

Université Mohamed Khider – Biskra
Faculté des Sciences et de la technologie
Département : Génie électrique
Ref :



جامعة محمد خيضر بسكرة
كلية العلوم و التكنولوجيا
قسم:.....
المرجع:.....

Thèse présentée en vue de l'obtention
Du diplôme de
Doctorat en sciences en : Electronique
Spécialité (Option) : Architecture des systèmes

Intitulé
Estimation hiérarchique du mouvement par ondelettes géométriques

Présentée par :
Fella CHARIF

Soutenue publiquement le 08/01/2015

Devant le jury composé de :

Dr. BAARIR Zineddine	Maitre de Conférences 'A'	Président	Université de Biskra
Dr. DJEDI Nouredine	Professeur	Rapporteur	Université de Biskra
Dr. BOUDHEN Toufik	Maitre de Conférences 'A'	Examineur	Université de Jijel
Dr. LACHOURI Abderrazak	Maitre de Conférences 'A'	Examineur	Université de Skikda

*A la mémoire de ma sœur Fadhila,
je dédie ce travail*

Remerciements

Je veux tout d'abord exprimer mes vifs remerciements au Pr. Noureddie DJEDI pour la confiance sans faille qu'il m'a accordé et pour son aide scientifique précieuse durant la préparation de ce travail de recherche.

Mes remerciements vont également aux membres de jury, Dr. Zineeddine BAARIR, maître de conférence à l'université de Biskra, pour l'honneur qu'il m'a fait en présidant le jury, Dr. Abderrazak LACHOURI, maître de conférence à l'université de Skikda et Dr. Toufik BOUDEN, maître de conférence à l'université de Jijel, d'avoir accepter d'examiner et d'évaluer ce travail.

Je tiens à remercier spécialement Dr. Foudhil CHARIF, Pr. J. BARRON, Pr. Abdelmalik AHMED-TALEB pour leurs aides.

A titre plus personnel, je remercie chaleureusement mon mari, Abderrazak BENCHABANE, pour son encouragement, son soutien ininterrompu et ses nombreux conseils tout le long de ma thèse.

Finalement je remercie mes parents, mes frères, mes sœurs, mes deux petits fils Abderraouf et Abderrahim ainsi que ma belle famille pour leur soutien qui m'a été bien utile durant ma thèse.

Résumé

La méthode de Horn & Schunck et la méthode de Lucas & Kanade sont considérées comme les méthodes les plus faciles à mettre en œuvre. Ce sont en effet les deux techniques les plus largement employées dans la communauté scientifique dans le cadre de l'étude du mouvement. Cependant, celles-ci souffrent de défauts qui leur sont propres ; ainsi, la méthode de Horn & Schunck non parallélisable le rend peu adapté aux applications en temps réel, tandis que la méthode de Lucas & Kanade est inadaptée lorsqu'il s'agit de traiter de grands déplacements. Il est toutefois possible de pallier ce dernier problème en utilisant une approche multi-échelles basée sur le pyramide Laplacien, Gaussien ou la transformée en ondelettes. Donc dans un problème d'estimation de mouvement, deux performances critiques sont à améliorer, le temps de calcul et la précision des résultats. Dans la présente thèse, et suite à la nature parallèle des réseaux de neurones et leur aptitude à résoudre des problèmes complexes en temps réel, on a proposé d'introduire les réseaux de neurones récurrents dans le problème de l'estimation de mouvement utilisant la méthode de Horn & Schunck reformulée par Metiche & Mansouri et la méthode de Lucas & Kanade. Pour la précision des résultats, on a introduit la transformée en curvelettes pour l'estimation multi-échelles de la méthode de Lucas & Kanade. Pour améliorer d'avantage les performances de la méthode de Lucas & Kanade multi-échelles, un préfiltrage utilisant la transformée en curvelettes a été également utilisé. Les résultats obtenus sur des séquences de test synthétiques et réelles montrent que les méthodes proposées nous permettent une nette amélioration de point de vue précision et rapidité. On a pu montrer que deux itérations sont suffisantes pour que le réseau de neurones atteigne une erreur de convergence très petite. De même l'introduction de la transformée en curvelettes dans l'estimation de mouvement multi-échelles nous permet d'aboutir à des résultats comparables avec des travaux récents. Pour achever cette thèse, une extension des deux méthodes proposées aux séquences d'images tridimensionnelles a été proposée.

Mots clés : Estimation de mouvement, transformée en curvelettes, Méthode de Lucas & Kanade, méthode de Horn & Schunck, Réseau de neurones de Hopfield, Réseau de neurones de Zhang.

Abstract

The Horn & Schunck and the Lucas & Kanade methods are considered as the easiest methods to be implemented. These are indeed the two techniques most widely used in the scientific community in the context of the study of motion. However, they suffer from their own shortcomings; the Horn & Schunck method not parallelizable makes it ill-suited to real-time applications, while the Lucas & Kanade method is inadequate when it comes to dealing with large displacements. However, it is possible to overcome the latter problem using a multi-scale approach based on the Laplacian, Gaussian pyramid, or the wavelet transform. So in the motion estimation problem, two critical performances need to be improved, the computation time and the accuracy of the results. In this thesis, and due to the parallel nature of neural networks and their ability to solve complex problems in real time, we have proposed to introduce the recurrent neural networks in the motion estimation problem using the Horn & Schunck method reformulated by Metiche & Mansouri, and the Lucas & Kanade method. For accuracy of the results, we have introduced the curvelet transform for the multi-scale estimation method of Lucas & Kanade. To more improve the performance of the multi-scale method, a prefiltering using the curvelet transform was also used. The results obtained on synthetic and real sequences show that the proposed methods allow us a clear improvement of the accuracy and the fastness. It has been found that two iterations are sufficient for the neural network to achieve a very small convergence error. Similarly the introduction of the curvelet transform in the multi-scale motion estimation allows us to achieve comparable results with recent work. To complete this thesis, an extension of the two proposed methods to the three-dimensional images sequences was proposed.

Key-words: Motion estimation, Curvelet transform, Lucas & Kanade method, Horn & Schunck method, Hopfield neural network, Zhang neural network.

ملخص

إن تقدير الحركة في سلاسل الصورة ثنائية و ثلاثية الأبعاد تعد من أهم مجالات الرؤية بالحاسوب, حيث نجدها في عدة مجالات منها مراقبة إشارة المرور الضوئية , التحكم في حركة الإنسان الآلي , التشخيص الطبي , ضغط سلاسل الصورة ... إلخ. تعتبر طريقة هورن و شانك وطريقة ليكاس و كناد من أسهل طرق تقدير الحركة القابلة للتنفيذ. فهما في الواقع الطريقتين الأكثر استخداماً على نطاق واسع في الأوساط العلمية في سياق دراسة الحركة. ومع ذلك، فإنهما تعانيان من أوجه القصور الخاصة بهما. فطريقة هورن و شانك الغير قابلة للتفرع يجعلها غير مناسبة للتطبيقات في الوقت الحقيقي ، في حين أن طريقة ليكاس و كناد غير كافية عندما يتعلق الأمر بالتعامل مع إزاحات كبيرة . ومع ذلك، فمن الممكن التغلب على المشكلة الأخيرة باستخدام نهج متعدد السلم المتعلق بهرم لابلاس أو غوس أو تحويل الموجات . مما لا شك فيه أن مشكلة تقدير الحركة تتعلق بتحسين عاملين أساسيين و هما الوقت اللازم للحساب و دقة النتائج . في هذه الأطروحة ، و نظراً لطبيعة الشبكات العصبية المتوازية و قدرتها على حل المشاكل المعقدة في الوقت الحقيقي، تم اقتراح إدخال الشبكات العصبية المتكررة في مشكلة تقدير الحركة باستخدام طريقة هورن و شانك المعدلة من طرف متيش و منصور و طريقة ليكاس و كناد. أما بالنسبة لدقة النتائج، فقد تم اقتراح استخدام التحويل المنحني الذي فرض وجوده يوماً بعد يوم في مجال معالجة الصورة لصلابة أسسه النظرية و سهولة استعماله. لهذا جاء عملنا ليرز مدى نجاعة وكفاءة هذه التقنية في تقدير حركة سلاسل الصورة . بحيث في المرة الأولى أدرجناها كوسيلة لبناء هرمي للصورة عوض تحويل لابلاس المستعمل في طريقة ليكاس و كناد المتعددة السلاسل لتحسين أداءها. و في المرة الثانية استخدمناها لترشيح الصورة قبل تقدير الحركة بطريقة ليكاس و كناد المتعددة السلاسل. النتائج التي تم الحصول عليها على سلاسل الصورة المركبة و الحقيقية تبين أن الطرق المقترحة أدت إلى تحسن واضح في دقة النتائج مقارنة مع النتائج المتحصل عليها انطلاقاً من الطرق الأصلية مع توفير سرعة فائقة في التقدير خاصة باستخدام شبكة زهناق التي تتقارب في تكرارين فقط. هذه النتائج المشجعة أدت إلى تمديد الطرق المقترحة لاستخدامها في تقدير الحركة في سلاسل الصور ثلاثية الأبعاد.

كلمات مفاتيح: تقدير الحركة، التحويل المنحني ، طريقة ليكاس و كناد ، طريقة هورن و شانك ، الشبكة العصبية لهوبفيلد، الشبكة العصبية لزهنق .

Liste des figures

1.1. Projection du mouvement 3D en 2D.....	7
1.2. Exemple de flot optique.....	8
1.3. Sphère uniforme immobile et en rotation.....	8
1.4. Principe d’occlusion.....	10
1.5. Principe d’ouverture.....	10
1.6. Exemple d’une procédure d’appariement de blocs.	18
1.7. Sous-ensembles de recherche.....	20
2.1 Fonctions d’approximation.....	22
2.2. Translation et dilatation des ondelettes.....	24
2.3. Grille dyadique.....	25
2.4. Algorithme Mallat pour la décomposition et la reconstruction d’un signal.....	27
2.5. Schéma de la TO non décimée en utilisant l’algorithme à trous.	28
2.6. Arbre de décomposition par la transformée en paquets d’ondelettes.....	29
2.7. Algorithme de Mallat pour la décomposition et la reconstitution d’une image.....	30
2.8. Décomposition multi niveaux par ondelettes.....	31
2.9. Exemples d’ondelettes de Daubechies pour différents ordres.....	31
2.10. Décomposition multi-résolutions sur différentes bases d’ondelettes.....	31
2.11. Classification des ondelettes géométriques.	32
2.12. Construction de la transformée en ridgelettes par le théorème de projection des coupes.	34
2.13. Principe de la transformée en curvelettes.....	35
2.14. Décomposition en curvelettes d’une image en 3 niveaux.....	36
2.15. Schémas de décomposition (a) et reconstruction (b) pyramide laplacienne	36
2.16. Pavage du plan fréquentiel obtenu par la décomposition pyramidale laplacienne.....	37
2.17. Banc de filtres directionnel et son pavage fréquentiel.....	37
2.18. Principe de la transformée en contourlettes.....	38
2.19. Découpage du plan fréquentiel pour la transformée en contourlettes.....	38
2.20. Exemple de transformée en contourlettes d’une image.....	39

2.21. Trois étapes du seuillage des coefficients en ondelettes.....	41
2.22. Quelques fonctions de seuillages.....	43
3.1. Schéma synoptique de l'estimateur hiérarchique de Lucas & Kanade.....	49
3.2. Architecture du réseau de Hopfield.....	53
3.3. Architecture du réseau de neurones de Zhang discret.....	54
3.4. Courbe de la fonction signe-puissance pour différentes valeur de r	55
3.5. Schéma synoptique de l'estimateur multi-échelles proposé.....	57
3.6. Colormap et affichage du même champ de vitesse sous forme de vecteurs.....	59
3.7. Influence des paramètres.....	60
3.8. Résultats d'une section diagonale de la séquence 'arbre divergente'	61
3.9. Performance de la méthode proposée pour la séquence 'Yosemite'	61
3.10. Performance de la méthode pour la séquence 'Sinusoïde' (EAM=0.5661° et ET= 0.0239°).....	61
3.11. Performance de la méthode pour la séquence 'arbre divergente' (EAM=6.7731° et ET= 3.2573°).....	62
3.12. Performance de la méthode pour la séquence 'arbre en translation' (EAM=0.9668° et ET= 0.2287°).....	62
3.13. Courbe de convergence pour la séquence 'Yoesmite'	63
3.14. Courbe de convergence pour la séquence 'arbre en translation'	63
3.15. Décomposition en curvelettes de l'image 'Hydrangea' en 2 niveaux.....	64
3.16. Débruitage de l'image 'Hydrangea'	65
3.17. Résultat du flot optique de la séquence 'Hydrangea' sans bruit.....	65
3.18. Erreur angulaire de la séquence 'Hydrangea'	65
3.19. Frame de la séquence 'Yosemite' et son flot optique réel.....	66
3.20. Flot optique estimé de la séquence 'Yosemite' sans bruit.....	66
3.21. Flot optique estimé de la séquence 'Yosemite' bruitée.....	66
3.22. Erreur angulaire de la séquence 'Yosemite'	67
3.23. Erreur angulaire pour différents niveaux de résolution.....	67
3.24. Erreur absolue pour différents niveaux de résolution	68
3.25. Séquences réelles de test.....	69
3.26. Flot estimé de la séquence 'Taxi de Humberg'	69
3.27. Flot estimé de la séquence Taxi de 'Rubic Cube'	70

4.1. Architecture du réseau DZNN1-1.....	75
4.2. Architecture du réseau RNZD2.....	77
4.3. Architecture du réseau pour résoudre le système.....	77
4.4. Schéma synoptique pour le calcul du flot optique par DZNN1-1.....	78
4.5. Schéma synoptique pour le calcul du flot optique par DZNN1-2.....	78
4.6. Schéma synoptique pour le calcul du flot optique par DZNN2.....	79
4.7. Complexité et mémoire de stockage.	80
4.8. Schéma d'implémentation du réseau DZNN1-1.	81
4.9. Schéma d'implémentation d'un bloc du réseau DZNN1-2.	81
4.10. Schéma d'implémentation du réseau DZNN2.	82
4.11. Effet du paramètre de lissage sur le comportement du réseau DZNN1-1.	82
4.12. Effet du paramètre de lissage sur le comportement du réseau DZNN1-2.	83
4.13. Effet du paramètre de lissage sur le comportement du réseau DZNN2.	83
4.14. Effet du paramètre de lissage sur la méthode de Jacobi.	83
4.15. Effet du paramètre de lissage sur la méthode originale de Horn & Schunck.....	84
4.16. Flot optique estimé par les différentes méthodes.....	85
4.17. Résultats de la séquence 'Yosemite' utilisant le réseau DZNN2.	86
4.18. Erreur de convergence pour la séquence 'Yosemite'.....	86
4.19. Résultats de la séquence 'Dimetrodon' utilisant le réseau DZNN2.....	86
4.20. Résultats de la séquence 'Grove3' utilisant le réseau DZNN2.....	87
4.21. Résultats de la séquence 'Grove2' utilisant le réseau DZNN2.....	87
4.22. Résultats de la séquence 'Dimetrodon' utilisant le réseau DZNN2.....	87
4.23. Frame de la séquence 'Taxi de Humbourg'.....	88
4.24. Performance du réseau pour la zone spécifiée.....	88
4.25. Résultats de la séquence 'Taxi de Humbourg'.....	89
5.1 Représentation des voisinages d'un voxel.	91
5.2. Image 3D de la séquence synthétique.....	96
5.3. 12 coupes horizontales de la séquence synthétique.	96
5.4. Coupes frontales selon x et y de la séquence synthétique.....	96
5.5. 12 coupes horizontales de la séquence réelle à l'instant $t = 2$	97
5.6. 12 coupes horizontales de la séquence réelle à l'instant $t = 3$	97

5.7. Coupe horizontale 3 pour différents instants.	98
5.8. Coupe horizontale 10 pour différents instants.	98
5.9. Coupes frontales selon x et y de la séquence réelle.....	98
5.10. Contours des projections selon x , y et z	99
5.11. Dérivées spatiotemporelle.....	100
5.12. Courbes de convergence du DZNN2-LKM3D pour la séquence synthétique.....	101
5.13. Flot optique 3D estimé par la méthode de LKM3D pour la séquence synthétique.....	101
5.14. Flot optique estimé des coupes horizontales 6 et 8 par la méthode de LKM3D pour la séquence synthétique.....	102
5.15. Projection du flot optique estimé par la méthode de LKM3D selon l'axe x et l'axe y pour la séquence synthétique.	102
5.16. Courbes de convergence du DZNN2-HSM3D pour la séquence synthétique.....	103
5.17. Flot optique 3D estimé par la méthode de HSM3D pour la séquence synthétique.....	103
5.18. Flot optique estimé des coupes horizontales 6 et 8 par la méthode de HSM3D pour la séquence synthétique.....	103
5.19. Projection du flot optique estimé par la méthode de HSM3D selon l'axe x et l'axe y pour la séquence synthétique.	104
5.20. Courbes de convergence du DZNN2-LKM3D pour la séquence réelle.....	105
5.21. Flot optique 3D estimé par la méthode de LKM3D pour la séquence réelle.	105
5.22. Flot optique estimé des coupes horizontales 3 et 10 par la méthode de LKM3D pour la séquence réelle.....	105
5.23. Projection selon x et y du flotoptique estimé par la méthode de LKM3D pour la séquence réelle.	106
5.24. Courbes de convergence du DZNN2-HSM3D pour la séquence réelle.....	106
5.25. Flot optique 3D estimé par la méthode de HSM3D pour la séquence réelle.....	106
5.26. Flot optique estimé des coupes horizontales 3 et 10 par la méthode de HSM3D pour la séquence réelle.....	107
5.27. Projection du flot optique estimé par la méthode de HSM3D selon l'axe x et l'axe y pour la séquence réelle.....	107
5.28. Evaluation du PSNR pour différentes coupes des deux méthodes proposée.....	108

A.1. Modèle de neurone formel.....	111
A.2. Schéma d'un réseau de neurones.....	112
A.3. Réseaux de neurones les plus utilisés.	113
A.4. Structure du réseau de Hopfield.....	113
B.1. Gradient de Simoncelli.....	116
B.2. Calcul des dérivées partielles de Simoncelli	116

Liste des tableaux

3.1. Performance de la méthode modifiée.....	62
3.2. Comparaison de performance entre la méthode originale et la méthode modifiée...	62
3.3. Comparaison quantitative avec d'autres méthodes.....	64
3.4. Comparaison quantitative avec la méthode de Flopis et Sachez	68
4.1. EPE et AAE de quelques séquences de test de la base de données Middlebury	87
B.1. Masques proposés par Simoncelli dans [99].....	115

Table des matières

Remerciements	i
Résumé	ii
Abstract	iii
ملخص	iv
Liste des figures	v
Liste des tableaux	x
Table des matières	xi
Introduction générale	1
Chapitre 1	
Etat de l’art sur l’estimation du mouvement	
	6
1.1. Introduction.....	6
1.2. Séquence vidéo numérique.....	6
1.3. Mouvement dans une séquence d’images	6
1.3.1. Mouvement réel	7
1.3.2. Mouvement apparent	7
1.3.3. Différence entre mouvement réel et mouvement apparent.....	8
1.4. Estimation de mouvement	8
1.4.1. Principales difficultés.....	8
1.4.1.1. Présence du bruit.....	9
1.4.1.2. Ambiguïté due à la projection.....	9
1.4.1.3. Problème des régions homogènes.....	9
1.4.1.4. Frontières d’occlusion.....	9
1.4.1.5. Problème d’ouverture.....	10
1.4.2. Contrainte de l’estimation de mouvement.....	10
1.4.3. Équation du flot optique	11
1.5. Panorama des méthodes d’estimation du mouvement.....	12
1.5.1. Méthodes différentielles.....	12
1.5.1.1. Méthodes variationnelles globales.....	13
1.5.1.2. Méthodes variationnelles locales.....	13

1.5.1.3. Méthodes hybrides.....	13
1.5.2. Méthodes fréquentielles	13
1.5.2.1. Méthode de corrélation.....	14
1.5.2.2. Méthode de corrélation de phase.....	15
1.5.2.3. Estimation du mouvement par banc de filtres de Gabor.....	16
1.5.2.4. Approches par ondelettes.....	17
1.5.3. Méthode d'appariement de blocs	18
1.5.3.1. Largeur de la zone de recherche.....	19
1.5.3.2. Taille des blocs.....	19
1.5.3.3. Types de fonction de coût.....	19
1.5.3.4. Types de recherches	20
1.6. Discussion sur les différentes méthodes.....	20
1.7. Conclusion.....	21
Chapitre 2	
Analyse multi-échelles par ondelettes géométriques	
	22
2.1. Introduction	22
2.2. Transformée en ondelettes	23
2.2.1. Définition.....	23
2.2.2. Transformée en ondelettes continue.....	24
2.2.3. Transformée en ondelettes discrètes.....	24
2.2.4. Analyse multi-résolutions et transformée en ondelettes.....	25
2.2.5. Algorithme rapide de Mallat	26
2.2.6. Ondelettes non décimées.....	28
2.2.7. Paquets d'ondelettes.....	29
2.2.8. Transformée en ondelettes bidimensionnelles.....	29
2.2.9. Exemple d'ondelettes.....	31
2.3. Ondelettes géométriques 2D.....	31
2.3.1. Transformée en ridgelettes.....	32
2.3.1.1. Principe de la transformée en ridgelettes.....	32
2.3.1.2. Algorithme de la transformée en ridgelettes	34
2.3.2. Transformée en curvelettes.....	34
2.3.2.1. Principe de la transformée en curvelettes	34
2.3.2.2. Algorithme de la transformée en curvelettes.....	35

2.3.3. Transformée en courbelettes.....	36
2.3.3.1. Décomposition en sous-bandes.....	36
2.3.3.2. Filtrage directionnel.....	37
2.3.4. Pouvoir d'approximation.....	39
2.4. Débruitage d'images par seuillage	39
2.4.1. Algorithme de débruitage par seuillage	40
2.4.2. Méthodes de seuillage	41
2.4.2.1. Seuillage doux.....	41
2.4.2.2. Seuillage dur.....	41
2.4.2.3. Seuillage de Breiman.....	41
2.4.2.4. Seuillage de Bruce et Gao.....	42
2.4.2.5. Seuillage de Kwon.....	42
2.4.2.6. Seuillage de Zhang.....	42
2.4.2.7. Seuillage de Mehdi <i>et al.</i>	42
2.4.3. Sélection du seuil.....	43
2.4.3.1. Seuillage global.....	43
2.4.3.2. Seuillage dépendant du niveau.....	44
2.4.3.3. Seuillage dépendant du nœud.....	44
2.5. Conclusion.....	44

