000. 64, 65, 74
- [Starck 04] J.L. Starck, M. Elad & D.L. Donoho. *Redundant Multiscale Transforms and their Application for Morphological Component Analysis*. Advances in Imaging and Electron Physics, vol. 132, 2004. 14, 118, 119, 121, 123, 127, 186
- [Starck 05] J.L. Starck, M. Elad & D.L. Donoho. *Image Decomposition via the Combination of Sparse Representations and a Variational Approach*. IEEE Transactions on Image Processing, vol. 14, pages 1570–1582, 2005. 117, 118, 121, 123
- [Szummer 96a] M. Szummer. *Temporal Texture Modeling*. PhD thesis, MIT, 1996. 40
- [Szummer 96b] M. Szummer & R.W. Picard. *Temporal Texture Modeling*. In IEEE International Conference on Image Processing (ICIP 96), pages 823–826, 1996. 29, 40, 183
- [Vese 02] L.A. Vese & S.J. Osher. *Modeling Textures with Total Variation Minimization and Oscillating Patterns in Image Processing*. Journal of Scientific Computing, vol. 19, pages 553–572, 2002. 118
- [Vishwanathan 07] S. V. Vishwanathan, A.J. Smola & R. Vidal. *Binet-Cauchy Kernels on Dynamical Systems and its Application to the Analysis of Dynamic Scenes*. Computer Vision, vol. 73, pages 95–119, 2007. 42
- [Wang 03] Y. Wang & S. Zhu. *Modeling Textured Motion : Particle, wave and Sketch*. In IEEE International Conference on Computer Vision (ICCV 03), pages 213–220, 2003. 24, 43
- [Willett 03] R.M. Willett & R.D. Nowak. *Platelets : A Multiscale Approach for Recovering Edges and Surfaces in Photon-Limited Medical Imaging*. IEEE Transactions on Medical Imaging, vol. 22, pages 332–350, 2003. 64
- [Woiselle 09] A. Woiselle, J.L. Starck & J.M. Fadili. *Inpainting with 3D Sparse Transforms*. In 22ème édition du colloque GRETSI, 2009. 126
- [Woolfe 06] F. Woolfe & A.W. Fitzgibbon. *Shift-Invariant Dynamic Texture Recognition*. In IEEE European Conference on Computer Vision (ECCV 06), pages 549–562, 2006. 25, 42

- [Wu 01] P. Wu, Y.M. Ro, C.S. Won & Y. Choi. *Texture Descriptors in MPEG-7*. In International Conference on Computer Analysis of Images and Patterns (CAIP 01), pages 21–28, 2001. 11
- [Ying 05] L. Ying, L. Demanet & E. Candès. *3D Discrete Curvelet Transform*. In International Society for Optical Engineering (SPIE 05), 2005. 75
- [Yuan 04] L. Yuan, F. Wen, C. Liu & H.Y. Shum. *Synthesizing Dynamic Texture with Closed-Loop Linear Dynamic System*. In IEEE European Conference on Computer Vision (ECCV 04), pages 603–616, 2004. 43
- [Zhao 06] G. Zhao & M. Pietikäinen. *Local Binary Pattern Descriptors for Dynamic Texture Recognition*. In International Conference on Pattern Recognition (ICPR 06), pages 211–214, 2006. 46
- [Zhao 07a] G. Zhao & M. Pietikäinen. *Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pages 915–928, 2007. 46, 47
- [Zhao 07b] G. Zhao & M. Pietikäinen. *Dynamic Texture Recognition Using Volume Local Binary Patterns*. In R. Vidal, A. Heyden & Y. Ma, editors, Dynamical Vision, volume 4358 of *Lecture Notes in Computer Science*, pages 165–177. Springer Berlin / Heidelberg, 2007. 46, 47, 183
- [Zhong 02] J. Zhong & S. Scarlaroff. *Temporal Texture Recognition Model Using 3D Features*. Rapport technique, 2002. 25, 45
- [Zhou 03] X.S. Zhou & T.S. Huang. *Relevance Feedback in Image Retrieval : A Comprehensive Review*. Multimedia Systems, vol. 8, pages 536–544, 2003. 155
- [Zhou 09] G. Zhou, N. Dong & Y. Wang. *Non-Linear Dynamic Texture Analysis and Synthesis Using Constrained Gaussian Process Latent Variable Model*. In Pacific-Asia Conference on Circuits, Communications and Systems (PACCS 09), pages 27–30, 2009. 43