Chapitre 3

Méthode de Lucas & Kanade multi-échelles par la transformée en courbelettes	45
3.1. Introduction	45
3.2. Approches existantes de LK.....	45
3.2.1. Méthode de LK originale	45
3.2.2. Méthode pyramidale de Lucas & Kanade	48
3.3. Approches proposées.....	49
3.3.1. Accélération de la méthode LKM.....	50
3.3.2. Méthode neuronale de Lucas & Kanade	50
3.3.2.1. Reformulation de la méthode de LK	50
3.3.2.2. Réseau de Hopfield.....	51
3.3.2.3. Réseau de neurones de Zhang	53
3.3.2.4. Algorithme de Lucas & Kanade neuronal (LKN)	55
3.3.3. Débruitage par la transformée en courbelettes.....	55

3.3.4. Estimateur multi-échelles par la transformée en curvelettes.....	56
3.3.4.1. Principe de l'estimateur	56
3.3.4.2. Algorithme final de la méthode proposée.....	57
3.4. Résultats des simulations.....	58
3.4.1. Séquences de test et critères d'évaluation.....	58
3.4.1.1. Séquences de test	58
3.4.1.2. Critères d'évaluation.....	58
3.4.2. Représentation du flot optique.....	59
3.4.3. Sélection des paramètres	60
3.4.4. Stratégie 1 (masque d'interpolation)	60
3.4.5. Stratégie 2 (Réseaux de neurones)	63
3.4.6. Stratégie 3 (débruitage par TC)	64
3.4.7. Stratégie 4 (approche multi-résolutions)	67
3.4.7.1. Comparaison quantitative avec d'autres algorithmes.....	68
3.4.7.2. Applications aux séquences réelles.....	68
3.5. Conclusion.....	70

Chapitre 4

Implémentation parallèle de la méthode de Horn & Schunck	71
4.1. Introduction.....	71
4.2. Méthode de Horn & Schunck.....	71
4.3. Inversion de la matrice A par le réseau de Zhang.....	73
4.3.1. Réseau de neurones à dynamique explicite.....	73
4.3.1.1. Principe.....	73
4.3.1.2. Discrétisation du réseau.....	74
4.3.1.3. Implémentation du réseau RNZD1-1	74
4.3.2. Réseau de neurones à dynamique implicite.....	76
4.3.2.1. Principe.....	76
4.3.2.2. Implémentation du réseau DZNN1-2.....	76
4.4. Résolution du système $Az = b$ par le réseau DZNN2.....	77
4.5. Implémentation de l'approche proposée	78
4.6. Complexité de l'algorithme.....	79
4.7. Résultats de simulation.....	81
4.7.1. Schémas d'implantation des réseaux.....	81

4.7.2. Réglage du paramètre de lissage.....	82
4.7.3. Résultats des séquences synthétiques.....	84
4.7.4. Résultats des séquences réelles.....	88
4.8. Conclusion.....	89

Chapitre 5

Extension à l'estimation de mouvement 3D	90
5.1. Introduction.....	90
5.2. Extension tridimensionnelle	90
5.2.1. Equation du flot optique 3D.....	90
5.2.2. Voisinage d'un voxel	91
5.3. Méthode de Lucas & Kanade 3D.....	91
5.3.1. Principe.....	91
5.3.3. Méthode de Lucas & Kanade 3D modifiée.....	92
5.4. Méthode de Horn & Schunck 3D.....	92
5.4.1. Principe	92
5.4.2. Méthode de Horn & Schunck 3D modifiée.....	94
5.4.2.1. Reformulation de la méthode de HS 3D	94
5.4.2.2. Convergence de l'algorithme proposé.....	95
5.5. Expérimentations et résultats.....	96
5.5.1. Séquences de test.....	96
5.5.1.1. Séquence synthétique.....	96
5.5.1.2. Séquence réelle médicale.....	97
5.5.2. Dérivées spatio-temporelles	100
5.5.3. Evaluation du flot optique estimé.....	101
5.5.3.1. Séquence synthétique	101
5.5.3.2. Séquence réelle.....	104
5.6. Conclusion	108
Conclusion générale	109

Annexe A

Aperçu sur les réseaux de neurones artificiels	111
A.1. Introduction.....	111
A.2. Neurone formel.....	111
A.3. Réseaux de neurones artificiels.....	112

A.4. Principaux types des réseaux de neurones.....	112
A.5. Réseaux de neurones récurrents pour l'optimisation.....	113
A.6. Réseaux de neurones de Hopfield.....	113
A.7. Principes de fonctionnement du réseau de Hopfield pour l'optimisation.....	114
A.9. Conclusion.....	114
Annexe B	
Gradient de Simoncelli	115
Bibliographie.....	117

Introduction générale

Le problème de l'estimation de mouvement consiste à extraire le mouvement apparent des objets d'une scène tridimensionnelle (3D) à partir d'une séquence d'images. On cherche alors à percevoir la projection des différents mouvements 3D de la scène sur le plan image. Le mouvement apparent appelé aussi flot optique est une variable cachée et n'est accessible que par l'analyse des variations temporelles de la séquence d'images. En d'autres termes, le flot optique peut être défini comme le champ de vitesse (ou de déplacement) produit dans le plan image par des objets en mouvement dans un espace 3D [1].

Il peut arriver des cas où le mouvement observé ne reflète pas toujours le mouvement réel 2D, un exemple de mouvement réel projeté dans le plan image et non observable est le cas d'un disque homogène tournant autour de l'axe passant par son centre et orthogonal au plan image. De même, la présence de variations de luminances dues au changement d'éclairage de la scène engendre un mouvement apparent même si le mouvement réel n'existe pas. En plus, le mouvement apparent est une combinaison des mouvements des objets dans la scène et de celui dû au déplacement de la caméra. Tous ces facteurs rendent l'analyse du contenu dynamique intrinsèque de la scène plus délicate [1].

L'estimation de mouvement est très rencontrée dans une grande diversité d'applications. Elle peut être utilisée pour des fins d'observation comme la détection de mouvement pour la surveillance, l'estimation de la densité du trafic routier, le suivi d'objets... Par ailleurs, l'estimation du mouvement est utilisée pour prédire les images suivantes, c'est le cas de la compression vidéo par compensation de mouvement [2].

Bien que le problème de l'estimation de mouvement ait constitué la préoccupation d'un grand nombre de chercheurs depuis l'avènement de la vision artificielle, il demeure toujours d'actualité, puisque aucune méthode universelle n'a été élaborée pour permettre de déterminer le mouvement dans toutes les situations et pour tous les types d'objets en mouvement.

Dès son apparition, l'approche proposée par Horn & Schunck [3] a permis au problème de l'estimation de mouvement de connaître un essor notable où une grande diversité

d'algorithmes et de méthodes sont venus enrichir ce domaine. Ces méthodes peuvent être classées en trois catégories principales: les méthodes différentielles, les méthodes de corrélation et les méthodes fréquentielles [4].

Les méthodes différentielles ou variationnelles sont historiquement les premières méthodes de calcul du flot optique. Elles ont l'avantage d'être bâties sur des bases mathématiques clairement établies et de fournir des résultats denses et cohérents. Les méthodes différentielles consistent à résoudre un problème d'optimisation (local ou global) en minimisant une fonctionnelle, généralement basée sur l'équation du flot optique à laquelle on ajoute une contrainte pour particulariser les solutions. Selon le type de la contrainte utilisée, on obtient différentes approches variationnelles [5-6], les approches variationnelles globales consistent à minimiser, sur le domaine entier de l'image, une fonctionnelle prenant en compte l'équation du flot optique ainsi qu'un terme de lissage, c'est-à-dire en ajoutant une contrainte de régularisation portant sur le gradient, le Laplacien (ou ordre supérieur) du champ de vitesse. La plus connue de ces méthodes est certainement celle proposée par Horn & Schunck [3] qui impose une contrainte globale de lissage sur le champ des vitesses. Les approches variationnelles locales consistent à prendre en compte des hypothèses supplémentaires sur un domaine de taille réduite pour particulariser le flot optique. On minimise alors un critère sur un petit domaine donnant ainsi son flot optique. La méthode locale la plus célèbre est celle de Lucas & Kanade [7].

Problématique

La méthode de Horn & Schunck (HS) et la méthode de Lucas & Kanade (LK) sur lesquelles se concentre cette thèse sont, parmi l'ensemble des catégories citées, les plus faciles à mettre en œuvre. Ce sont en effet les deux techniques les plus largement employées par la communauté scientifique dans le cadre de l'étude du mouvement. Cependant, celles-ci souffrent de défauts qui leur sont propres. Ainsi, la méthode de Horn & Schunck non parallélisable la rend peu adaptée aux applications en temps réel, tandis que la méthode de Lucas & Kanade est inadaptée lorsqu'il s'agit de traiter de grands déplacements. Il est toutefois possible de pallier ce dernier problème en utilisant une approche multi-échelles basée sur le pyramide Laplacien, Gaussien ou la transformée en ondelettes [8-10].

La transformée en ondelettes, par sa construction optimale sert à représenter les discontinuités ponctuelles unidimensionnelles [11]. L'extension au domaine 2D est

généralement réalisée par simple produit tensoriel séparable. Malheureusement, la représentation en ondelettes de discontinuités situées le long des courbes lisses est très peu efficace et génère beaucoup de coefficients. Pour remédier à ce problème, la transformée en ridgelettes et la transformée en curvelettes ont été proposés par Candès et Donoho [12-13]. Récemment, ces deux transformées ont été appliquées au domaine du filtrage de bruit gaussien avec des résultats très prometteurs [14].

De nos jours, les réseaux de neurones artificiels deviennent un sujet de recherche très dynamique dû au progrès de la technologie VLSI (Very Large Scale Integration) qui a considérablement simplifié leur implémentation, une tâche qui n'était pas faisable dans le passé. Les réseaux de neurones (RN) sont connus par leur potentiel naturel vis-à-vis des autres méthodes pour les traitements en temps réel dû spécifiquement à leur parallélisme [15]. Récemment, un nouveau type des réseaux dynamiques a été proposé par Y. Zhang *et al.* pour résoudre en temps réel une variété de problèmes statiques ou variables dans le temps [16]. La nouveauté de ce réseau réside dans l'utilisation d'un vecteur/matrice erreur au lieu de la norme de l'erreur utilisée couramment dans les réseaux de neurones basés sur le gradient.

L'estimation de mouvement n'est pas à l'abri de ces techniques, plusieurs recherches ont été élaborées pour l'implémentation des méthodes différentielles utilisant les réseaux de neurones. Dans [17], C. Grava *et al.* ont proposé un réseau cellulaire, tandis que dans [18], H. Li *et al.* ont proposé d'utiliser le réseau de Hopfield pour l'implémentation de la méthode de Horn & Schunck .

Objectif de la thèse

L'objectif fixé au début de cette thèse était de produire des méthodes différentielles d'estimation de mouvement précises, robustes et rapides. Notre contribution consiste, d'une part à introduire les réseaux de neurones récurrents dans le problème de l'estimation de mouvement utilisant la méthode de Horn & Schunck reformulée par Metiche et Mansouri [19] et la méthode de Lucas & Kanade, et d'autre part, d'introduire la transformée en curvelettes (TC) pour l'estimation multi-échelles de la méthode de LK. Pour améliorer d'avantage les performances de la méthode de LK multi-échelles (LKM) [10], un préfiltrage utilisant la transformée en curvelettes a été également utilisé. La figure ci-dessous donne une illustration de la démarche de notre contribution dans la présente thèse.

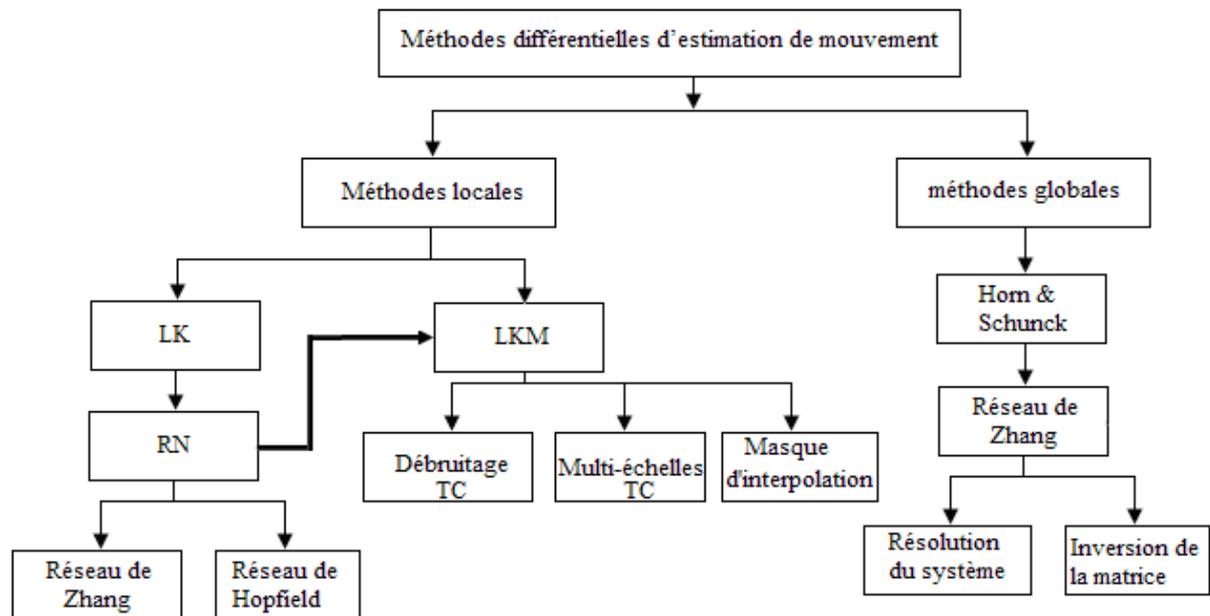


Figure 1. Organigramme de notre contribution pour l'estimation de mouvement.

Pour achever cette thèse, une extension des deux méthodes proposées aux séquences d'images tridimensionnelles a été proposée.

Organisation du manuscrit

Le travail présenté dans cette thèse s'articule autour de l'estimation de mouvement utilisant les réseaux de neurones dynamiques et les ondelettes géométriques. Il est organisé en cinq chapitres :

Chapitre 1 Etat de l'art sur l'estimation du mouvement: Ce chapitre introductif est consacré à un état de l'art sur les méthodes d'estimation de mouvement d'une séquence d'images. Après une introduction sur les définitions et les problèmes généraux de la mesure du mouvement, nous présentons les techniques principales d'estimation de mouvement imposées dans le domaine de la vision par ordinateur. Enfin, une dernière section sera consacrée à une discussion sur certains problèmes soulevés dans ce chapitre.

Chapitre 2 Généralités sur les ondelettes : dans le chapitre 2, nous donnerons une brève discussion sur les définitions et les propriétés des ondelettes. Nous examinerons les nouveaux outils d'analyse multi-résolutions qui sont les ridgelettes, les curvelettes et les contourlettes. Nous donnerons par la suite un aperçu général sur les méthodes de débruitage par seuillage des images corrompues par un bruit additif.

Chapitre 3 Méthode de Lucas & Kanade multi-échelles par la transformée en curvelettes:

Pour augmenter la performance de la méthode multi-échelles de Bougier, nous avons proposé d'utiliser la transformée en ondelettes géométrique soit pour le débruitage des images de la séquence, soit pour l'estimation multi-échelles de la méthode de LK. Pour accélérer l'algorithme du calcul du flot optique, deux réseaux de neurones (réseau de Hopfield et le réseau de Zhang) ont été utilisés. Ces deux contributions constituent l'objet du chapitre 3.

Chapitre 4 Implémentation parallèle de la méthode de Horn & Schunck:

dans le chapitre 4, nous présenterons une nouvelle implémentation qui sera dédiée à l'utilisation d'une version discrète du réseau de Zhang pour l'estimation du flot optique. Celle-ci est réalisée en se basant sur la reformulation de Mitiche de la méthode différentielle de Horn & Schunck qui engendre un système large d'équations linéaires. Pour résoudre ce système, nous proposons trois types de réseaux de neurones DZNN1-1, DZNN1-2 et DZNN2. Ces trois réseaux sont caractérisés par leur rapidité de convergence sans altérer la précision des résultats.

Ces deux derniers chapitres introduisent de nouvelles représentations, et sont accompagnés d'expérimentations montrant leur applicabilité et leur utilité pour l'estimation de mouvement.

Chapitre 5 Extension à l'estimation du mouvement 3D:

dans ce chapitre nous nous intéressons à l'étape d'estimation de mouvements tridimensionnels à partir d'une séquence d'images volumiques. Nous présentons deux extensions tridimensionnelles, une pour la méthode de Horn & Schunck et l'autre pour la méthode de Lucas & Kanade.

Une conclusion générale sur les travaux réalisés lors de cette thèse ainsi que les perspectives terminent le corps de ce manuscrit.

L'annexe A contient un rapport technique concernant les réseaux de neurones.

L'annexe B présente les dérivées spatiotemporelles par le filtre de Simoncelli.

Chapitre 1

Etat de l'art sur l'estimation du mouvement

1.1. Introduction

L'estimation du mouvement consiste à extraire un vecteur de mouvement à partir d'une séquence d'images en faisant l'hypothèse que l'intensité (ou la couleur) est conservée au cours du déplacement. L'estimation de mouvement est exploitée dans plusieurs applications, on la retrouve essentiellement dans la compression vidéo et dans le domaine médical [6].

Ce chapitre introductif est consacré à la description de quelques méthodes usuelles d'estimation du mouvement à partir d'une séquence d'images. Après une brève discussion sur le mouvement et les problèmes généraux de sa mesure, nous présenterons les principales techniques d'estimation imposées dans le domaine de la vision par ordinateur. Enfin, une dernière section est consacrée à une discussion sur certains problèmes soulevés dans ce chapitre.

1.2. Séquence vidéo numérique.

Une image numérique est constituée d'un ensemble de points élémentaires appelés pixels. Le couple des réels (x, y) représente la position spatiale d'un pixel, et la valeur $I(x, y)$ son niveau de gris. Dans une séquence vidéo numérique, deux trames (images) successives présentent de nombreuses similarités caractérisant le fond et les objets en mouvement. Ces trames sont composées d'un même ensemble d'objets. Une trame est une représentation planaire d'une scène ou d'un objet, généralement situé dans un espace tridimensionnel.

1.3. Mouvement dans une séquence d'images

Le mouvement est un réel problème en vidéo puisqu'il décrit un contexte en trois dimensions (3D) alors que les images sont une projection de scènes 3D sur un plan

bidimensionnel comme montré dans la figure (1.1). L'information ainsi obtenue est statique et incomplète. En général, le mouvement est représenté par un vecteur de mouvement qui décrit une transformation d'une image en deux dimensions vers une autre. La figure (1.1) montre le principe de réduction.

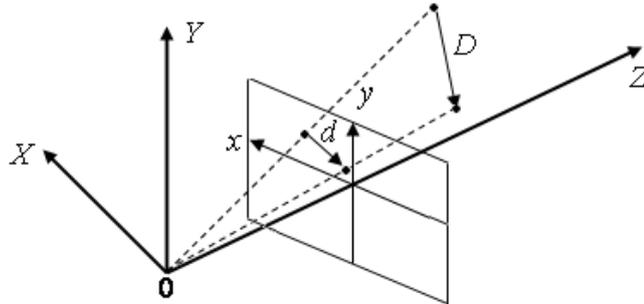


Figure 1.1. Projection du mouvement 3D en 2D

La projection d'images successives d'une scène sur un plan 2D complète et enrichit l'information sur la scène 3D projetée (scène réelle), en reproduisant les mouvements de cette dernière. Ces mouvements sont causés par le déplacement d'un objet dans la scène ou le déplacement de la caméra. Deux types de mouvements sont distingués [5] ; le mouvement réel et le mouvement apparent.

1.3.1. Mouvement réel

Le mouvement réel anime la scène réelle, dans l'espace réel 3D. Ce mouvement réel est observé soit par l'oeil humain, soit par un système de prise de vue.

1.3.2. Mouvement apparent

Le mouvement apparent est un mouvement 2D perçu dans une séquence à travers des variations spatio-temporelles de la luminance. Le champ des vitesses apparentes porte le nom de flot optique. Le mouvement apparent, appelé flot optique, est une variable cachée et n'est accessible que par l'analyse des variations temporelles de la séquence d'images. Par définition, le flot optique est le champ de vitesse (ou de déplacement) produit dans le plan image par des objets en mouvement dans un espace 3D. Il est représenté par des vecteurs comme montré dans la figure (1.2). Chaque Vecteur représente le déplacement d'un point de l'image. Le terme de flot optique a été inventé par le psychologue James Jérôme Gibson dans une étude sur la vision humaine [5].

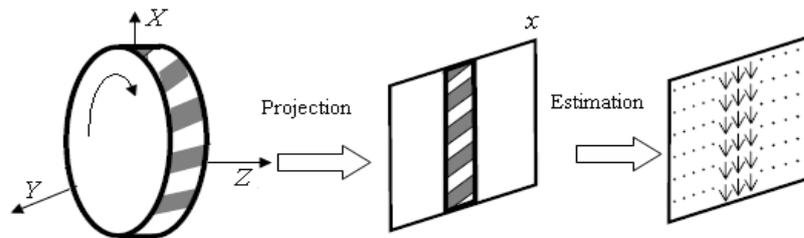


Figure 1.2. Exemple de flot optique

1.3.3. Différence entre mouvement réel et mouvement apparent

Lors de la projection du mouvement réel d'une scène tridimensionnelle (3D) sur un plan bidimensionnel (2D), on perd une dimension. Des informations relatives à cette troisième dimension sont perdues. Cette perte d'informations témoigne que le mouvement apparent et le mouvement réel sont différents. Par exemple dans la figure (1.3), les vitesses apparentes des points situés sur une sphère uniforme en rotation sont nulles, donc différentes des projections des vraies vitesses de ces points [2,5].

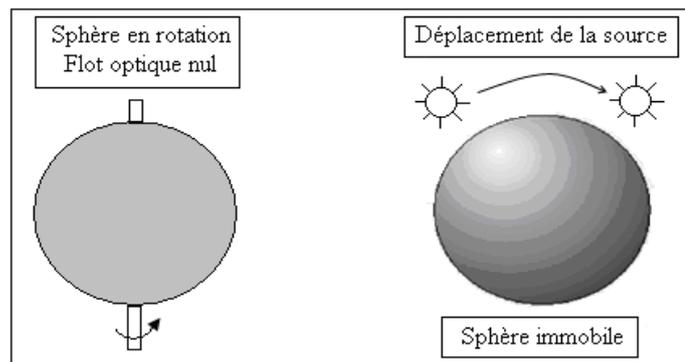


Figure 1.3. Sphère uniforme immobile et en rotation

1.4. Estimation de mouvement

L'estimation du mouvement consiste à mesurer la projection 2D dans le plan de l'image d'un mouvement réel 3D, dû à la fois au mouvement des objets dans la scène et aux déplacements de la caméra.

Le calcul du flot optique est une étape de traitement de l'image dite de bas niveau. Il consiste à extraire un champ de vitesses dense à partir d'une séquence d'images en faisant l'hypothèse que l'intensité est conservée au cours du déplacement.

1.4.1. Principales difficultés

L'estimation du mouvement dans une séquence d'images dynamiques présente plusieurs difficultés, on cite entre autres [1,4,20] :

1.4.1.1. Présence du bruit : Au cours du processus de formation d'une image, plusieurs sources contribuent à la présence du bruit. Citons à titre d'exemple le bruit du capteur, et le bruit de quantification intrinsèquement lié à l'utilisation d'une carte de numérisation d'images.

1.4.1.2. Ambiguïté due à la projection

La projection de l'espace tridimensionnel sur le plan de l'image provoque une perte d'informations pertinentes. Certains objets disparaissent plus ou moins longtemps derrière d'autres objets, ce qui pose des problèmes d'estimation à la frontière entre les deux objets [5]. De même, plusieurs points de l'espace peuvent être projetés sur un même point de l'image. Suivant l'orientation des objets par rapport à la caméra, une droite peut devenir un point et un plan peut devenir une droite, ce qui pose entre autres des problèmes de mise en correspondance.

1.4.1.3. Problème des régions homogènes

Il n'est possible de détecter le mouvement que dans la mesure où il y a des changements entre les images consécutives. Dans le cas d'un objet en mouvement dont le déplacement entre deux images consécutives est inférieur à la taille de cet objet, une zone de chevauchement apparaît où l'objet « glisse » sur lui-même. Lorsque l'intensité lumineuse de l'objet dans cette zone est homogène, cette zone n'apporte aucune information sur le mouvement.

1.4.1.4. Frontières d'occlusion

Le phénomène d'occlusion dans une séquence vidéo est dû à l'apparition et à la disparition ou aux croisements d'objets dans la vidéo. C'est le résultat du mouvement de la caméra et des objets dans la scène. Cependant, la frontière d'occlusion extérieure d'un objet est en réalité une zone de transition entre cet objet et d'autres objets, plus éloignés de la caméra et partiellement cachés. Ces objets peuvent effectuer des mouvements différents et il est difficile de savoir à quel objet correspond le mouvement estimé [2]. En effet, les techniques d'estimation de mouvement entre deux images prises à des instants t et $t+1$ d'une séquence vidéo sont basées sur les ressemblances existantes entre ces deux images. Si des zones apparaissent entre les deux images, les ressemblances diminuent et l'estimation échoue dans ces zones. La figure (1.4) donne un exemple illustratif du principe d'occlusion.

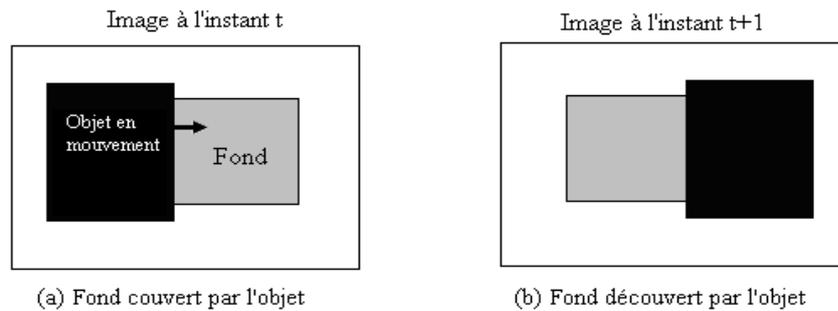


Figure 1.4. Principe d'occlusion

1.4.1.5. Problème d'ouverture

Le problème d'ouverture [5] apparaît lorsque l'information apportée par les variations de l'intensité lumineuse contenues dans le voisinage ne permet pas de le retrouver dans l'image suivante. Le schéma de la figure 1.5 illustre cette propriété dans le cadre d'un contour en mouvement. Le contenu informationnel d'un voisinage centré sur le pixel A ne permet pas de définir le déplacement de la portion de contour de manière exacte. Par contre, la structure des voisinages centrés sur les points B ou C permet de déterminer de manière unique leurs déplacements respectifs.

Dans le cas d'un voisinage centré en A, le seul vecteur déplacement pouvant être déterminé avec une certitude totale est celui qui correspond à un mouvement dans la direction normale au contour présent dans le voisinage.

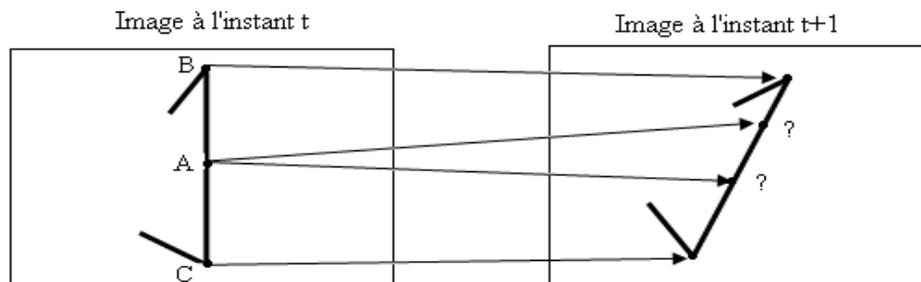


Figure 1.5. Principe d'ouverture

1.4.2. Contrainte de l'estimation de mouvement

Le changement de la luminance produit des déformations dans les objets sans que cela corresponde à un mouvement. Pour que le mouvement des objets soit clairement détectable, les suppositions suivantes devront être vérifiées [4,6]:

- Les objets sont des corps rigides c'est-à-dire que la déformation des objets est négligée au moins pour un minimum d'images voisines pour garantir le même mouvement pour tous les pixels du même objet

- Le mouvement des objets est translationnel pour un minimum d'images voisines.
- L'intensité est spatialement et temporellement uniforme c'est-à-dire que l'intensité des objets est constante lors du mouvement.

1.4.3. Équation du flot optique

Une séquence d'images peut être représentée par sa fonction de l'intensité $I(x, y, t)$. L'hypothèse de conservation de l'intensité stipule que l'intensité d'un point physique de la séquence d'images ne varie pas au cours du temps [6], c'est à dire :

$$I(p, t) = I(p + V(p)dt, t + dt) \quad (1.1)$$

avec $p = (x, y)^T$ et $V(p) = (u, v)^T$ le vecteur vitesse associé au point p à l'instant t . Les composantes u et v représentent respectivement les vitesses selon les directions x et y . Comme cette équation n'a pas une solution unique, nous devons faire l'hypothèse que l'intensité varie lentement avec x , y et t . Si cette hypothèse est respectée, nous pouvons développer le premier membre de l'équation (1.1) en séries de Taylor à l'ordre 1 autour de (x, y, t) .

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + e \quad (1.2)$$

La variable e contient les termes d'ordres plus élevés. Puisque $I(x + dx, y + dy, t + dt) = I + dI$, en divisant par dt , et en négligeant les termes d'ordre supérieurs, l'équation (1.2) peut être réduite à :

$$\frac{dI}{dt} = \frac{dx}{dt} \frac{\partial I}{\partial x} + \frac{dy}{dt} \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \quad (1.3)$$

par l'hypothèse que l'intensité lumineuse ne varie pas avec le temps, qui conduit à $\frac{dI}{dt} = 0$.

L'équation du flot optique se résume alors à la formulation suivante :

$$\frac{dx}{dt} \frac{\partial I}{\partial x} + \frac{dy}{dt} \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \quad (1.4)$$

Si on note :

$u = \frac{dx}{dt}$ la vitesse apparente suivant la direction x , $v = \frac{dy}{dt}$ la vitesse apparente suivant la

direction y , $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) = (I_x, I_y)$ gradient de l'image au pixel (x, y) à l'instant t et $I_t = \frac{\partial I}{\partial t}$

la dérivée de l'intensité d'un pixel par rapport au du temps. L'équation (1.4) devient:

$$I_x u + I_y v + I_t = 0 \quad (1.5)$$

Cette équation est appelée « équation de contrainte du flot optique ou équation de contrainte du mouvement apparent » couramment notée ECMA. Elle peut être reprise sous forme vectorielle comme :

$$\vec{\nabla} I \cdot \vec{V} = -I_t \quad (1.6)$$

Afin de trouver une solution unique pour u et v , nous avons besoin de deux équations indépendantes, ce qui n'est pas le cas, car l'équation (1.5) ne permet pas de déterminer de manière unique le flot optique. En effet, on se trouve en présence d'un système comportant une équation à deux inconnues, c'est un système sous-déterminé; le problème est dit mal posé. Mais cette dernière équation permet d'estimer la composante normale de la vitesse [4,6].

$$V_N = -\frac{I_t}{\|\nabla I\|} \quad (1.7)$$

Avec V_N la projection du vecteur vitesse sur le gradient spatial d'intensité ∇I .

La relation (1.7) fait apparaître un problème intrinsèque à l'estimation du mouvement : l'hypothèse de conservation de la luminance appliquée en un point de l'image ne permet de retrouver que la composante de vitesse parallèle au gradient spatial d'intensité. Cette indétermination est connue sous le nom du problème d'ouverture. De plus, l'estimation est impossible dans le cas où $\nabla I = 0$.

1.5. Panorama des méthodes d'estimation du mouvement

Les méthodes d'estimation du mouvement peuvent être regroupées en trois catégories [4,20] :

- les méthodes de mise en correspondance;
- les méthodes différentielles;
- les méthodes fréquentielles.

1.5.1. Méthodes différentielles

Les méthodes différentielles ou variationnelles consistent à résoudre un problème d'optimisation (local ou global) en minimisant une fonctionnelle, généralement basée sur l'équation du flot optique à laquelle on ajoute une contrainte pour particulariser les solutions. Selon le type de contrainte utilisée, on obtient différentes approches variationnelles:

1.5.1.1. Méthodes variationnelles globales

Ce type d'approches consiste à minimiser sur le domaine entier de l'image une fonctionnelle prenant en compte l'équation du flot optique ainsi qu'un terme de lissage, c'est à dire en ajoutant une contrainte de régularisation portant sur le gradient, le Laplacien (ou ordre supérieur) du champ de vitesse. La plus connue de ces méthodes est certainement celle proposée par Horn & Schunck [3] qui impose une contrainte globale de lissage sur le champ des vitesses.

1.5.1.2. Méthodes variationnelles locales

Les méthodes locales consistent à prendre en compte des hypothèses supplémentaires sur un domaine de taille réduite pour particulariser le flot optique. On minimise alors un critère sur un petit domaine, et on obtient ainsi le flot optique de ce petit domaine. La méthode locale la plus célèbre est celle de Lucas & Kanade [7].

1.5.1.3. Méthodes hybrides

La littérature récente présente des méthodes dites 'locales globales' qui ajoutent un filtrage temporel aux méthodes précédentes, donc le lissage et le filtrage temporel peuvent être considérés comme étant effectués sur toute l'image, mais la résolution de l'algorithme reste purement locale [21-22].

1.5.2. Méthodes fréquentielles

L'estimation du mouvement est également possible en considérant le domaine fréquentiel. Dans ce cas, on recense deux types d'approches [23-27] :

Les méthodes exploitant le fait qu'un décalage spatial entraîne un déphasage dans le domaine fréquentiel [23-24].

Les méthodes considérant la distribution de l'énergie dans l'espace des fréquences lorsque le mouvement est vu comme un ensemble d'orientations spatio-temporelles [5,26,27].

Les techniques basées sur l'étude de la phase sont utilisées pour l'estimation de la disparité dans les problèmes de la stéréovision. L'idée est fondée sur l'équivalence translation/déphasage de la transformée de Fourier (TF). On considère deux images $I_1 = I(p, t)$ et $I_2 = I(p, t + 1)$ de $N \times M$ pixels, ou I_2 est une version décalée de I_1 . Le

problème consiste à déterminer automatiquement le décalage $V = (u, v)$ entre ces deux images. Mathématiquement, ceci est exprimé par l'équation suivante :

$$I_2(x, y) = I_1(x + u, y + v) + \text{Bruit}(x, y) \quad (1.8)$$

Le bruit provient des capteurs (I_1 et I_2) qui peuvent correspondre à la même scène, mais prises par deux caméras différentes.

Le problème consiste à déterminer u et v . Nous présentons deux méthodes permettant de déterminer le décalage : la corrélation et la corrélation de phase. Il est, par ailleurs, utile de passer dans le domaine fréquentiel. En négligeant le bruit, on peut écrire la relation suivante entre les transformées de Fourier G_1 et G_2 de I_1 et I_2 , respectivement :

$$G_2(f) = G_1(f)e^{j2\pi fV} \quad (1.9)$$

avec $G_i = TFD\{I_i\} = |G_i|e^{j\theta_i}$, $i = \{1, 2\}$

1.5.2.1. Méthode de corrélation

La fonction d'intercorrélation entre I_1 et I_2 peut être calculée rapidement en considérant la transformée de Fourier discrète (TFD⁻¹ désigne la transformée de Fourier discrète inverse) :

$$c_{I_1 I_2}(x, y) = TFD^{-1}\{G_1(f) G_2^*(f)\} \quad (1.10)$$

En effet, on a :

$$\begin{aligned} c_{I_1 I_2}(x, y) &= TFD^{-1}\{G_1(f) G_2^*(f)\} \\ &= (I_1 * \tilde{I}_2)(x, y) = \sum_{a,b} I_1(a, b) \tilde{I}_2(x - a, y - b) = \sum_{a,b} I_1(a, b) I_2(a - x, b - y) \end{aligned} \quad (1.11)$$

avec $\tilde{I}_2(a, b) = I_2(-a, -b)$.

D'autre part, en négligeant le bruit, on peut écrire :

$$C_{I_1 I_2}(f) = G_1(f) G_2^*(f) \approx |G_1(f)|^2 e^{-j2\pi(fV)} \quad (1.12)$$

En utilisant les propriétés de la transformée de fourrier, on en déduit :

$$c_{I_1 I_2}(x, y) \approx c_{I_1 I_1}(x - u, y - v) \quad (1.13)$$

C'est donc la fonction d'autocorrélation de I_1 , centrée en (u, v) . En général, la fonction d'autocorrélation d'une image possède la forme d'un pic, et on peut donc déterminer $V = (u, v)$ en cherchant la position du maximum de $c_{I_1 I_2}(x, y)$.

L'algorithme de la méthode de corrélation est le suivant :

1. Calculer les TF de $I_1 = I(p, t)$ et $I_2 = I(p, t + 1)$, soit G_1 et G_2 ;

2. Calculer $C_{I_1 I_2}(f)$ par la relation : $C_{I_1 I_2}(f) = G_1(f) G_2^*(f)$;
3. Calculer la fonction d'inter-corrélation $c_{I_1 I_2}$ la TF inverse de $C_{I_1 I_2}(f)$;
4. Rechercher V , le maximum de $C_{I_1 I_2}$.

1.5.2.2. Méthode de corrélation de phase

Cette technique estime la translation 2D entre chaque paire d'images en prenant, pour chacune d'elles, sa transformée de Fourier. Son principe repose sur la recherche de la différence de phases pour chaque fréquence et le calcul de nouveau de la transformée de Fourier inverse [23,24].

Si $I_1 = I(p, t)$ et $I_2 = I(p, t + 1)$ sont les deux images de la scène se recouvrant partiellement et G_1, G_2 leurs transformées de Fourier respectives, alors la différence de phase entre ces deux images dans le domaine de Fourier est égale à leur spectre de puissance croisé normalisé (SPCN) exprimé par :

$$e^{j(\theta_1 - \theta_2)} = \frac{|G_1| \cdot |G_2| e^{j(\theta_1 - \theta_2)}}{|G_1| \cdot |G_2|} = \frac{|G_1| e^{j\theta_1} \cdot |G_2| e^{-j\theta_2}}{|G_1| \cdot |G_2|} = \frac{G_1 G_2^*}{\|G_1 G_2^*\|} \quad (1.14)$$

D'autre part, la transformation entre les deux images est une simple translation exprimée par le vecteur $V = (u, v)$, soit :

$$I_2(p) = I_1(p + V) = I_1(p) \delta(p - V) \quad (1.15)$$

δ est la fonction de Dirac .

La transformée de Fourier de cette équation est donnée par :

$$G_2(f) = G_1(f) e^{j2\pi f V} \quad (1.16)$$

ainsi, il est clair que la différence de phase $e^{j(\theta_1 - \theta_2)}$ entre les deux images n'est autre que $e^{j2\pi f V}$ pour chaque fréquence f .

à partir de l'équation 1.14, on trouve que la transformée de Fourier inverse de cette différence de phase (SPCN) produit la fonction de Dirac du vecteur de translation $\delta(p - V) = \delta(x - u, y - v)$:

$$d(p) = \delta(p - V) = F^{-1}(e^{j2\pi f V}) = F^{-1}(e^{j(\theta_1 - \theta_2)}) \quad (1.17)$$

Il est évident d'après l'équation 1.17, que pour calculer le déplacement 2D (translations suivant x et y), il suffit de chercher le pic de cette représentation spatiale (La fonction $d(p)$ présente en général un pic très net en $V = (u, v)$).

L'algorithme de la méthode de corrélation de phase est le suivant :

1. Calculer G_1 et G_2 les TF de I_1 et I_2 .
2. Calculer χ le SPCN de G_1 et G_2 .
3. Calculer d la TF inverse de χ .
4. Rechercher V , le maximum de d .

Cette méthode ne renvoie pas un résultat dense (seuls les maximums locaux sont considérés), et elle donne (dans sa version initiale) des déplacements entiers. On citera enfin l'utilisation du filtrage spatio-temporel [27-28] et de la transformée en ondelettes [29-33] pour l'estimation de mouvement à partir du domaine fréquentiel.

1.5.2.3. Estimation du mouvement par banc de filtres de Gabor

Une troisième approche fréquentielle consiste à appliquer l'équation du flot optique non pas directement sur la séquence $I(t)$, mais sur l'image convoluée par un banc de filtres G_i (avec $i = 1, \dots, N$, et N le nombre de filtres), les composantes de vitesse $V = (u, v)$ peuvent être extraites du produit de convolution ; on obtient alors [27] :

$$\left(\frac{\partial I}{\partial x} * G_i \right) \cdot u + \left(\frac{\partial I}{\partial y} * G_i \right) \cdot v + \frac{\partial I}{\partial t} * G_i = 0, \quad i = 1, \dots, N \quad (1.18)$$

où $*$ est l'opérateur de convolution 2D et G_i est la réponse impulsionnelle du filtre de Gabor, on obtient donc :

$$G_i(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} e^{j2\pi f_0(x \cos \theta_i + y \sin \theta_i)} \quad (1.19)$$

avec f_0 est la fréquence centrale, σ est la largeur de la fonction gaussienne et $\theta_i = \frac{i\pi}{N}$ est

l'orientation de chaque filtre. La convolution des dérivées spatio-temporelles de la séquence d'images par un banc de N filtres G_i donne pour chaque pixel $p(x, y)$ de l'image un système à N équations dont les deux inconnues sont les composantes du vecteur vitesse (u, v) .

Sur cette base, plusieurs méthodes ont été développées :

Heeger [4,20]: Il s'agit d'utiliser une pyramide gaussienne pour évaluer les différents ordres de grandeur de vitesse. Chaque niveau de la pyramide est filtré par une famille de 12 filtres de Gabor, puis on effectue une estimation au sens des moindres carrés sur l'énergie des filtres pour satisfaire l'équation fréquentielle du flot optique.

Weber et Malik [28] : ajoutent une étape supplémentaire, qui consiste à permuter l'opérateur de convolution avec l'opérateur de dérivation, permet de réduire le nombre de filtrages :

$$\frac{\partial(I * G_i)}{\partial x} u + \frac{\partial(I * G_i)}{\partial y} v + \frac{\partial(I * G_i)}{\partial t} = 0, \quad i = 1, \dots, N \quad (1.20)$$

Dans ce cas, l'opération de filtrage porte directement sur l'image $I(t)$, et non plus sur les dérivées spatio-temporelles. Ainsi, en chaque pixel de la séquence d'images, nous obtenons N équations de mouvement, ce qui permet de remonter au flot optique en résolvant le système d'équations (1.20) par la méthode des moindres carrés.

Le réglage des paramètres f_0 , σ et N est déterminé pour estimer de façon précise le mouvement. Ces trois paramètres influent en particulier sur le choix de calcul de filtres sur l'image (gradients), sur le conditionnement du système d'équations du flot optique, ainsi que sur le coût calculatoire de l'opération.

Globalement, ce type d'approches est moins sensible aux variations d'éclairage et est plus robuste dans le cas d'observations bruitées. Mais Le problème majeur de ces techniques est qu'elles sont coûteuses en terme de temps de calcul à effectuer afin d'obtenir des bons résultats car elles utilisent un grand nombre d'images consécutives (jusqu'à 7 images [6]) et, comme dans le cas des méthodes différentielles, elles ne sont applicables directement qu'à des mouvements de faible amplitude.

1.5.2.4. Approches par ondelettes

Les méthodes basées sur les ondelettes sont très populaires en traitement d'image. Dans le cadre du flot optique, elles sont équivalentes aux décompositions sur des familles de filtres présentées précédemment et induisent donc un filtrage excessif en plus d'une paramétrisation a priori (choix des ondelettes) du calcul.

Méthode Bernard [29]:

La méthode de Bernard consiste à projeter l'équation du flot optique sur une base d'ondelettes $(\Psi^n)_{n=1, \dots, N}$ translatée de $V = (u, v)$ dans l'espace. Ceci nous amène à un système de N équations à résoudre :

$$\iint \left(\frac{\partial I}{\partial t} + u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} \right) \Psi^n(X - V) \cdot dx dy = 0 \quad \text{pour } n = 1, \dots, N \quad (1.21)$$

avec $X = (x, y)$.

Méthode Szeliski [34]:

Szeliski a proposé de décomposer les vecteurs vitesse recherché v_x et v_y sur des fonctions particulières:

$$v_x(x_i, y_i) = \sum_j \hat{u}_j B_j(x_i, y_i) \quad (1.22)$$

$$v_y(x_i, y_i) = \sum_j \hat{v}_j B_j(x_i, y_i) \quad (1.23)$$

avec les B_j étant des fonctions non nulles sur un certain intervalle et nulles ailleurs, translatées spatialement les unes par rapport aux autres.