Décompositions spatio-temporelles pour l'étude des Textures Dynamiques Contribution à l'indexation vidéo

Résumé : Nous nous intéresserons dans cette thèse à l'étude et la caractérisation des Textures Dynamiques (TDs), avec comme application visée l'indexation dans de grandes bases de vidéos.

Ce thème de recherche étant émergent, nous proposons une définition des TDs, une taxonomie de celles-ci, ainsi qu'un état de l'art. La classe de TD la plus représentative est décrite par un modèle formel qui considère les TDs comme la superposition d'ondes porteuses et de phénomènes locaux.

La construction d'outils d'analyse spatio-temporelle adaptés aux TDs est notre principale contribution. D'une part, nous montrons que la transformée en curvelets 2D+T est pertinente pour la représentation de l'onde porteuse. D'autre part, dans un objectif de décomposition des séquences vidéos, nous proposons d'utiliser l'approche par Analyse en Composantes Morphologiques. Notre contribution consiste en l'apport et l'étude de nouvelles stratégies de seuillage. Ces méthodes sont testées sur plusieurs applications : segmentation spatio-temporelle, décomposition de TDs, estimation du mouvement global d'une TD, ...

Nous avons de plus montré que l'Analyse en Composantes Morphologiques et les approches multi-échelles donnent des résultats significatifs pour la recherche par le contenu et l'indexation de Textures Dynamiques de la base de données DynTex.

Cette thèse constitue ainsi un premier pas vers l'indexation automatique de textures dynamiques dans des séquences d'images, et ouvre la voie à de nombreux développements sur ce sujet nouveau. Enfin, le caractère générique des approches proposées permet d'envisager leurs applications dans un cadre plus large mettant en jeu par exemple des données 3D.

Mots clés : Textures Dynamiques, décompositions multi-échelles 2D+T, Analyse en Composantes Morphologiques, indexation vidéo.

Spatio-temporal decompositions for the study of Dynamic Textures Contribution to video indexing

Summary : This report is focused on the study and the characterization of Dynamic Textures (DTs), with the aim of video indexing in large databases.

This research topic being new and emerging, we propose a taxonomy, a definition of DTs and a state of the art. The most representative DT class is described by a model that considers DTs as the superposition of several wavefronts and local oscillating phenomena.

The design of spatio-temporal analysis tools adapted to DT is our main contribution. We first show that the 2D+T curvelet transform is relevant for representing wavefronts. In order to analyse and better understand the DTs, we propose in a second step to adapt the Morphological Component Analysis approach using new thresholding strategies. These methods are tested on several applications : decomposition of DTs, spatio-temporal segmentation, global motion estimation of a DT, ...

We have shown that Morphological Component Analysis and multi-scale approaches enable significant results for content-based retrieval applications and dynamic texture indexing on the DynTex database.

This thesis constitutes a first step towards automatic indexing of DTs in image sequences and opens the way for many new developments in this topic. Moreover, the proposed approaches are generic and could be applied in a broader context, for instance the processing of 3D data.

Keywords : Dynamic textures, video indexing, multiscale 2D+T transform, morphological component analysis.

**Laboratoire Informatique,
Image et Interaction**
Avenue Michel Crépeau
17042 La Rochelle Cedex 01

**Laboratoire Mathématiques,
Image et Applications**
Avenue Michel Crépeau
17042 La Rochelle Cedex 01