1.5.3. Méthode d'appariement de blocs

La méthode de l'appariement de blocs (block-matching), appelée aussi dans la littérature correspondance de blocs ou encore similarité de blocs, est la plus utilisée et adaptée par la plupart des standards de compression vidéo due à sa simplicité pour les implémentations Hardware.

Le principe général de l'appariement de blocs est d'exploiter les redondances temporelles existantes entre des images consécutives. Pour cela, considérons une séquence vidéo dans laquelle nous voulons estimer le mouvement des différents objets qui y sont contenus. Pour simplifier l'estimation, nous ne considérons que le mouvement présent entre deux images successives A et B , et chaque image est subdivisée en blocs de tailles égales et chaque bloc est considéré comme étant un objet indépendant [35-37]. Voici le procédé général de l'approche (voir figure 1.6 pour un exemple) :

1. Séparation de l'image A en blocs de $N \times M$.
2. Sélection d'un bloc de l'image A , déplacement à toutes les positions d'une région de recherche choisie de l'image B et évaluation de la fonction de coût.
3. Choix de la position de l'image B où la fonction de coût est minimale.
4. Le vecteur correspondant est associé au bloc.
5. Répéter jusqu'à ce que tous les blocs dans l'image A soient appariés.

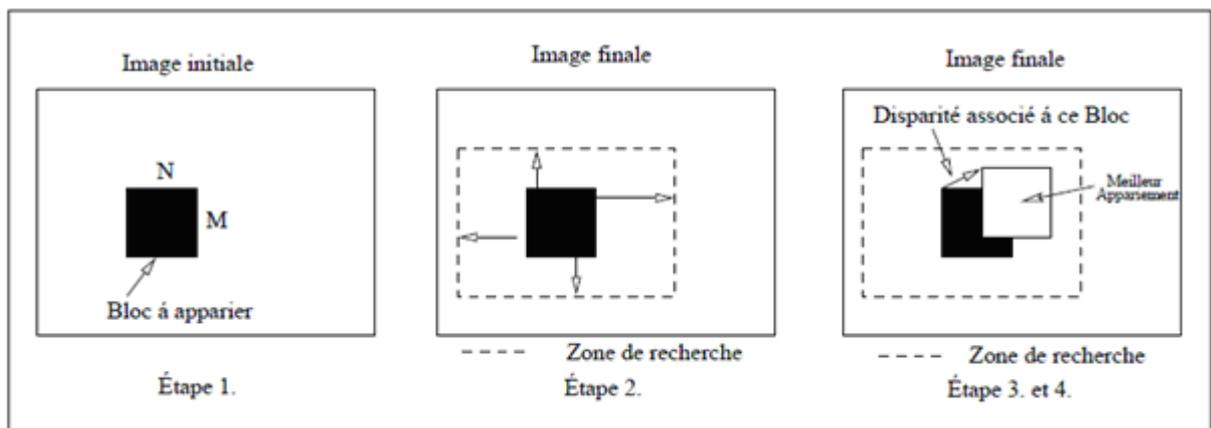


Figure 1.6. Exemple d'une procédure d'appariement de blocs.

Les méthodes de mise en correspondance de blocs peuvent être classées en fonction :

1. Du critère de mise en correspondance ;
2. De la dimension du bloc et celle de la fenêtre de recherche ;
3. De la stratégie de recherche.

1.5.3.1. Largeur de la zone de recherche

L'utilisation d'une fenêtre de recherche permet de limiter le nombre de blocs de référence que l'algorithme doit tester. La largeur de la zone de recherche affecte directement la vitesse et la précision de l'encodeur vidéo. Dans la plupart des standards, c'est un paramètre limité à partir des expériences ou imposé par les contraintes des implémentations hard [38]. Cependant, les séquences de mouvement rapide, où les objets changent leurs positions d'une image à l'autre d'une distance considérable, doivent subir une recherche dans une zone couvrant ce déplacement, engendrant bien sûr un temps de calcul plus long. Par contre, les séquences de mouvement lent doivent minimiser la zone de recherche pour profiter du temps de codage.

1.5.3.2. Taille des blocs

La taille du bloc choisie affecte clairement la recherche et la comparaison des blocs et ainsi la performance de l'estimation de mouvement. En choisissant une taille importante des blocs (32,64,...), on réduit la précision des vecteurs du mouvement obtenus du fait que les grands blocs contiennent des objets de mouvements différents en vitesse et en direction, mais d'un autre côté, on gagne en matière de la taille des informations émises concernant les vecteurs de mouvement puisque le nombre de blocs est réduit.

Le choix d'une taille réduite (2,4,...) permet d'explicitier le mouvement des objets et les limites des objets seront bien identifiées. Il permet aussi de produire des vecteurs de mouvement très précis. D'un autre côté, les petits blocs souffrent du temps de calcul important. Les tailles de 8x8 et de 16x16 des blocs sont considérées généralement adéquates pour les applications de compression vidéo.

1.5.3.3. Types de fonction de coût

C'est la fonction principale du Block-Matching, qui décide à quel point un bloc est similaire à un autre (elle est appelée aussi Mesure de Distorsion de Bloc (MDB)). Son choix est nécessaire pour obtenir un bloc optimal. Elle a aussi un impact direct sur la complexité de calcul et sur la précision de l'estimation de mouvement. Plusieurs fonctions sont rencontrées dans la littérature dont les plus utilisées sont:

Différence moyenne absolue

$$DMA = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I_c(x+i, y+j) - I_r(x+i+dx, y+j+dy)| \quad (1.24)$$

Erreur moyenne quadratique

$$EMQ = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (I_c(x+i, y+j) - I_r(x+i+dx, y+j+dy))^2 \quad (1.25)$$

avec I_c , I_r l'image cible et l'image de référence de taille $N \times N$, (x, y) les coordonnées du bloc cherché et (dx, dy) le vecteur de mouvement.

1.5.3.4. Types de recherches

Il existe de nombreuses méthodes de block-matching dans l'ensemble des articles [35-40] cherchant à optimiser l'efficacité et la rapidité de l'algorithme. Nous nous citons diverses méthodes largement utilisées, de la plus simple et la plus ancienne, à savoir la recherche exhaustive, à des méthodes très récentes et optimales : la recherche dite en "3 pas", la recherche avec éliminations successives, la recherche sur une grille en diamant, la recherche sur une grille hexagonale. Ces méthodes permettent une forte réduction de la complexité de l'estimation de mouvement. Les algorithmes rapides testent des sous-ensembles de la fenêtre de recherche. Ces sous-ensembles peuvent être de type "carré", "fois" et "plus" (figure 1.7).

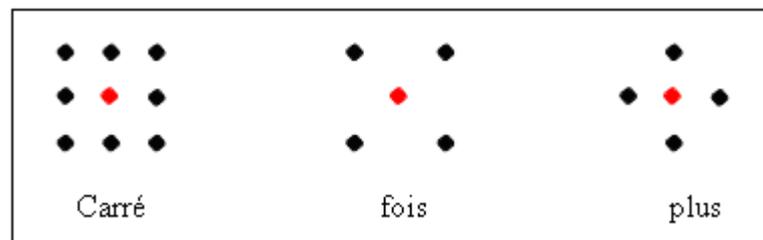


Figure 1.7. Sous-ensembles de recherche

1.6. Discussion sur les différentes méthodes

Les méthodes différentielles, basées sur la résolution de l'équation du flot optique, présentent de nombreux avantages face aux méthodes de mise en correspondance et face aux méthodes fréquentielles. L'équation du flot optique permet une estimation sous-pixellique directe du mouvement, contrairement aux méthodes de mise en correspondance. La mesure du mouvement ne nécessite qu'un calcul local des dérivées spatio-temporelles de la séquence. Ces opérations ont un coût de calcul faible comparé aux filtrages spatiotemporels imposés par les méthodes fréquentielles. Ces deux avantages, associés au fait que l'équation du flot

optique est linéaire par rapport au vecteur vitesse, expliquent le succès et le nombre très important de travaux portant sur les méthodes différentielles. Leur principal inconvénient est leur fondement basé sur des hypothèses d'intensité de lumière constante, et de faibles déplacements (de l'ordre du pixel). Lors de déplacements plus importants, il est néanmoins possible de surmonter le problème par des approches multi-résolutions [10], en supposant que l'hypothèse d'intensité constante reste valable malgré la diminution de la résolution.

Les méthodes de block-matching ne sont pas dépendantes de l'équation du flot optique. Les hypothèses d'intensité constante et de faibles déplacements n'entrent pas en considération. Elles sont plus robustes lors de variations d'intensité et traitent des déplacements d'amplitudes quelconques. Les points faibles de ce type de méthodes sont d'une part la complexité des algorithmes et d'autre part, il est nécessaire que la surface de corrélation présente un maximum bien défini. En effet, ces méthodes fonctionnent moins bien sur des zones de l'image homogène.

1.7. Conclusion

Nous avons rappelé dans ce chapitre quelques notions sur la théorie de l'estimation du mouvement. La notion du flot optique a été développée plus profondément. Nous avons notamment présenté plusieurs techniques d'estimation du flot optique. On peut observer que certaines techniques opèrent dans le domaine spatio-temporel, d'autres dans le domaine fréquentiel. Dans le domaine spatio-temporel, nous avons rappelé l'équation de contraintes spatiotemporelles du flot optique qui relie les composantes du vecteur vitesse aux gradients spatiotemporels. Les approches qui découlent de cette équation s'appellent méthodes différentielles. Ces dernières méthodes seront développées d'avantage dans les chapitres suivants.

Chapitre 2

Analyse multi-échelles par ondelettes géométriques

2.1. Introduction

La théorie des ondelettes ait naissance dans le milieu des années 80 [11]. De nos jours, elle s'utilise dans de nombreux domaines comme l'analyse harmonique, la vision par ordinateur, le traitement du signal, la compression d'images et autres. Son succès est dû à son adaptabilité aux données et à sa facilité d'implémentation. La transformée en ondelettes (TO) est un outil très efficace dans l'analyse et le débruitage de signaux non stationnaires présentant des discontinuités ponctuelles. Dans les images, les discontinuités sont principalement générées par les contours et par conséquent elles sont distribuées le long d'une courbe lisse dans le plan spatial. Dans la dernière décade, une diversité de nouveaux outils d'analyse multi-échelle basés sur une analyse en ondelettes dans un espace géométrique a été proposée [38]. Ces nouvelles techniques visent à améliorer le pouvoir d'approximation des ondelettes en introduisant une notion de *directionnalité* ce qui leur permet de mieux tenir compte de la géométrie présente dans l'image.

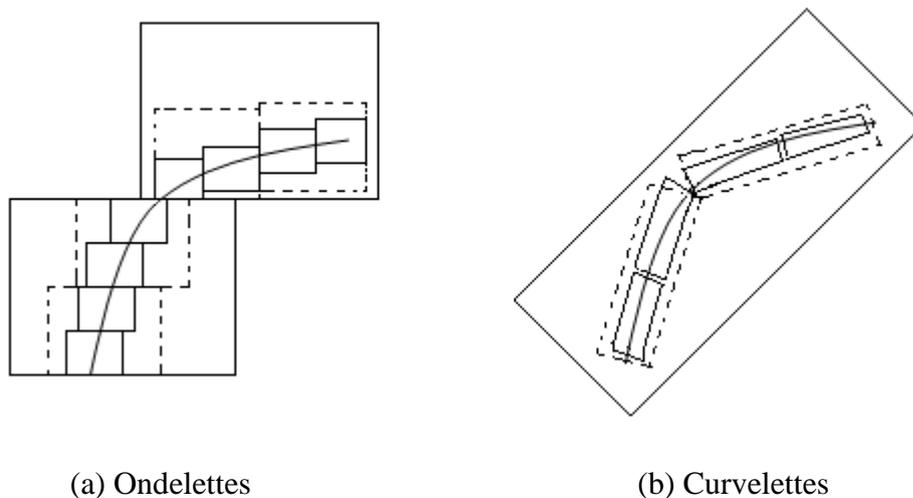


Figure 2.1 Fonctions d'approximation des ondelettes et des curvelettes.

La figure (2.1) montre une comparaison de l'approximation non-linéaire des ondelettes et des curvelettes. On remarque que les ondelettes nécessitent beaucoup de coefficients pour représenter précisément un tel contour. Cependant les curvelettes peuvent représenter un contour lisse avec moins de coefficients pour la même précision.

Dans ce chapitre on propose un rappel sur les définitions et les propriétés des ondelettes ainsi que les nouveaux outils d'analyse multi-résolution qui sont les ridgelettes [12], les curvelettes [13,42-43] et les contourlettes [44-47].

2.2. Transformée en ondelettes

Dès la naissance des ondelettes, elles ont gagné un intérêt considérable pour le traitement des signaux. L'idée est de représenter un signal à plusieurs résolutions, ce qui permet de représenter ses tendances principales dans seulement quelques coefficients. En traitement d'images, les ondelettes ont été employées pour différentes applications telles que le débuitage, la compression et l'estimation de mouvement [29-33]. Cette section présente les ondelettes dans ses deux versions ; les ondelettes continues unidimensionnelles et les ondelettes discrètes sous les différentes constructions possibles. Puis une extension aux signaux bidimensionnels sera également présentée.

2.2.1. Définition

Les fonctions d'ondelettes ont été introduites par Grossman et Morlet pour modéliser des signaux sismiques. Les ondelettes sont des fonctions générées par translation et dilatation d'une fonction ψ appelée ' ondelette mère '. L'ensemble de ces fonctions d'ondelettes forme une famille permettant d'analyser le signal dans un espace transformé. Une base d'ondelettes est définie par [11,48]:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right) \quad (2.1)$$

où $a \in \mathfrak{R}^*$, $b \in \mathfrak{R}$ désignent le paramètre de dilatation et le paramètre de translation respectivement. Le facteur de normalisation $a^{1/2}$ garantit que toutes les ondelettes ont la même énergie. La figure (2.2) montre une ondelette mère translatée et dilatée. On voit que la résolution en fréquence est meilleure lorsque le facteur d'échelle a est grand, mais la résolution temporelle est alors mauvaise.

2.2.2 Transformée en ondelettes continues

La transformée en ondelettes continues (TOC) d'un signal $x(t)$ est définie par :

$$TOC_{a,b}(x) = \int_{-\infty}^{+\infty} x(t)\psi_{a,b}^*(t)dt = \langle x(t), \psi_{a,b}(t) \rangle \quad (2.2)$$

où $*$ désigne le complexe conjugué, $\langle \cdot, \cdot \rangle$ est le produit scalaire. Les fonctions $\psi_{a,b}^*$ sont obtenues par le décalage temporel d'une version compressée/dilatée de l'ondelette mère $\psi(t)$. Elles sont définies par l'équation (2.1).

Le signal peut être reconstruit à partir de ses coefficients d'ondelettes utilisant la formule :

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} TOC_{a,b}(x) \psi_{a,b}(t) \frac{da}{a^2} \right] db \quad (2.3)$$

où C_ψ est un nombre constant qui ne dépend que de l'ondelette choisie $\psi(t)$.

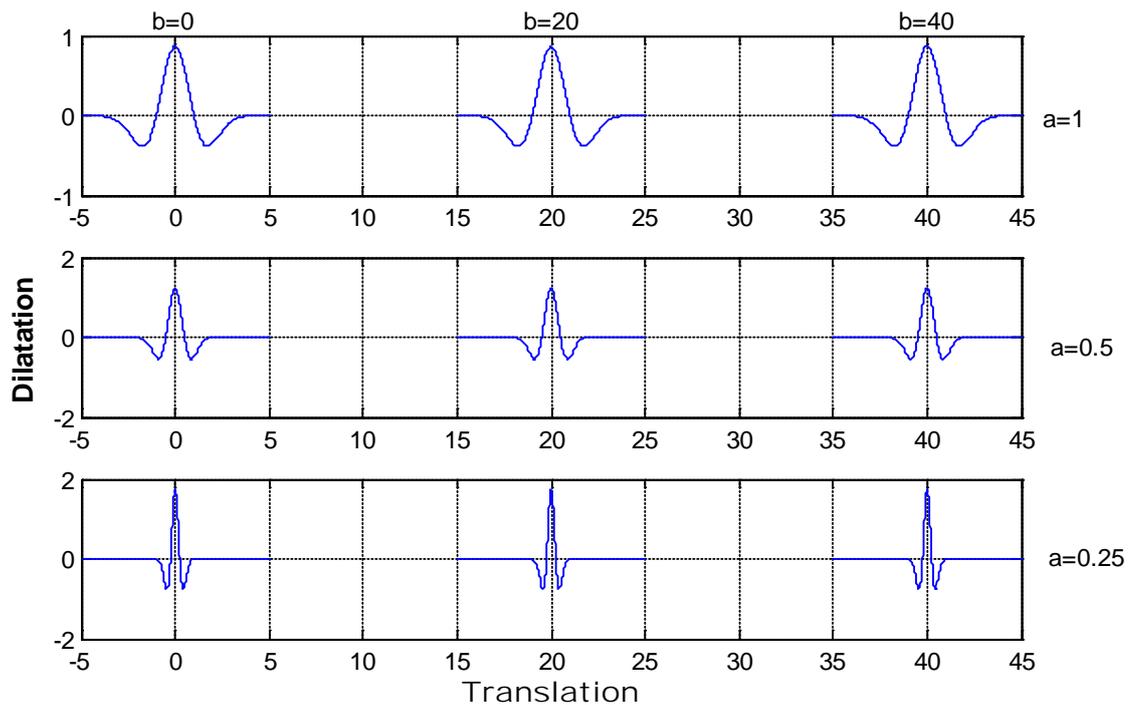


Figure 2.2. Translation et dilatation des ondelettes.

2.2.3. Transformée en ondelettes discrètes

La transformée en ondelettes discrètes (TOD) est issue de la version continue, à la différence de cette dernière, la TOD utilise un facteur d'échelle et une translation discrétisés. Donc les bases d'ondelettes sont alors définies par les fonctions :

$$\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k); (j, k) \in \mathbb{Z}^2 \quad (2.4)$$

Cette transformée est appelée transformée dyadique, elle est représentée dans la figure (2.3).

La transformée en ondelettes discrètes du signal $x(t)$ est définies par :

$$TOD_{j,k}(x) = \int_{-\infty}^{+\infty} x(t) \psi_{j,k}^*(t) dt \quad (2.5)$$

En choisissant adéquatement ψ , la famille $\psi_{j,k}$ constitue une base orthonormée, on pourra dès lors récupérer le signal original par la transformée inverse:

$$x(t) = \sum_j \sum_k TOD_{j,k}(x) \psi_{j,k}(t) \quad (2.6)$$

Cette équation permet d'assurer une reconstruction exacte.

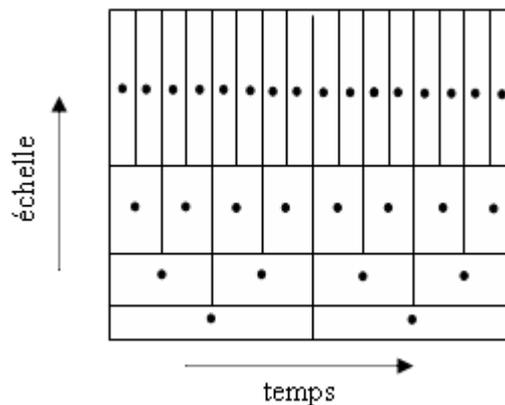


Figure.2.3. Grille dyadique.

2.2.4. Analyse multi-résolutions et transformée en ondelettes

L'analyse multi-résolutions permet d'analyser les signaux à différents niveaux de résolution. Elle consiste à projeter le signal $x(t)$ sur une série de sous-espaces orthogonaux ; les espaces d'approximations V_j et les espaces de détails W_j .

Afin de combiner la transformée en ondelettes et l'analyse multi-résolutions, une nouvelle fonction est introduite : 'la fonction d'échelle' ou 'fonction père', que l'on peut considérer comme la fonction complémentaire de l'ondelette. Elle permet d'extraire les composantes basses fréquences du signal, alors que l'ondelette extraira les composantes hautes fréquences. L'introduction de la fonction d'échelle et de la famille de fonctions qui en découle, nous permet d'obtenir une approximation du signal; les détails sont ensuite codés par les ondelettes. Cette méthode permet de décomposer en ondelettes n'importe quelle fonction indépendamment de son comportement à l'infini.

La fonction d'échelle ϕ doit vérifier :

$$\int_{-\infty}^{+\infty} \phi(x) dx = 1 \quad (2.7)$$

Et les fonctions qui en découlent doivent vérifier :

$$\phi_{j,k}(t) = 2^{-j/2} \cdot \phi(2^{-j}t - k) \quad (2.8)$$

Pour un signal $x(t)$ à énergie finie, les coefficients de détails sont définis par :

$$d_x(j,k) = \langle x, \psi_{j,k} \rangle \quad (2.9)$$

et les coefficients d'approximations sont définis par :

$$a_x(j,k) = \langle x, \phi_{j,k} \rangle \quad (2.10)$$

L'approximation du signal $x(t)$ à la résolution 2^{-j} correspond à sa projection dans V_j :

$$A_j x(t) = \sum_{k \in \mathbb{Z}} a_x(j,k) \phi_{j,k}(t) \quad (2.11)$$

Le détail du signal $x(t)$ à la résolution 2^{-j} correspond à sa projection dans W_j :

$$D_j x(t) = \sum_{k \in \mathbb{Z}} d_x(j,k) \psi_{j,k}(t) \quad (2.12)$$

L'approximation du signal à un niveau j correspond donc à une approximation plus grossière complétée par le détail :

$$A_j x(t) = A_{j+1} x(t) + D_{j+1} x(t) \quad (2.13)$$

L'analyse multi-résolutions permet de représenter un signal $x(t)$ sous la forme d'une somme de détails et d'une approximation :

$$x(t) = A_j x(t) + \sum_{j=1}^J D_j x(t) = \sum_{k \in \mathbb{Z}} a_x(j,k) \phi_{j,k}(t) + \sum_{j=1}^J \sum_{k \in \mathbb{Z}} d_x(j,k) \psi_{j,k}(t) \quad (2.14)$$

avec J nombre de niveaux.

La fonction d'ondelettes ψ permettant de relever les hautes fréquences est représentée par un filtre numérique passe-haut pour définir les détails, et la fonction d'échelle ϕ , qui permet de relever les basses fréquences est représentée par un filtre passe-bas pour définir les parties lisses ou douces du signal (approximations).

2.2.5. Algorithme rapide de Mallat

La transformée en ondelettes discrète est implantée à l'aide de l'algorithme rapide de Mallat [48] en utilisant l'analyse multi-résolutions. Cet algorithme est basé sur la définition

d'une paire de filtres (h, g) associés aux fonctions d'échelle et d'ondelette par produit scalaire comme suit [48]:

$$\begin{aligned} h(n) &= \langle \phi, \phi_{-1,n} \rangle \\ g(n) &= \langle \psi, \phi_{-1,n} \rangle \end{aligned} \tag{2.15}$$

avec:
$$\phi_{-1,n}(t) = 2^{-1/2} \phi(2t - n) \tag{2.16}$$

L'algorithme de Mallat consiste à calculer les coefficients de détails et d'approximations en utilisant des filtrages et décimations successifs en introduisant le signal à analyser dans deux filtres passe-bas (h) et passe-haut (g). A ce niveau, deux vecteurs seront obtenus : A_j et D_j . Les éléments du vecteur A_j sont appelés coefficients d'approximation, ils correspondent aux plus basses fréquences du signal, tandis que les éléments du vecteur D_j , appelés coefficients de détail correspondent aux plus hautes fréquences. La procédure peut être répétée avec les éléments du vecteur A_j et successivement avec chaque nouveau vecteur A_j obtenu. Les sorties des filtres sont sous-échantillonnées par un facteur 2. Le processus de décomposition peut être répété J fois, avec J le nombre maximal de niveaux.

On peut reconstruire le signal grâce à des filtres \tilde{h} et \tilde{g} selon l'algorithme présenté en figure (2.4). L'approximation A_j à un niveau donné j est la somme des coefficients de détails D_{j+1} et d'approximations A_{j+1} du niveau supérieur préalablement filtrés et ré-échantillonnés.

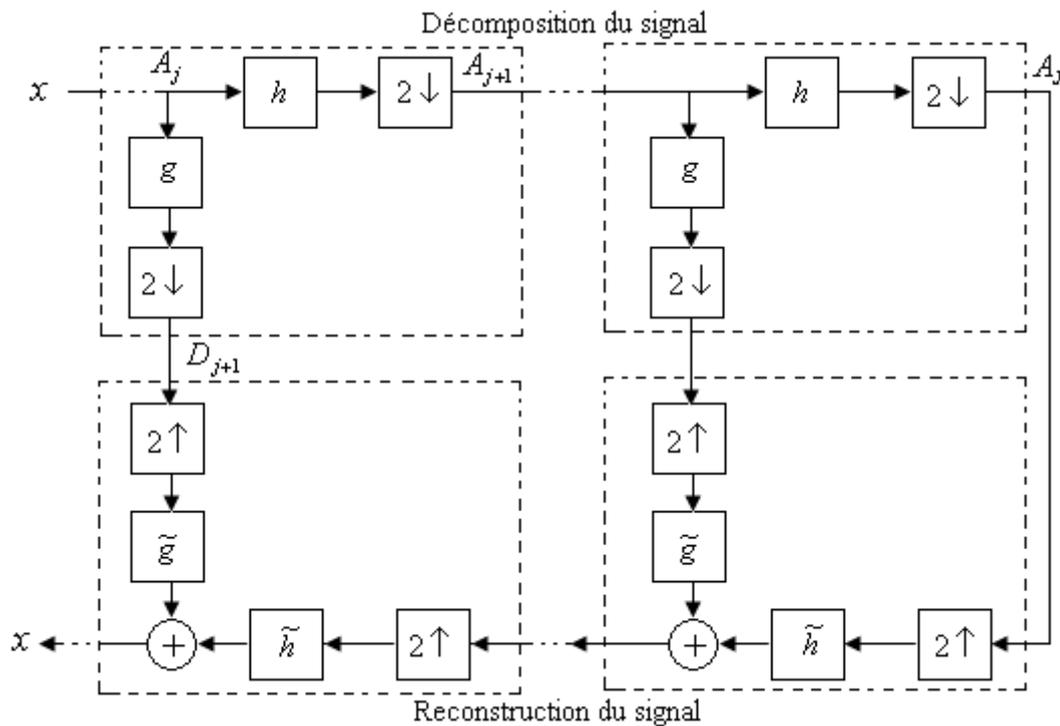


Figure.2.4. Algorithme de Mallat pour la décomposition et la reconstruction d'un signal.

Les filtres de reconstruction \tilde{h} et \tilde{g} sont les filtres conjugués en quadrature de h et g respectivement:

$$\begin{aligned}\tilde{h}(n) &= h(-n) \\ \tilde{g}(n) &= g(-n)\end{aligned}\tag{2.17}$$

2.2.6. Ondelettes non décimées

Les ondelettes orthogonales séparables sont connues par leur non-invariance par translation. On peut évidemment obtenir une transformée invariante par translation si les différentes échelles ne sont pas décimées, ceci est possible si on dispose de l'espace mémoire et le temps de calcul suffisants. Le problème est que contrairement aux ondelettes décimées, dont le nombre de coefficients est identique au nombre d'échantillons, on obtient de la redondance dans le cas non décimé. La transformée en ondelettes non décimée peut être réalisée de différentes manières, on peut effectuer plusieurs transformées décimées, après avoir appliqué un décalage au signal, ou on peut utiliser des filtres modifiés par insertion de zéros, c'est l'idée de l'algorithme à trous. Cet algorithme proposé par Holdschneider *et al.* [41], utilise un filtre différent à chaque échelle. En effet, on définit pour tout filtre $h(n)$, un filtre $h_j(n)$ obtenu en insérant entre chaque entrée de h , $2^j - 1$ zéros. On peut alors utiliser le schéma classique de la décomposition représenté en figure (2.4), dans lequel les filtres de chaque bloc j sont changés par h_j , et où les décimations ont été supprimées [41]. La reconstruction se fait en moyennant les reconstructions des différents filtres. L'algorithme à trous est montré dans la figure (2.5).

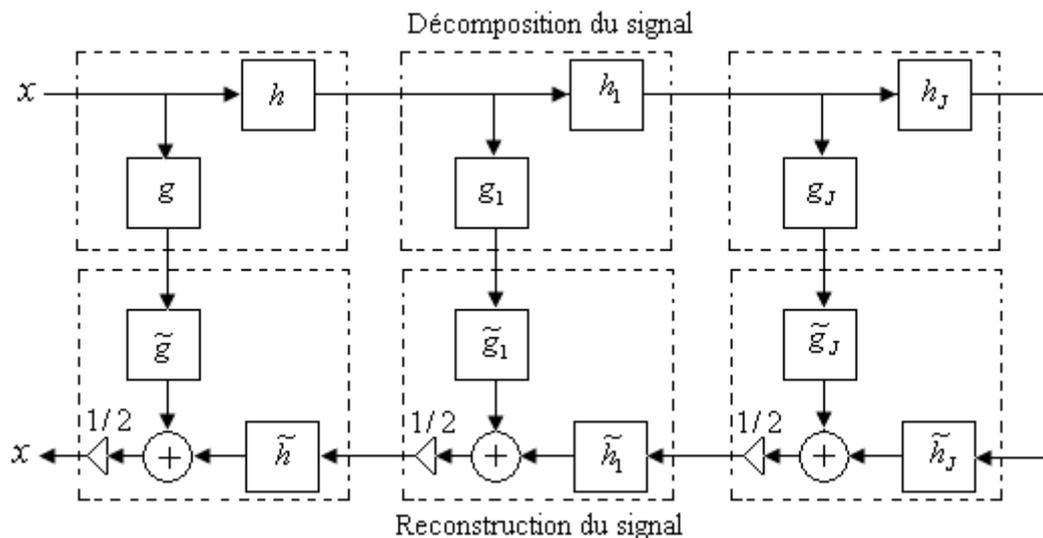


Figure.2.5. Schéma de la TO non décimée en utilisant l'algorithme à trous.

2.2.7. Paquets d'ondelettes

Les paquets d'ondelettes sont une généralisation de la décomposition en ondelettes qui offre une analyse plus riche. Elles ont été introduites par Coifman *et al.* [41] au début des années 90 pour pallier le manque de résolution fréquentielle de l'analyse en ondelettes. Dans les paquets d'ondelettes, le signal est décomposé en deux vecteurs, le vecteur des détails et celui des approximations. Ainsi, la décomposition est effectuée sur les deux vecteurs de détails et d'approximation en deux sous vecteurs de détails et d'approximation également et ainsi de suite. On obtient alors le schéma hiérarchique semblable à l'exemple de la figure (2.6) où la lettre a désigne les coefficients des approximations et d ceux des détails.

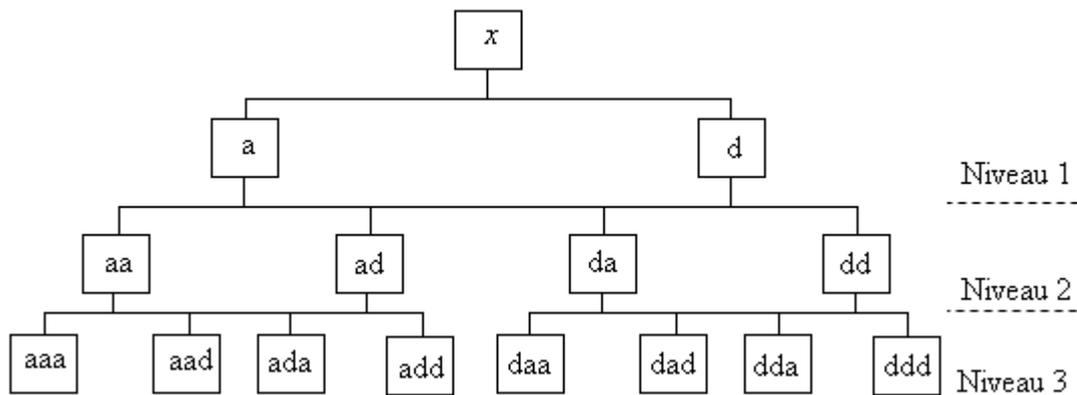


Figure 2.6. Arbre de décomposition par la transformée en paquets d'ondelettes.

2.2.8. Transformée en ondelettes bidimensionnelles

La théorie des ondelettes peut être généralisée sur plusieurs dimensions. Ceci est possible en appliquant de manière séquentielle et séparable sur les différentes dimensions du signal. Pour les signaux 2D, La reformulation mathématique donne naissance à une fonction d'échelle à deux dimensions et à trois ondelettes 2D :

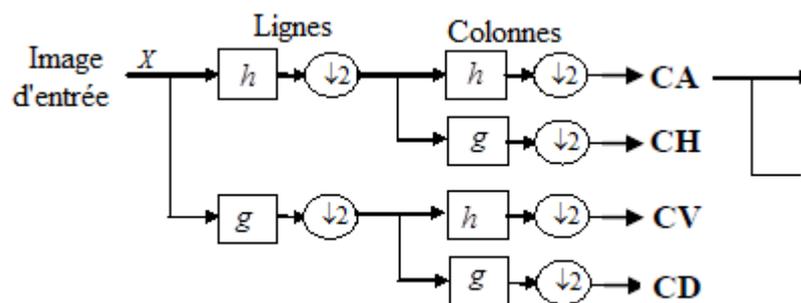
$$\begin{cases} \varphi^0(x, y) = \varphi(x)\varphi(y) \\ \psi^1(x, y) = \phi(x)\psi(y) \\ \psi^2(x, y) = \psi(x)\phi(y) \\ \psi^3(x, y) = \psi(x)\psi(y) \end{cases} \quad (2.18)$$

Cet ensemble de fonctions d'ondelettes fournit une représentation dyadique de l'image avec trois sous-bandes de coefficients d'ondelettes par résolution et une unique image d'approximation.

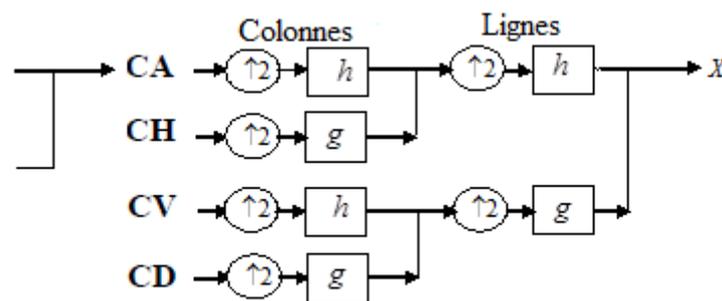
Pour calculer les coefficients d'approximations et de détails d'une image I , on utilise la généralisation de l'algorithme pyramidal présenté au paragraphe (2.2.5). Chaque étape de cet

l'algorithme est appliquée successivement aux lignes puis aux colonnes de l'image. Le principe de décomposition et de reconstruction des signaux 2D est montré dans la figure (2.7). On obtient, pour un niveau de décomposition j , une sous-image d'approximations A_j et trois sous-images de détails D_j^1, D_j^2, D_j^3 selon l'orientation fréquentielle horizontale verticale et diagonale. Les sous-bandes de détails D^1, D^2 et D^3 correspondent respectivement aux orientations horizontales (hautes fréquences en y , basses fréquences en x), verticales (le contraire), et diagonales (hautes fréquences en x et y).

Les quatre sous images (sub-images) résultent dénotant CA (les coefficients d'approximations), CH (les coefficients de détails horizontaux), CV (les coefficients de détails verticaux) et CD (les coefficients de détails diagonaux). Le signal approximation est à son tour décomposé et ainsi de suite. La figure (2.8) illustre la stratégie de l'algorithme de Mallat bidimensionnel pour la décomposition des signaux bidimensionnels tels que l'image. La reconstruction est l'opération similaire, mais inversement [11,48] ; à partir des coefficients d'approximations et de détails, on applique une interpolation suivie d'un filtrage selon les lignes puis selon les colonnes.



(a) Décomposition d'une image



(b) Reconstruction d'une image

Figure.2.7. Algorithme de Mallat pour la décomposition et la reconstitution d'une image.

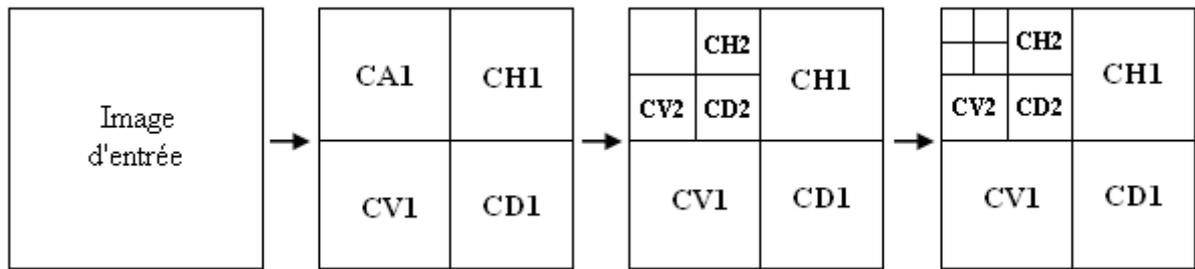


Figure 2.8. Décomposition multi niveaux par ondelettes

2.2.9. Exemple d'ondelettes

Il existe plusieurs types d'ondelettes dont les plus populaires sont l'ondelette de Morlet, le chapeau mexicain, Symmlet, l'ondelette de Daubechies pour différents ordres et l'ondelette de Haar. La figure (2.9) et la figure (2.10) présentent quelques ondelettes de Daubechies pour différents ordres et une décomposition multi-résolution sur différentes bases d'ondelettes d'une image respectivement [49].

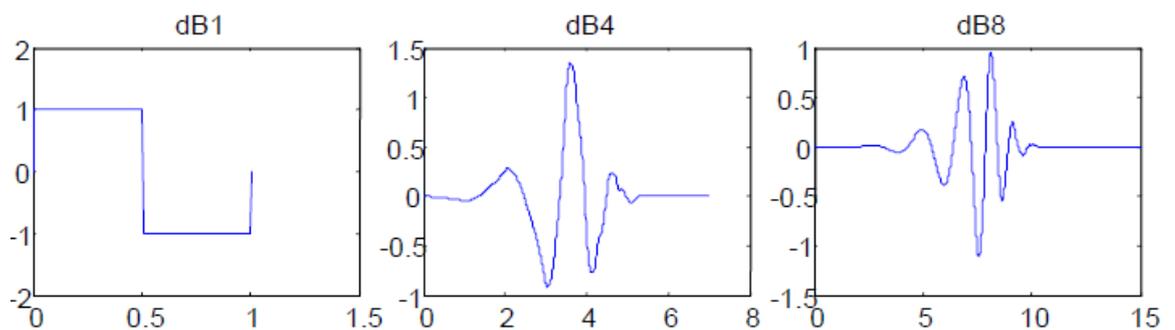
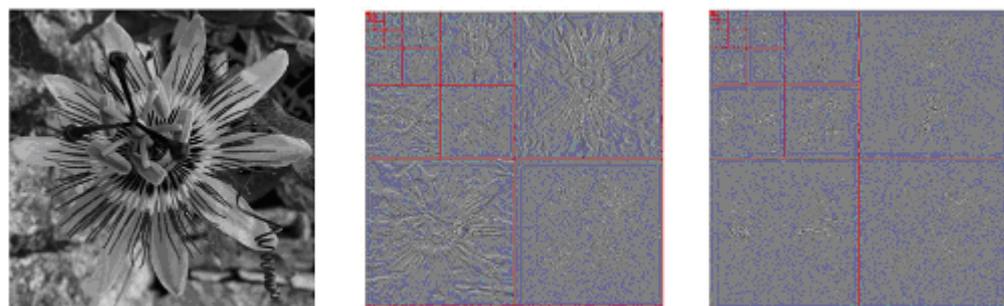


Figure 2.9. Exemples d'ondelettes de Daubechies pour différents ordres.



(a) Originale

(b) Haar

(c) Daubechies 8

Figure 2.10. Décomposition multi-résolution sur différentes bases d'ondelettes.

2.3. Ondelettes géométriques 2D

La transformée en ondelettes pour la représentation d'images mérite d'être améliorée pour représenter efficacement les structures non horizontales et non verticales. De nouvelles

transformées géométriques ont été développées dans l'intention de corriger cette limitation. On trouve deux grandes approches : l'approche non-adaptative et l'approche adaptative [49]. Les transformées géométriques non adaptatives utilisent des bases de fonctions fixes. On trouve dans cette famille les transformées en ridgelettes, curvelettes et contourlettes [50]. Les transformées dites adaptatives adaptent soit l'image à la transformée en ondelettes 2D, soit la transformée en ondelettes au contenu de l'image. Parmi ces transformées, on trouve à titre d'exemple la transformée en bandelettes [51]. La figure (2.11) résume la classification des ondelettes géométriques. Dans ce qui suit, on présente les ridgelettes, les curvelettes et les contourlettes.

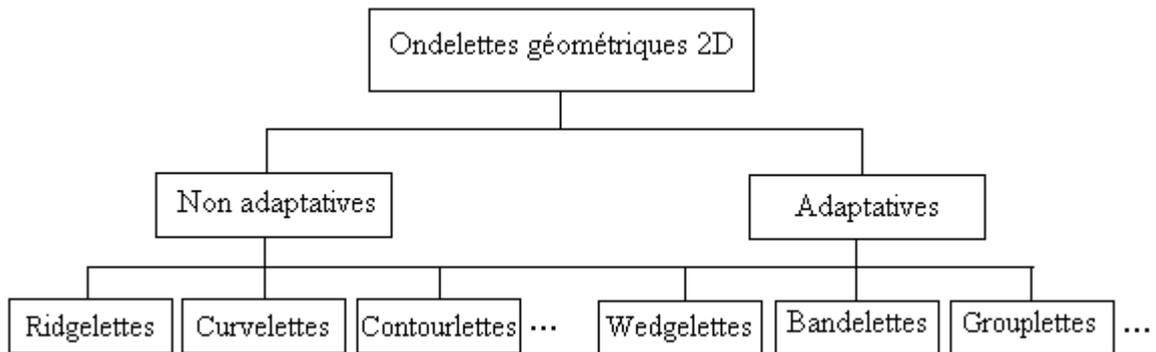


Figure. 2.11. Classification des ondelettes géométriques.

2.3.1. Transformée en Ridgelettes

L'extension au cas des images (2D) des ondelettes ne fait qu'utiliser le principe de la séparabilité avec une même ondelette 1D en filtrant horizontalement, puis verticalement l'image. L'analyse de l'image se fait alors suivant trois directions principales ; horizontale, verticale et oblique. Le contenu d'une image ne se limite pas seulement à ces trois directions, Candès a proposé dans [12] une nouvelle transformée plus générale prenant en compte les limites de la transformée en ondelettes. Cette nouvelle transformée est nommée la transformée en ridgelettes .

2.3.1.1. Principe de la transformée en ridgelettes

La ridgelette est définie comme une ondelette construite le long d'une ligne 'ridge' d'orientation θ définie dans le plan cartésien (x_1, x_2) par l'équation $x_1 \cos \theta + x_2 \sin \theta = cste$:

$$\psi_{a,b,\theta}(x_1, x_2) = a^{-1/2} \psi((x_1 \cos \theta + x_2 \sin \theta - b) / a) \quad (2.19)$$

où $a > 0$ est le facteur d'échelle, $b \in \mathbb{R}$ le paramètre de translation, $\theta \in [0, 2\pi[$ l'angle de projection et ψ la base d'ondelettes.

Les coefficients de ridgelettes Rid_I d'une image I sont obtenus par projection sur cette base :

$$Rid_I(a, b, \theta) = \iint \psi_{a, b, \theta}^*(x_1, x_2) I(x_1, x_2) dx_1 dx_2 \quad (2.20)$$

Cette projection est intimement liée à la transformée de Radon (T.Rad) qui consiste à intégrer une image selon un ensemble de lignes :

$$Rad_I(\theta, t) = \iint I(x_1, x_2) \delta(x_1 \cos \theta + x_2 \sin \theta - t) dx_1 dx_2 \quad (2.21)$$

où δ est la distribution Dirac.

Par conséquent, la transformée en ridgelettes (2.20) est obtenue par une transformée en ondelettes 1D (TO1D) des lignes ($\theta = cste$) de la transformée de Radon de l'image.

$$Rid_I(a, b, \theta) = \int Rad_I(\theta, t) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt \quad (2.22)$$

La transformée de Radon peut être calculée à partir de la transformée de Fourier inverse de la transformée de Fourier de l'image le long des lignes radiales.

$$\hat{I}(\lambda \cos \theta, \lambda \sin \theta) = TF[Rad_I(\theta, t)] = \int Rad_I(\theta, t) e^{-\lambda i t} dt \quad (2.23)$$

Par conséquent, l'équation (2.22) peut être écrite dans le domaine fréquentiel :

$$Rid_I(a, b, \theta) = TF_{\lambda}^{-1} \left[\hat{I}(\lambda \cos \theta, \lambda \sin \theta) TF_t[\psi(t/a)] \right] \quad (2.24)$$

La transformée en ridgelettes discrète s'obtient en appliquant une transformée en ondelettes 1D le long de $Rid_I(a, b, \theta)$, en utilisant la variable d'intégration t . Pour cela, on suit les trois étapes suivantes:

- Calcul de la transformée de Fourier 2D (FFT 2D)
- Conversion de ces ordonnées cartésiennes en coordonnées polaires.
- Calcul de la transformée inverse 1D (IFFT 1D).

Autrement dit, la transformée en ridgelettes est obtenue par une transformée en ondelettes 1D des lignes ($\theta = cste$) de la transformée de Radon de l'image. L'une des méthodes pour calculer la transformée de Radon est d'utiliser le théorème de projection des coupes: on calcule la transformée de Fourier 2D de l'image, on transforme cette image en une image en coordonnées polaires (droites passant par les fréquences nulles avec différentes orientations), puis on applique une transformée de Fourier inverse 1D sur chacune des lignes (pour θ). Il ne reste ensuite qu'à ajouter une transformée en ondelettes 1D suivant ces mêmes lignes pour obtenir la transformée en ridgelettes. La figure (2.12) résume les étapes d'une transformée en ridgelettes.

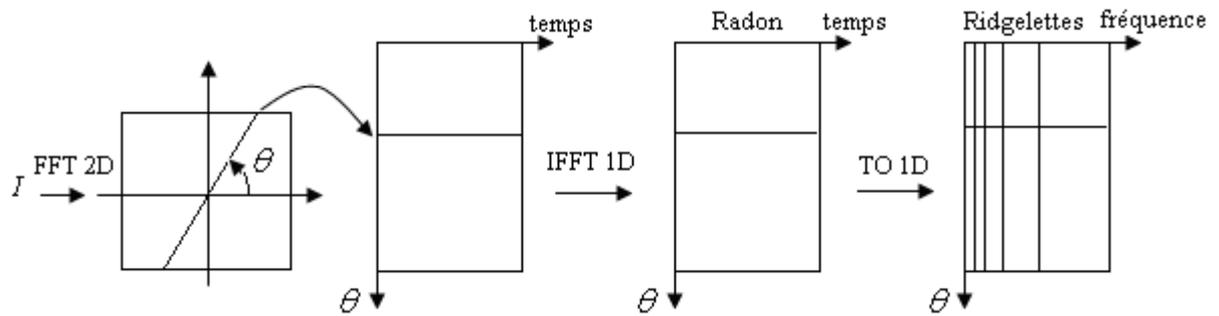


Figure.2.12. Construction de la transformée en ridgelettes par le théorème de projection des coupes.

D'après l'équation (2.20), on peut remarquer que la transformée en ridgelettes autorise une analyse uniquement le long des droites traversant complètement l'image. Ce qui en résulte que les contours dans l'image sont modélisés par des segments. Pour remédier ce problème, Candès et Donoho proposent d'effectuer la transformée en ridgelettes sur des blocs images de taille $B \times B$ se recouvrant de $B/2$ pour éviter les effets de blocs au lieu de l'image globale [41].

2.3.1.2. Algorithme de la transformée en ridgelettes

L'algorithme de la transformée en ridgelettes se déroule comme suit :

- 1- Effectuer la transformée de Fourier sur l'image I ;
- 2- Ré-échantillonner l'espace fréquentiel cartésien en coordonnées polaires;
- 3- Pour chaque ligne radiale indexée par θ_n , effectuer la transformée de Fourier inverse pour obtenir la transformée de Radon $\{Rad_I(\theta_n, t_n)\}_{n=1, \dots, N}$ pour l'angle θ_n .
- 4- Effectuer la transformée en ondelettes dans le domaine spatial (équation 2.22);

La transformée en ridgelettes inverse effectue simplement les transformations inverses sur chaque étape 4 à 1.

2.3.2. Transformée en curvelettes

2.3.2.1. Principe de la transformée en curvelettes

L'idée de la transformée en curvelettes est d'appliquer la décomposition pyramidale en ridgelettes non plus sur l'image elle-même, mais sur chacune des sous-bandes obtenues en sortie de la transformée en ondelettes non décimées de l'image. Pour cela, une analyse multi-résolutions est appliquée à l'image avant de lui appliquer la transformée en ridgelettes localement sur des blocs dyadiques.

2.3.2.2. Algorithme de la transformée en curvelettes

L'algorithme de la transformée en curvelettes est le suivant :

- 1- Effectuer l'analyse multi-échelles en différents niveaux J

$$I(x, y) = A_I^J(x, y) + \sum_{j=1}^{J-1} HF_I^j(x, y) \quad (2.25)$$

où A_I^J est la version grossière de l'image I , et HF_I^j sont les détails de I à l'échelle 2^{-j} .

- 2- Initialiser la taille des blocs à $B^1 = B_{\min}$ (on utilise $B_{\min} = 16$).

- 3- Pour chacune des images hautes fréquences $HF_I^j, j=1:J$, effectuer la transformation en ridgelettes locale :

- Si $j \bmod 2 = 1$, $B^{j+1} = 2B^j$;
- Sinon, $B^{j+1} = B^j$.

Les figures 2.13 et 2.14 présentent le principe de la transformée en curvelettes ainsi qu'un exemple de décomposition en curvelettes d'une image en 3 niveaux.

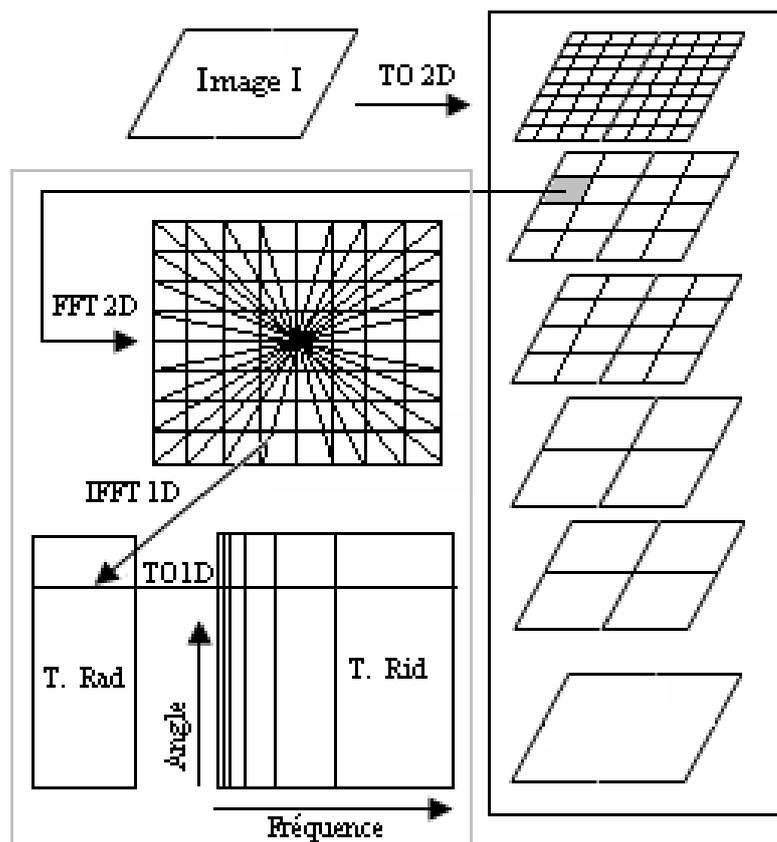


Figure. 2.13. Principe de la transformée en curvelettes.

La reconstruction est obtenue en déroulant l'algorithme dans l'ordre inverse (transformée en ridgelettes inverse, reconstruction de la partition initiale de l'image, filtrage dual pour la reconstruction de l'image à partir des sous-bandes).

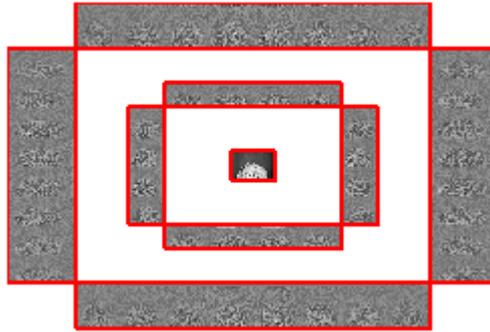


Figure. 2.14. Décomposition en curvelettes d'une image en 2 niveaux.

2.3.3. Transformée en contourlettes

Cette transformée proposée par Do *et al.*[46] correspond à une variante de la transformée en curvelettes en utilisant un banc de filtres directionnels proposé par Bamberger *et al.* [52] à la place de la transformée en ridgelettes. Une autre différence avec les curvelettes réside dans l'utilisation de la décimation au fur et à mesure de la montée en échelle lors de l'analyse fréquentielle.

2.3.3.1. Décomposition en sous-bandes

La première étape de la décomposition fait appel à une pyramide laplacienne (PL) de Burt *et al.* [53]. Le principe de cette décomposition d'une image I est le suivant [41] :

- 1- On génère une version à résolution plus faible de I par un filtrage passe-bas.
- 2- On sous-échantillonne cette nouvelle image, on obtient alors une image c .
- 3- On obtient l'image des détails en faisant la soustraction entre I et une prédiction de I obtenue par sur-échantillonnage et filtrage dual de c . L'image d ainsi obtenue correspond à une version de l'image d'origine filtrée par un filtre passe-bande.

La figure (2.15) décrit les étapes de décomposition et de reconstruction de l'image I .

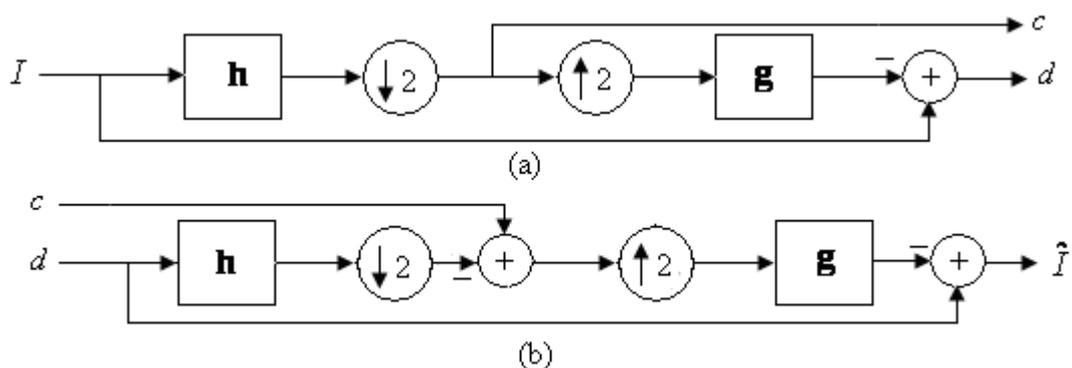


Figure 2.15. Schéma de décomposition (a) et de reconstruction (b) d'une pyramide laplacienne.

On applique cette décomposition à chaque niveau de résolution pour obtenir la structure de la pyramide. La reconstruction est le pseudo-inverse de l'algorithme de décomposition (valable uniquement dans le cas où les filtres sont orthogonaux), l'image reconstruite sera notée \hat{I} . La décomposition suivant la pyramide laplacienne permet d'obtenir un pavage du plan fréquentiel en bandes concentriques comme montré dans la figure (2.16).

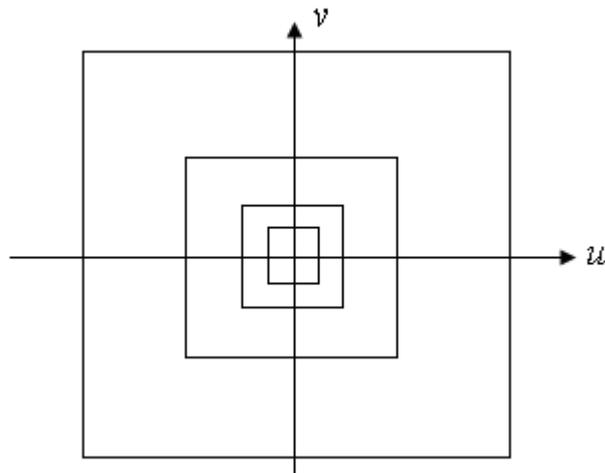


Figure 2.16. Pavage du plan fréquentiel obtenu par la décomposition pyramidale laplacienne.

2.3.3.2. Filtrage directionnel

Le principe du filtrage directionnel 2D utilisé par Do et Vetterli [44] est de fournir les réponses à travers un banc de filtres orientés d'une image d'entrée d , comme il est montré dans la figure (2.17a). Ce filtrage directionnel permet d'obtenir un pavage par tranches orientées du plan fréquentiel. Ceci est illustré dans la figure (2.17b).

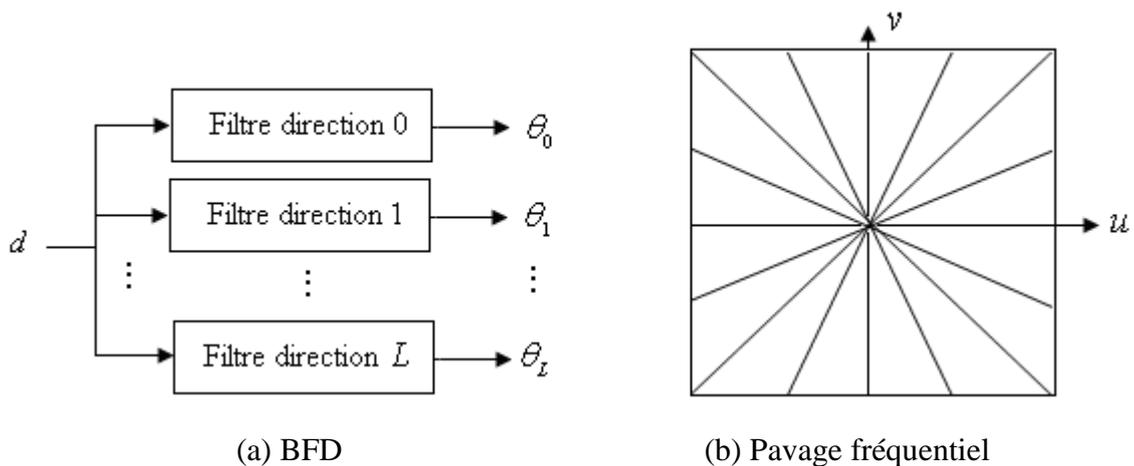


Figure 2.17. Banc de filtres directionnels et son pavage fréquentiel.

Do et Vetterli combinent les aspects d'analyse multi-résolutions et de filtrage directionnel pour construire la transformée en contourlettes appelée aussi 'Banc pyramidal de filtres directionnels BPDF'. La figure (2.18) montre le principe de la transformée en contourlettes. Il

consiste à appliquer un filtrage directionnel sur chacune des images de détails issues de la décomposition pyramidale. On obtient alors un pavage du plan fréquentiel où chaque couronne représentant un niveau de résolution est elle-même redécoupée en portions correspondant aux directions que l'on se fixe pour chaque sous-bande (voir la figure (2.19)).

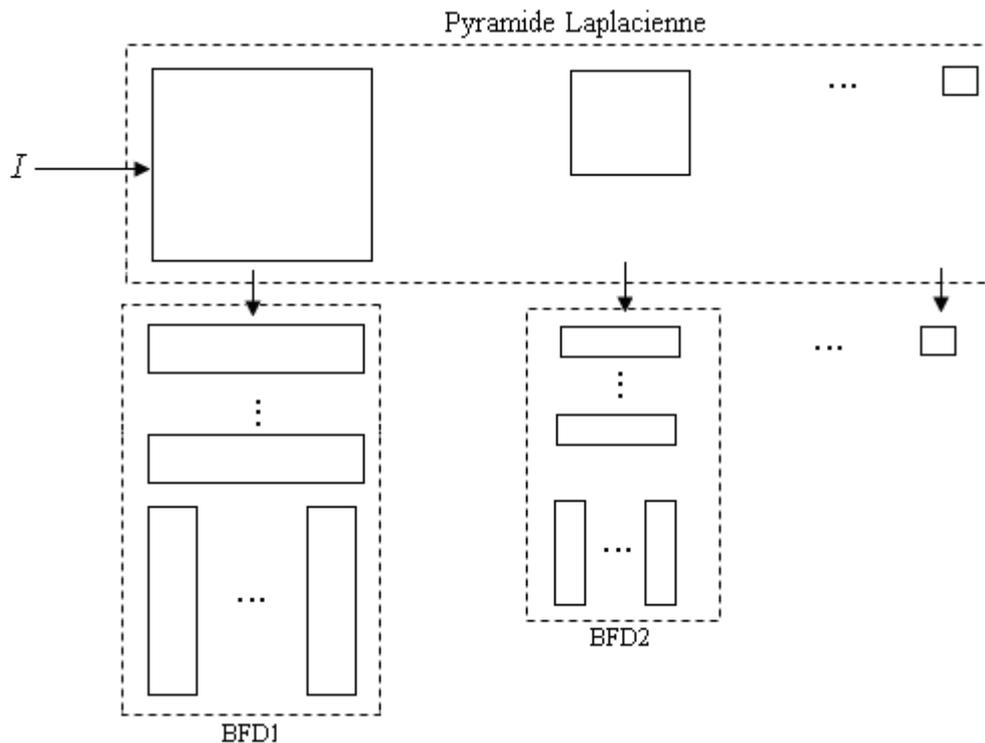


Figure 2.18. Principe de la transformée en contourlettes.

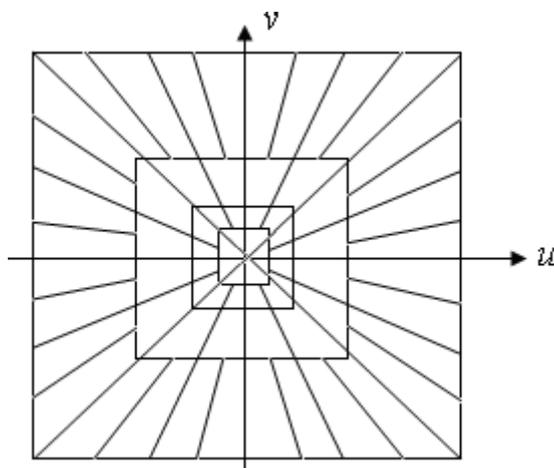


Figure 2.19. Découpage du plan fréquentiel pour la transformée en contourlettes.

La figure (2.20) donne un exemple de transformées en contourlettes. En haut à droite, on trouve l'approximation à basse résolution, puis de droite à gauche on voit apparaître trois sous-bandes elles-mêmes filtrées respectivement suivant 8, 8 et 16 directions [41].

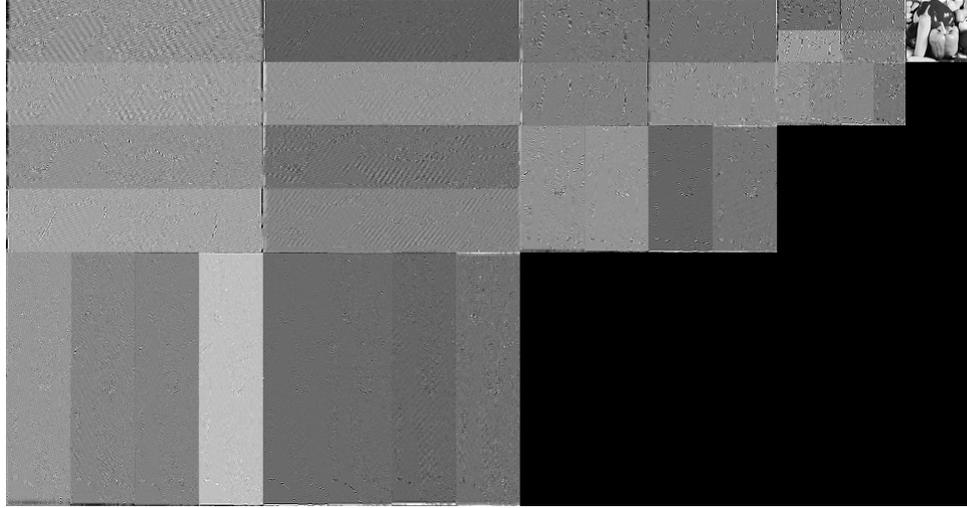


Figure 2.20. Exemple de transformée en contourlettes d'une image.

2.3.4. Pouvoir d'approximation

La théorie de l'approximation étudie le taux d'erreur résultant entre une fonction et son approximée. Pour les M plus grands coefficients de la transformée en ondelettes (curvelettes, contourlettes), le taux d'erreur après reconstruction est donné respectivement par [13, 41,46]:

$$\left\| f - \hat{f}_M^{ond} \right\|_2^2 \leq O(M^{-1}) \quad (2.26)$$

$$\left\| f - \hat{f}_M^{curv} \right\|_2^2 \leq O(M^{-2}(\log M)^3) \quad (2.27)$$

$$\left\| f - \hat{f}_M^{cont} \right\|_2^2 \leq O(M^{-2}(\log M)^3) \quad (2.28)$$

où \hat{f}_M^{ond} , \hat{f}_M^{curv} , \hat{f}_M^{cont} désignent les fonctions reconstruites à partir des M plus grands coefficients de la transformée en ondelettes, en curvelettes et en contourlettes respectivement. On note ici que plus M est grand, l'erreur d'approximation est faible. De même, l'approximation utilisant la transformée en curvelettes est meilleure que celle obtenue par la transformée en ondelettes.

2.4. Débruitage d'images par seuillage

Le débruitage est un processus qui consiste à éliminer le bruit avec une dégradation minimale des contours de l'image. La complexité de cette tâche est toujours liée à la nature du bruit et au type de l'image. Les méthodes standard de débruitage appliquées dans le domaine spatial aussi bien que dans celui du domaine fréquentiel de l'image bruitée présentent des

limitations pour la réduction de bruit et la qualité de l'image restaurée qui est généralement floue.

Motivées par la simplicité d'application et l'efficacité de la transformée en ondelettes, des méthodes basées sur le seuillage par transformée d'ondelettes ont vu le jour [54]. Ces techniques sont basées sur la construction d'estimateurs statistiques à base d'ondelettes et nécessitent essentiellement le calcul d'un seuil qui correspond à l'amplitude maximale du bruit, puis faire le seuillage dans le sens que chaque coefficient d'ondelettes de l'image est comparé au seuil calculé; si le coefficient est plus petit que le seuil, alors il est mis à zéro, autrement il est inchangé ou son amplitude est réduite légèrement.

Soit y une image corrompue par un bruit n . On peut ainsi écrire :

$$y = x + n \quad (2.29)$$

La transformée en ondelettes (TO) étant une fonction linéaire, on l'applique sur l'équation (2.29) et on obtient :

$$Wy = Wx + Wn \quad (2.30)$$

W symbolise la transformée en ondelettes (ou curvelettes ou ridgelettes). Soit $T(\cdot)$ la fonction de seuillage par ondelettes, alors le schéma de débruitage par ondelettes peut être exprimé selon l'équation:

$$\hat{x} = W^{-1}(T(Wy)) \quad (2.31)$$

où $T(Wy)$ est le vecteur des coefficients de la TO souillée et \hat{x} l'image débruitée. Ceci s'interprète simplement par l'application de TO sur l'image bruitée y , on obtient ainsi le vecteur des coefficients de la TO (Wy) sur lequel on applique la fonction de seuillage $T(Wy)$. L'image débruitée \hat{x} quant à elle, est obtenue en appliquant la transformée en ondelettes inverse (TOI) sur le vecteur des coefficients seuillés ($W^{-1}(T(Wy))$). La figure (2.21) présente les étapes de débruitage.

2.4.1. Algorithme de débruitage par seuillage

L'algorithme de base de débruitage par ondelettes peut être décomposé en trois étapes essentielles:

- La décomposition par la transformée en ondelettes.
- Le seuillage des coefficients issus par la décomposition.
- La reconstruction par la transformée en ondelettes inverse.

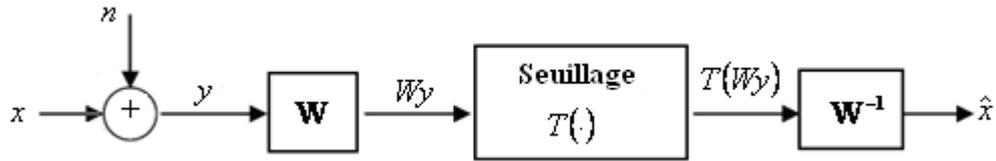


Figure 2.21. Trois étapes de débruitage par seuillage .

2.4.2. Méthodes de seuillage

Dans la littérature, on trouve plusieurs méthodes de seuillage, dont les plus connues et appliquées sont :

2.4.2.1. Seuillage doux

Le seuillage doux consiste à éliminer les coefficients au dessous d'un seuil T et de soustraire ce seuil des autres coefficients. La fonction du seuillage s'écrit [54] :

$$T_{doux}(x) = \begin{cases} \text{sign}(x)(|x| - T) & |x| \geq T \\ 0 & |x| < T \end{cases} \quad (2.32)$$

où, $T_{doux}(\cdot)$ est la fonction de seuillage, x sont les coefficients de détails de la transformée en ondelettes.

2.4.2.2. Seuillage dur

Le seuillage dur est plus catégorique, un coefficient donné est considéré soit comme représentant du bruit pur ou représentant une portion du signal saint. La fonction de seuillage dans ce cas est de la forme [54]:

$$T_{dur}(x) = \begin{cases} x & |x| > T \\ 0 & |x| \leq T \end{cases} \quad (2.33)$$

Les deux fonctions de seuillage ci-dessus ont des avantages et des inconvénients ; le seuillage doux n'est pas efficace pour les grands coefficients, le seuillage dur présente des discontinuités d'où les résultats ont tendance à avoir une grande variance et être instables. Pour remédier aux inconvénients du seuillage dur et doux, plusieurs versions de fonctions de seuillage ont été proposées.

2.4.2.3. Seuillage de Breiman

La fonction de seuillage proposée par Breiman [55] est de la forme :

$$T_B(x) = \begin{cases} x - \frac{T^2}{x} & |x| > T \\ 0 & |x| \leq T \end{cases} \quad (2.34)$$

2.4.2.4. Seuillage de Bruce et Gao

La fonction de seuillage proposée par Bruce et Gao [56] est de la forme :

$$T_{G-B}(x) = \begin{cases} 0 & |x| \leq T_1 \\ \text{sign}(x) \frac{T_2(|x| - T_1)}{T_2 - T_1} & T_1 < |x| \leq T_2 \\ x & |x| > T_2 \end{cases} \quad (2.35)$$

Cette fonction de seuillage groupe les avantages du seuillage dur et doux, mais son inconvénient réside qu'elle exige deux seuils. Cela alourdit d'avantage les procédures du choix de seuil [57].

2.4.2.5. Seuillage de Kwon

La fonction de seuillage proposée par Kwon [58] est de la forme :

$$T_K(x) = \begin{cases} \frac{1}{\mu} \text{sign}(x) \cdot T \cdot ((1 + \mu)^{\frac{|x|}{T}} - 1) & |x| < T \\ \mu & |x| < T \\ x & |x| > T \end{cases} \quad (2.36)$$

μ est une constante

2.4.2.6. Seuillage de Zhang

Zhang a proposé l'utilisation d'une nouvelle fonction de seuillage douce [59] :

$$T_Z(x) = x + \frac{1}{2} \left(\sqrt{(x-T)^2 + \lambda} - \sqrt{(x+T)^2 + \lambda} \right) \quad (2.37)$$

où λ est un paramètre à fixer.

2.4.2.7. Seuillage de Mehdi *et al.*

Mehdi *et al.* [60] proposent la fonction de seuillage suivante :

$$T_M(x) = \begin{cases} x - 0.5 \frac{T^2}{x} & |x| > T \\ 0.5 \frac{x^3}{T^2} & |x| \leq T \end{cases} \quad (2.38)$$

La figure (2.22) présente quelques fonctions de seuillages.

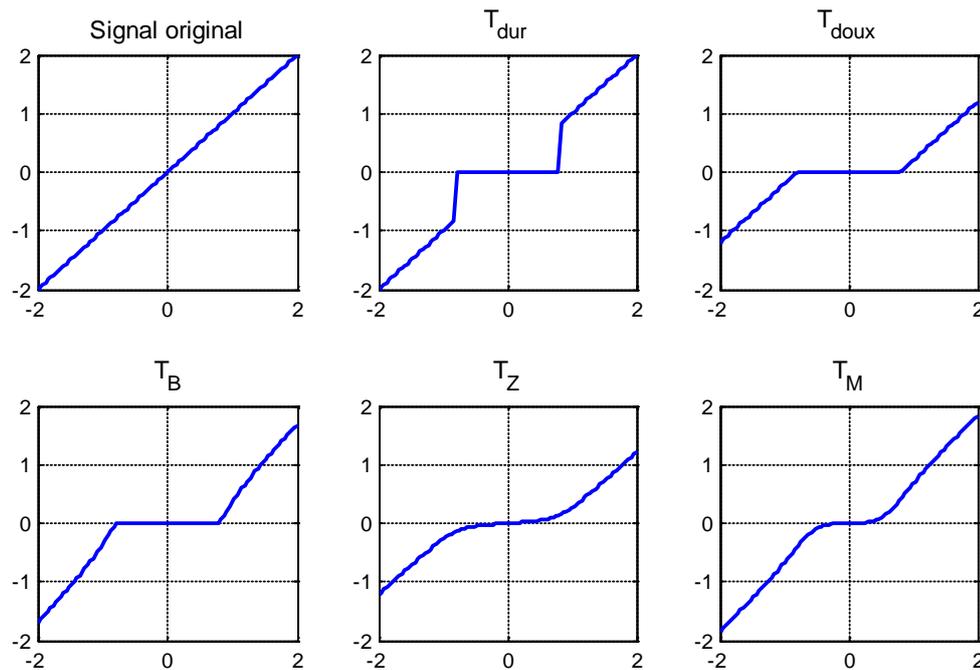


Figure 2.22. Quelques fonctions de seuillages.

2.4.3. Sélection du seuil

Le calcul du seuil est une tâche très importante dans la mesure où on estime le niveau du bruit en fonction de sa nature et la nature de l'image. On distingue les différents cas suivants :

1. Seuillage global.
2. Seuillage dépendant du niveau.
3. Seuillage dépendant du noeud.

2.4.3.1. Seuillage global

Appliquer un seuillage global revient à estimer un seuil unique quelque soit le niveau de décomposition. Le seuil est calculé en utilisant l'équation [54]:

$$T = \sigma \sqrt{2 \log(N)} \quad (2.39)$$

où N désigne le nombre de points du signal (taille de l'image) et σ la variance du bruit qui est estimée à partir de la valeur médiane des coefficients d_1 de la sous-bande de détail diagonale du premier niveau de décomposition par l'équation suivante :

$$\sigma = \frac{\text{median}(d_1)}{0.6745} \quad (2.40)$$

Ce seuil est performant si le signal informatif n'est pas très riche en détail, c'est-à-dire si ces coefficients sont rares.

2.4.3.2. Seuillage dépendant du niveau

On calcule un seuil pour chaque niveau de décomposition. Il est donné par la formule:

$$T_j = \sigma_j \sqrt{2 \log(N)} \quad (2.41)$$

où j représente le niveau de décomposition et σ_j l'écart-type du bruit calculé dépendamment du niveau en question qui est donné par la formule:

$$\sigma_j = \frac{\text{median}(|d_1^j|)}{0.6745} \quad (2.42)$$

où σ_1^j représente les coefficients des détails obtenus au niveau j correspondant au nœud de la plus haute résolution.

2.4.3.3. Seuillage dépendant du nœud

Une extension du seuillage dépendant du niveau est celui du seuillage dépendant du nœud qui consiste à effectuer un calcul du seuil pour chaque sous-bande [54]. Le seuil est calculé en utilisant la formule :

$$T_{j,k} = \sigma_{j,k} \sqrt{2 \log(N)} \quad (2.43)$$

Où j représente le niveau de décomposition, k la sous-bande, et $\sigma_{j,k}$ l'écart-type du bruit calculé dépendamment du niveau et la sous-bande en question qui est donné par la formule :

$$\sigma_{j,k} = \frac{\text{median}(|d_k^j|)}{0.6745} \quad (2.44)$$

Où d_k^j représentent les coefficients obtenus au niveau j et la sous-bande k .

2.5. Conclusion

Dans ce chapitre, nous avons rappelé la théorie de la transformée en ondelettes continues et discrètes d'un signal monodimensionnel et bidimensionnel. Ensuite nous avons présenté des nouveaux outils d'analyse multi-résolutions directionnels dits "ondelettes géométriques". Ces nouvelles techniques sont regroupées en deux grandes familles: les ondelettes géométriques adaptatives et non adaptatives (fixes). Ces dernières, contenant les ridgelettes, les curvelettes et les contourlettes, sont largement détaillées. Pour achever ce chapitre, une application des ondelettes et ses variantes directionnelles dans le débruitage d'images bruitées a été présentée.

Chapitre 3

Méthode de Lucas & Kanade multi-échelles par la transformée en curvelettes

3.1. Introduction

Les techniques multi-échelles sont largement employées pour la mesure du mouvement [9,10]. On trouve deux approches dans la littérature: la première utilise les niveaux de résolution simultanément, dans une approche ‘coarse and fine’ alors que la seconde, elle utilise les niveaux de résolution séquentiellement, du plus grossier au plus fin, dans une approche ‘coarse to fine’. La technique de Lucas et Kanade multi-échelles est l’une des méthodes différentielles les plus utilisées [7,10,61]. Elle offre une robustesse au bruit et une grande capacité à estimer les grands déplacements. Pour améliorer les performances de la méthode de Lucas et Kanade, deux points critiques sont à considérer : la précision et le temps de calcul. Pour la précision, nous avons introduit la transformée en curvelettes pour un prétraitement des images. Cette transformée, qui est un outil très intéressant, a fait ses preuves dans le domaine du traitement d’images. Quant à la rapidité, nous avons proposé d’utiliser deux types de réseaux de neurones récurrents, le réseau de Hopfield [62] et le réseau de Zhang [63-65] pour l’estimation du flot optique de tous les pixels simultanément au lieu de son calcul séquentiel dans la méthode de Lucas & Kanade.

3.2. Approches existantes de LK

3.2.1. Méthode de LK originale

Lucas et Kanade [7] ont proposé une méthode qui impose une contrainte locale de lissage. Ils ont supposé que l’équation du flot optique demeurerait constante dans de petites régions de l’image Ω . Ceci est exprimé par :

$$E = \min \sum_{p \in \Omega} w^2(p) (I_x(p)u + I_y(p)v + I_t(p))^2 \quad (3.1)$$

avec $I(x, y, t)$ est une séquence d'image, $V(p) = (u, v)^T$ le vecteur vitesse associé au point $p = (x, y)^T$ à l'instant t . Les composantes u et v sont respectivement la vitesse selon les directions x et y . I_x , I_y et I_t sont les dérivées spatio-temporelles de $I(x, y, t)$. Les valeurs $w(p)$ sont définies de telle sorte qu'elles donnent plus d'influence aux pixels situés au centre de Ω qu'à ceux situés à la périphérie du voisinage [66].

La solution de l'équation (3.1) se traduit sous forme matricielle de la manière suivante :

$$A^T W^2 A V = A^T W^2 b \quad (3.2)$$

où, pour n pixels dans le voisinage Ω , nous notons :

$$A = (\nabla I(p_1), \dots, \nabla I(p_n))^T, \quad W = \text{diag}[w(p_1), \dots, w(p_n)] \text{ et}$$

$$b = -(I_t(p_1), \dots, I_t(p_n))^T. \quad \nabla I \text{ est le gradient spatial de } I.$$

En appliquant la méthode de calcul des moindres carrés, nous obtenons:

$$V = G^{-1} d \quad (3.3)$$

avec :

$$G = A^T W^2 A = \begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_x I_y & \sum W^2 I_y^2 \end{bmatrix} \quad (3.4)$$

$$d = A^T W^2 b = - \begin{bmatrix} \sum W^2 I_x I_t \\ \sum W^2 I_y I_t \end{bmatrix} \quad (3.5)$$

L'inconvénient de cette méthode est que dans certains cas, l'inversion de la matrice G n'est pas possible ou peut être instable.

Afin de rendre cette méthode plus robuste, nous devons mesurer la stabilité numérique du calcul par un autre moyen tels que l'étude du déterminant ou les valeurs propres de la matrice G . Dans [4], Barron *et al.* ont proposé d'utiliser les valeurs/vecteurs propres:

Si la plus petite valeur propre (λ_1) de la matrice G est supérieure à un seuil T , alors le vecteur V est exprimé par la relation (3.3).

Si $\lambda_1 < T$, alors le vecteur V n'est pas calculé sur ce site, car l'inversion de la matrice est jugée instable ($V = [0,0]^T$).

Si $\lambda_1 \geq T$ et $\lambda_2 < T$, nous calculons le vecteur normale par la relation $V_N = (V \cdot e_2) \cdot e_2$ où e_2 est le vecteur propre correspondant à la valeur propre λ_2 .

Algorithme de Lucas & Kanade (LK) :

Début

(1) Initialisation :

- Choix de la taille du voisinage $\Omega = n \times n$.
- Choix de la fonction de pondération w .
- Choix d'un seuil T .
- Calculer toutes les dérivées spatiales et temporelles I_x , I_y et I_t .
- Calculer les éléments de la diagonale de la matrice des poids W .

(2) Pour chaque pixel de l'image analysée :

❖ **Création du système linéaire:** (Création d'un système linéaire (3.3) sur une fenêtre de taille $n \times n$, autour du pixel analysé).

- Calculer le vecteur d .
- Calculer la matrice G .

❖ **Résolution du système :**

- Calculer les deux valeurs propres (λ_1, λ_2) et sélectionner la plus petite valeur (λ_1)

Si $\lambda_1 > T$ alors : $V \leftarrow G^{-1}d$

Si $\lambda_1 < T$ alors : $V \leftarrow [0,0]^T$

Si $\lambda_1 \geq T$ et $\lambda_2 < T$ alors :

- Calcul du vecteur propre correspondant à λ_2 .
- $V_N \leftarrow (V \cdot e_2) \cdot e_2$

Fin pour

Fin

Cette méthode est intéressante, car elle est hautement parallélisable (chaque calcul sur une petite fenêtre est indépendant des autres) et moins sensible au bruit. Elle permet aussi le calcul de mouvements locaux. Cependant, elle est inadaptée lorsqu'il s'agit de traiter de grands déplacements. Ce problème est résolu à l'aide l'implémentation pyramidale.

3.2.2. Méthode pyramidale de Lucas & Kanade

Lorsque les amplitudes des déplacements sont grandes ou inconnues à priori, il est judicieux d'effectuer l'analyse sur plusieurs bandes de fréquences, ce qui fait appel à une représentation multi-échelles [67]. Les déplacements à amplitudes différentes peuvent donc être estimés à des niveaux de résolution différents. Cependant, plus la résolution est grossière moins les estimations obtenues sont précises. Par conséquent, les déplacements de grands objets sont estimés avec peu de précision et il serait intéressant de pouvoir combiner les résultats obtenus dans des sous-images différentes, afin d'ajuster les estimations initiales [10]. Soit $V_k = (u_k, v_k)^T$ le flot optique estimé au niveau k de la pyramide d'images, le mouvement réel peut s'écrire comme la somme de V_k et d'un mouvement résiduel ΔV_k à déterminer :

$$V = V_k + \Delta V_k \quad (3.6)$$

Connaissant V_k , l'hypothèse de la conservation de la luminance s'écrit :

$$I(p + V_k + \Delta V_k, t + 1) - I(p, t) = 0 \quad (3.7)$$

à nouveau, l'équation (3.7) est linéarisée par un développement de Taylor, mais cette fois-ci autour de $p + V_k$:

$$\nabla I(p + V_k, t + 1) \Delta V_k + I_t(p, t) = 0 \quad (3.8)$$

Tout algorithme différentiel de calcul du flot optique pourra être utilisé pour estimer le déplacement résiduel ΔV_k . Pour que l'estimation soit plus précise, nous utilisons l'algorithme itératif de Lucas & Kanade [10,61]:

$$\Delta V_k = G^{-1} d \quad (3.9)$$

Durant chaque itération, la matrice G ne se change pas, donc elle est calculée une seule fois, et le vecteur d prend la valeur :

$$d^{it} = - \begin{bmatrix} \sum W^2 I_x I_t^{it} \\ \sum W^2 I_y I_t^{it} \end{bmatrix} \quad (3.10)$$

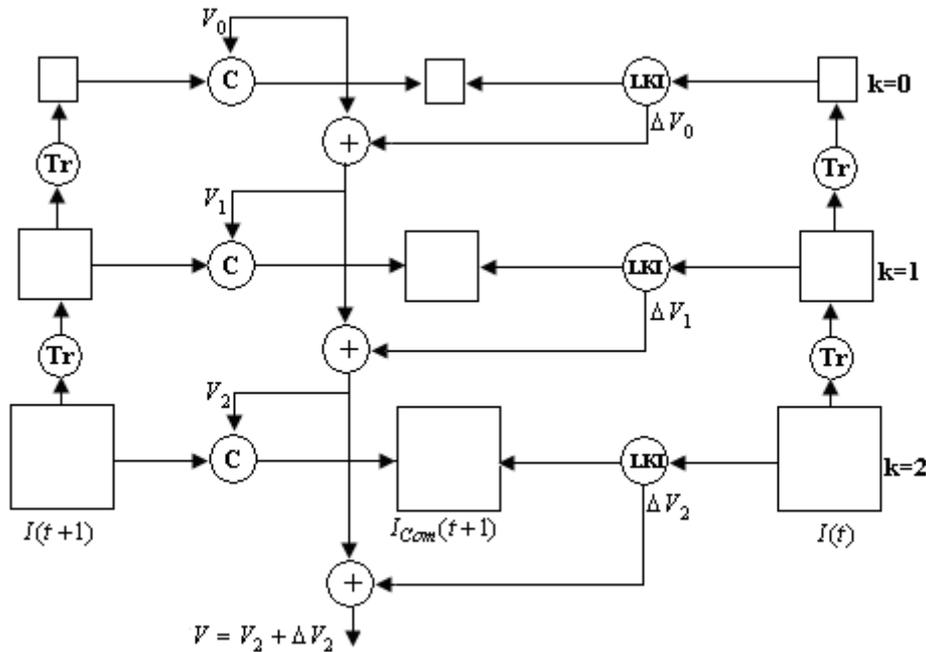
Dans l'itération suivante $it + 1$:

$$V_k^{it+1} = V_k^{it} + \Delta V_k \quad (3.11)$$

On arrête le processus itératif quand ΔV_k atteint un seuil fixé à l'avance ou pour un nombre d'itérations donné. Cette estimation incrémentale du mouvement est effectuée du sommet à la

base de la pyramide, c.-à-d. on projette le flot optique estimé au niveau de la résolution k à un niveau de résolution plus fin $k+1$ grâce à la formule de changement de base. La figure (3.1) illustre ce mécanisme.

L'image compensée $I_{com}(p,t+1) = I(p+V_k, t+1)$ est calculée à partir de l'image originale $I(p,t+1)$ par interpolation bilinéaire ou spline cubique, car V_k n'étant pas une grandeur entière.



Tr: Transformation Pyramidale, LKI: Algorithme itératif de LK, com: Compensation de mouvement

Figure 3.1. Schéma synoptique de l'estimateur hiérarchique de Lucas & Kanade.

3.3. Approches proposées

Pour le problème de l'estimation de mouvement dans une séquence d'images, deux performances critiques sont à améliorer, le temps de calcul et la précision des résultats. La méthode itérative multi-échelles de Lucas & Kanade (LKM) est l'une des meilleures méthodes dans ce domaine. Pour accélérer le processus d'estimation et dû à la nature parallèle des réseaux de neurones, on a introduit dans cette section deux types de réseaux de neurones récurrents: le réseau de Hopfield et le réseau de Zhang [68]. De plus, on a introduit un algorithme basé sur l'utilisation d'un masque d'interpolation pour le calcul de l'image compensée par le vecteur de mouvement [66]. Du côté précision des résultats, on a proposé d'utiliser la transformée en curvelettes pour la représentation multi-échelles et le préfiltrage des images [68].

3.3.1. Accélération de la méthode LKM

Pour accélérer le processus itératif, on a introduit un masque de convolution m pour calculer l'image compensée [69]:

$$I_{Com}^{it-1}(t+1) = I(t+1) * m \quad (3.12)$$

L'équation (3.10) devient :

$$d^{it} = - \left[\begin{array}{c} \sum W^2 I_x (I_{Com}^{it-1}(t+1) - I(t)) \\ \sum W^2 I_y (I_{Com}^{it-1}(t+1) - I(t)) \end{array} \right] \quad (3.13)$$

Sous ces hypothèses nous pouvons écrire :

$$\begin{aligned} \sum_{x,y} W^2 I_x(x,y) I_{Com}^{it-1}(x,y,t+1) &= \sum_{x,y} W^2 I_x(x,y) I(x,y,t+1) * m \\ &= \sum_{0 \leq i,j \leq 1} m(i,j) \sum_{x,y} W^2 I_x(x,y) I(x-i,y-j,t+1) = \sum_{0 \leq i,j \leq 1} m(i,j) s(i,j) \end{aligned} \quad (3.14)$$

Le même raisonnement peut se faire pour le deuxième composant du vecteur d^{it} . Puisque seuls les coefficients du masque m changent à chaque itération, les autres termes sont calculés une seule fois d'où un petit nombre d'opérations, qui dépend de la taille du masque, est nécessaire pour chaque itération [66].

3.3.2. Méthode neuronale de Lucas & Kanade

3.3.2.1. Reformulation de la méthode de LK

La méthode de LK nécessite l'inversion de la matrice G pour chaque pixel de l'image. Ainsi, nous obtenons le système suivant [68]:

$$\left[\begin{array}{cc} \sum W^2 I_{ix}^2 & \sum W^2 I_{ix} I_{iy} \\ \sum W^2 I_{ix} I_{iy} & \sum W^2 I_{iy}^2 \end{array} \right] \begin{bmatrix} u_i \\ v_i \end{bmatrix} = - \begin{bmatrix} \sum W^2 I_{ix} I_{it} \\ \sum W^2 I_{iy} I_{it} \end{bmatrix} \quad (3.15)$$

avec $i \in \{1, 2, \dots, N\}$ et N est la taille de l'image.

Pour estimer le flot optique de tous les pixels simultanément, nous posons [68]:

$$z = [z_1, \dots, z_{2N}]^T \quad (3.16)$$

$$\tilde{d} = [d_1, \dots, d_{2N}]^T \quad (3.17)$$

$$\tilde{G} = \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,2N} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,2N} \\ \vdots & \vdots & \ddots & \vdots \\ g_{2N,1} & g_{2N,2} & \cdots & g_{2N,2N} \end{bmatrix} \quad (3.18)$$

avec les coordonnées pour $i \in \{1, 2, \dots, N\}$:

$$\begin{cases} z_{2i-1} = u_i \\ z_{2i} = v_i \end{cases} \quad (3.19)$$

$$\begin{cases} d_{2i-1} = -\sum W^2 I_{ix} I_{it} \\ d_{2i} = -\sum W^2 I_{iy} I_{it} \end{cases} \quad (3.20)$$

$$\begin{cases} g_{2i-1, 2i-1} = \sum W^2 I_{ix}^2 \\ g_{2i, 2i} = \sum W^2 I_{iy}^2 \\ g_{2i-1, 2i} = g_{2i, 2i-1} = \sum W^2 I_{ix} I_{iy} \end{cases} \quad (3.21)$$

Nous obtenons le nouveau système :

$$\tilde{G}V = \tilde{d} \quad (3.22)$$

La nouvelle matrice \tilde{G} est de taille $2N \times 2N$ dont les éléments de la diagonale sont les éléments de la matrice G de l'équation (3.4).

Motivés par la nature parallèle de calcul des réseaux de neurones, nous avons proposé d'éviter l'inversion de la matrice \tilde{G} en résolvant directement le système d'équations (3.22) par le réseau de neurones de Hopfield ou le réseau de Zhang.

3.3.2.2. Réseau de Hopfield

L'idée d'introduire le réseau de neurones de Hopfield (HNN :: Hopfield neural network) dans le problème de l'estimation du flot optique revient à trouver un lien entre une fonction de coût à minimiser et la fonction d'énergie du HNN données par (voir annexe A) :

$$E_H = -\frac{1}{2} X^T W X - b^T X \quad (3.23)$$

L'architecture du réseau de Hopfield est déduite à partir de l'équation (3.22). La solution de cette équation consiste à trouver le vecteur V qui minimise la fonction de coût définie par :

$$j(V) = \frac{1}{2} \|\tilde{G}V - \tilde{d}\|^2 \quad (3.24)$$

Cette fonction est équivalente après développement :

$$j(V) = \frac{1}{2} V^T \tilde{G}^T \tilde{G} V - \tilde{d}^T \tilde{G} V \quad (3.25)$$

Nous pouvons noter ici qu'un terme $\tilde{d}^T \tilde{d}$ a été ignoré dans l'expression (3.25) du fait qu'il ne dépend pas de V .

En comparant l'équation (3.23) et l'équation (3.25), nous pouvons obtenir la matrice des poids des connexions W et le vecteur des biais b du HNN. Le vecteur des états du HNN X correspond aux paramètres à estimer V . Ainsi, l'état stable du réseau obtenu après la convergence correspond bien aux amplitudes recherchées.

$$\begin{cases} W = -\tilde{G}^T \tilde{G} \\ b = \tilde{G}^T \tilde{d} \\ X = V \end{cases} \quad (3.26)$$

La dynamique du réseau est régie par l'équation différentielle [62]:

$$\frac{du_i}{dt} = \sum_{m=1}^{2N} w_{im} v_m + b_i \quad (3.27)$$

$$v_m(t) = f(u_m(t)), \quad m = 1, 2, \dots, 2N \quad (3.28)$$

où w_{im} est le poids de connexion à valeur complexe entre les deux neurones i et m et b_i la valeur du biais du neurone i . v_m est le signal de sortie du neurone m donné par $v_m = f(u_m)$ avec $f(\cdot)$ est une fonction d'activation.

Le modèle discret équivalent est obtenu par l'utilisation de la formule d'Euler dans l'expression (3.27).

$$u_i(k+1) = u_i(k) + h \left(\sum_{m=1}^{2N} w_{im} v_m + b_i \right) \quad (3.29)$$

$$v_m(k+1) = f(u_m(k+1)), \quad m = 1, 2, \dots, 2N \quad (3.30)$$

où h est le pas d'intégration.

L'implémentation du réseau dans ces deux versions continue et discrète est inspirée directement des équations (3.27) et (3.29). Elle est représentée sur la figure (3.2), avec $L = 2N$.

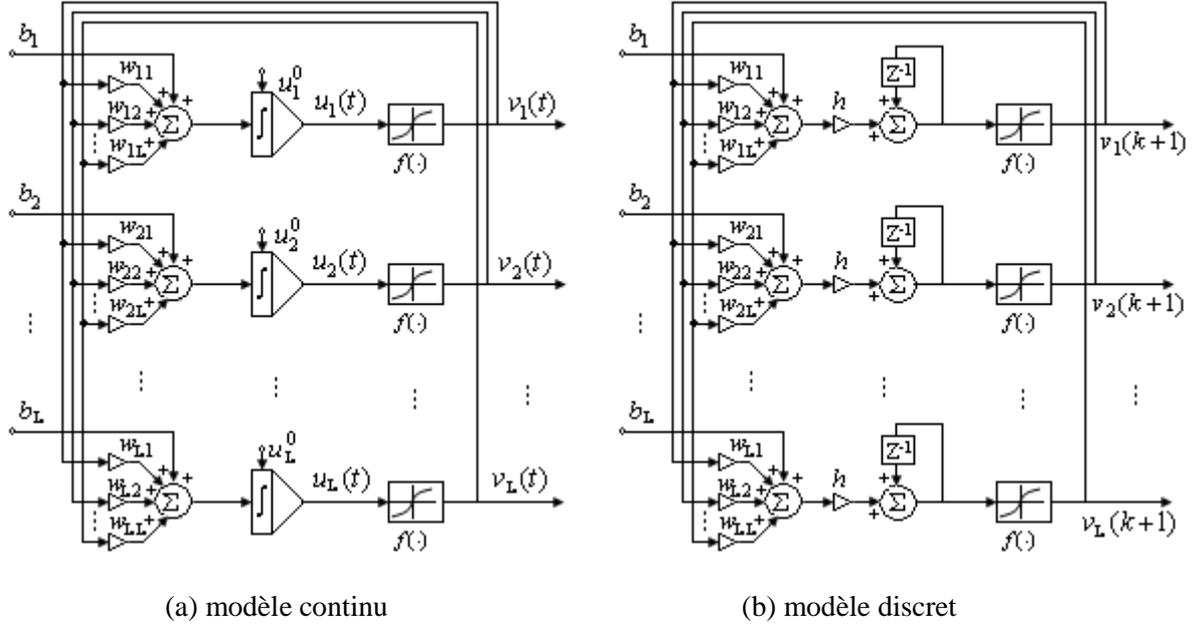


Figure 3.2. Architecture du réseau de Hopfield.

3.3.2.3. Réseau de neurones de Zhang

Pour résoudre l'équation $\tilde{G}V = \tilde{d}$ nous utilisons le réseau de Zhang dont le principe est de minimiser un vecteur-erreur de la forme [64] :

$$\boldsymbol{\varepsilon}(t) = \tilde{G}V - \tilde{d} \quad (3.31)$$

Le négatif du gradient dans la direction descendante:

$$\frac{d\boldsymbol{\varepsilon}(t)}{dt} = -\mathbf{\Gamma}f(\boldsymbol{\varepsilon}(t)) \quad (3.32)$$

où la matrice $\mathbf{\Gamma} \in \mathfrak{R}^{2N \times 2N}$ définie positive est utilisée pour modifier la cadence de convergence et $f(\cdot) : \mathfrak{R}^{2N} \rightarrow \mathfrak{R}^{2N}$ dénote le vecteur fonction d'activation.

Le développement des équations (3.31) et (3.32) nous permet d'écrire :

$$\tilde{G}\dot{V}(t) = -\mathbf{\Gamma}f(\tilde{G}V(t) - \tilde{d}) \quad (3.33)$$

Pour assurer que tous les éléments du vecteur erreur convergent avec la même cadence et dans le même temps, il suffit de prendre $\mathbf{\Gamma} = \gamma \cdot I$ avec $\gamma > 0$, ceci induit:

$$\tilde{G}\dot{V}(t) = -\gamma \cdot f(\tilde{G}V(t) - \tilde{d}) \quad (3.34)$$

Pour l'implémentation du réseau sur des calculateurs numérique, nous exprimons la dynamique du $i^{\text{ème}}$ neurone à partir de l'équation (3.34) sous la forme :

$$\dot{v}_i(t) = -\gamma \cdot f\left(\sum_{j=1}^{2N} g_{ij}v_j(t) - d_i\right) + (1 - g_{ii})\dot{v}_i(t) - \sum_{j=1, j \neq i}^{2N} g_{ij}\dot{v}_j(t) \quad (3.35)$$

En appliquant la formule d'Euler $\dot{v}_i(t) = (v_i((k+1)h) - v_i(kh))/h$, on obtient la version discrète du modèle de Zhang (DZNN : Discrete Zhang Neural Network) [70]. k et h sont l'indice de l'itération et le pas de discrétisation respectivement. La figure (3.3) présente l'architecture du réseau discret pour $L = 2N$.

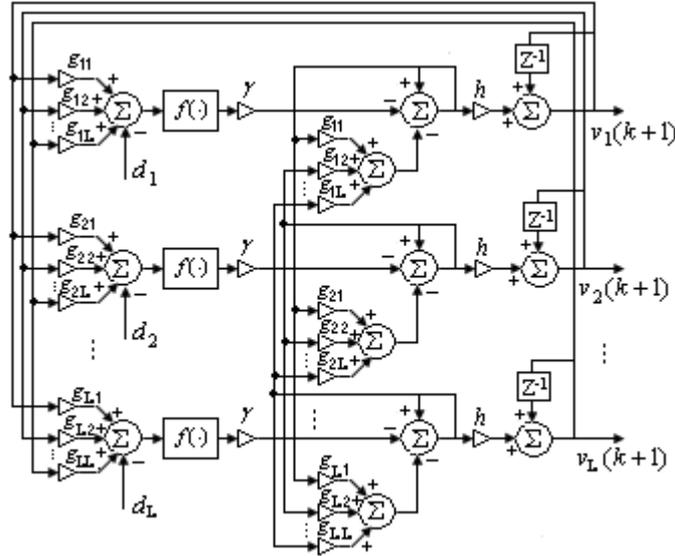


Figure 3.3. Architecture du réseau de neurones de Zhang discret.

On rencontre dans littérature, une diversité de fonction d'activation, dans notre travail, nous nous sommes limités aux deux fonctions les plus utilisées pour le réseau de Zhang ; la fonction d'activation linéaire:

$$f(\varepsilon_i) = \varepsilon_i \quad (3.36)$$

et la fonction signe – puissance [71-72]:

$$f(x) = \frac{1}{2} \left(sig^r(x) + sig^{\frac{1}{r}}(x) \right), 0 < r < 1 \quad (3.37)$$

$$\text{avec : } sig^r(x) = \begin{cases} |x|^r & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -|x|^r & \text{si } x < 0 \end{cases} \quad (3.38)$$

La figure (3.4) présente la courbe de la fonction signe-puissance pour différentes valeurs de r

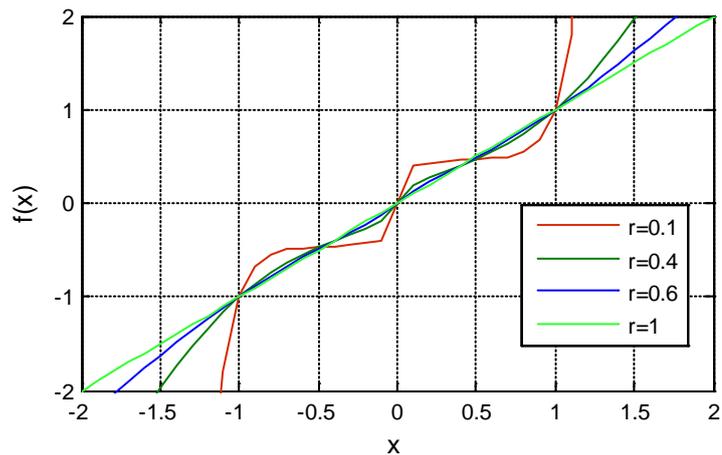


Figure 3.4. Courbe de la fonction signe-puissance pour différentes valeurs de r .

3.3.2.4. Algorithme de Lucas & Kanade neuronal (LKN) :

Début

(1) Initialisation :

- Choix de la taille du voisinage Ω .
- Choix de la fonction de pondération w .
- Calculer toutes les dérivées spatiales et temporelles I_x , I_y et I_t .
- Choix du réseau de neurones.
- Choix des paramètres du réseau.

(2) Pour tous les pixels de l'image analysée :

❖ Création du système linéaire: équation (3.22).

- Calculer la matrice \tilde{G} .
- Calculer le vecteur \tilde{d}

❖ Résolution du système :

- Utiliser le réseau de neurones pour résoudre le système (3.22).

Fin pour

Fin

3.3.3. Débruitage par la transformée en curvelettes

Pour rendre la méthode proposée plus robuste, on a utilisé la transformée en curvelettes pour un pré-filtrage d'images. Après le calcul des coefficients des curvelettes C_I de l'image d'entrée I , on leur applique un filtre non linéaire et enfin on calcule la transformée en

curvelettes inverse [14,73]. La relation entrée-sortie pour le filtre utilisé dans le domaine de la transformée en curvelettes est:

$$\tilde{C}_I = \begin{cases} C_I & \text{si } (|C_I| \geq T_1) \\ C_I \cdot \varphi(T, T_1) & \text{si } (|C_I| < T_1) \\ 0 & \text{si } (|C_I| < T) \end{cases} \quad (3.39)$$

avec $\varphi(T, T_1)$ est une fonction de pondération amortie de 1 à 0 dans un intervalle déterminé par les seuils T et T_1 . Ce filtre est robuste si les valeurs des seuils sont comprises dans l'intervalle $[0 \quad 4\sigma_I / (2\pi)^{1/2}]$, où σ_I^2 représente la variance de la variable en entrée C_I . Le meilleur affaiblissement du bruit, réalisé par ce filtre, est obtenu pour la valeur du seuil de $T_1 = 0.797\sigma_I$ [68].

Pour évaluer la qualité du filtrage, nous mesurons l'erreur quadratique moyenne (EQM) et le rapport signal sur bruit crête (PSNR en dB) entre chaque pixel de l'image \tilde{I} obtenue par le débruitage utilisant la TC et l'image d'entrée I [73] :

$$EQM = \frac{1}{N.M} \sum_{i=1}^N \sum_{j=1}^M (I(i, j) - \tilde{I}(i, j))^2 \quad (3.40)$$

$$PSNR(dB) = 10 \cdot \log_{10} \left[\frac{255^2}{EQM} \right] \quad (3.41)$$

où M et N étant les dimensions de l'image

3.3.4. Estimateur multi-échelles par la transformée en curvelettes

3.3.4.1. Principe de l'estimateur

Le principe de cette approche est similaire à celui de la méthode multi-échelles de Bouget [10] à l'exception de la pyramide gaussienne remplacée par la transformée en curvelettes. L'implantation de cette technique passe par trois étapes :

Première étape : Construction d'une pyramide d'images par la transformée en curvelettes à partir des images de la séquence. La décomposition en curvelettes de l'image donne plusieurs coefficients correspondant aux différents niveaux et différentes orientations. Ces coefficients représentent l'approximation a_j et les détails d_j avec $j = 1 : J$.

Deuxième étape : Estimation grossière du mouvement par la méthode de LK neuronale

Troisième étape : Estimation incrémentale multi-échelles du mouvement. On projette le flot optique estimé au niveau de résolution j à un niveau de résolution plus fin $j-1$.

La figure (3.5) illustre un schéma synoptique de la méthode proposée.

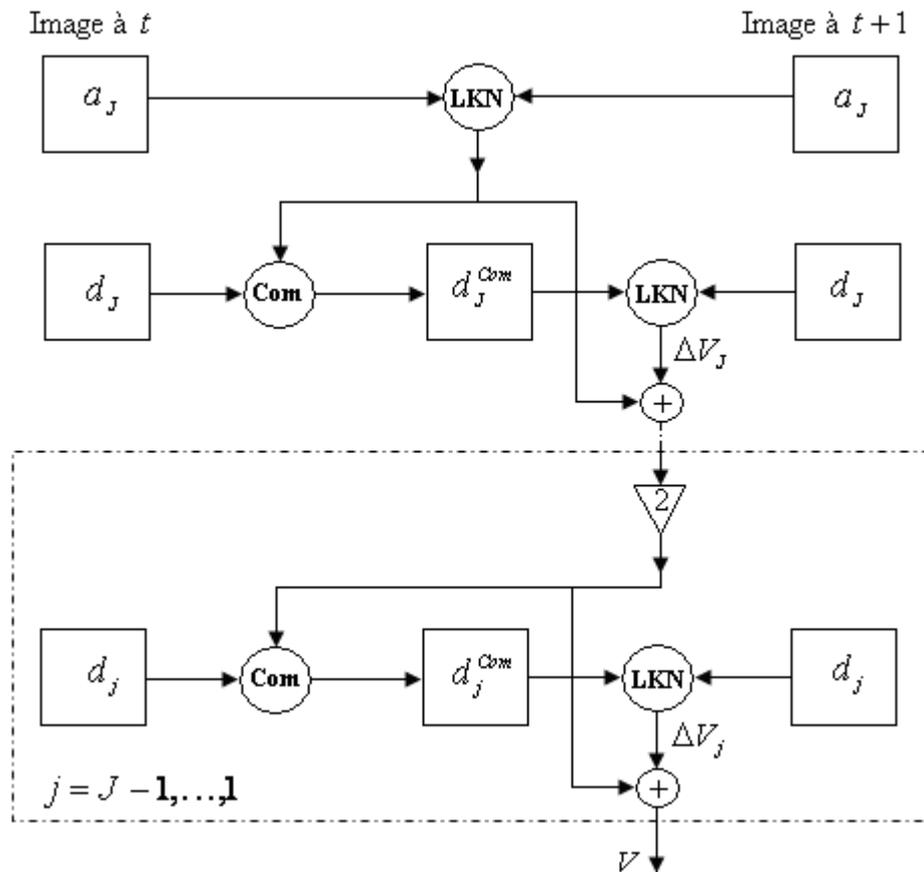


Figure 3.5. Schéma synoptique de l'estimateur multi-échelles proposé.

3.3.4.2. Algorithme final de la méthode proposée

En résumé, l'algorithme complet de la méthode proposée est organisé selon les étapes suivantes.

Début

Etape 1:

- Décomposer les deux frames de la séquence d'images par la transformée en curvelettes en J niveaux de résolutions (une sous-image d'approximations et des sous-images de détails selon l'orientation fréquentielle).

Etape 2 :

- Utiliser l'algorithme proposé dans la section (3.3.2) pour trouver une estimation grossière du vecteur du mouvement V_J .

Etape 3

Pour $k = J : -1 : 1$ faire :

- Compenser l'image par le vecteur du mouvement V_k .

- Utiliser le vecteur du mouvement calculé à la résolution k pour calculer le résidu Δv^k par l'algorithme LKN .
- $V^k \leftarrow V^k + \Delta v^k$
- Projeter V_k au niveau de résolution k à un niveau de résolution plus fin $k-1$

Fin pour

Fin

3.4. Résultats des simulations

3.4.1. Séquences de test et critères d'évaluation

3.4.1.1. Séquences de test

Afin de valider l'approche, nous avons testé notre algorithme sur des séquences vidéo artificielles et réelles. Plusieurs séquences artificielles dont le flot optique réel est connu sont souvent utilisées [4,74]. Elles permettent une analyse quantitative sur l'erreur d'estimation.

La séquence Yosemite, de taille 252×316 , représente le survol d'une chaîne montagneuse par un avion. L'amplitude du flot optique va de 0 pixel/image au point d'expansion à 5 pixels/image dans le coin inférieur gauche. Il existe une frontière de mouvement importante entre le ciel et les montagnes.

La séquences Arbre en divergence de taille 150×150 , a un mouvement divergent de 0 pixels/image au centre à 2.0 pixels/image à droite.

La séquences Arbre en translation de taille 150×150 , a un mouvement de translation de 1.7 pixels/image à gauche à 2.3 pixels/image à droite.

La séquence sinusoïdale de taille 100×100 a une translation de 1.58 dans le sens horizontal de 0.863 dans le sens vertical.

Quelques séquences fournies récemment par l'université de Middlebury : Hydrangea, Dimetrodon, Grover2 et Grover3.

3.4.1.2. Critères d'évaluation

Pour juger la qualité de l'estimateur du flot optique, nous avons utilisé l'erreur angulaire moyenne EAM entre le flot optique des séquences artificielles et le mouvement estimé [4]:

$$EAM = \frac{1}{N.M} \sum_{i=1}^N \sum_{j=1}^M \arccos\left(\frac{u_c u_e + v_c v_e + 1}{\sqrt{(u_c^2 + v_c^2 + 1)(u_e^2 + v_e^2 + 1)}}\right) \quad (3.42)$$

où $\mathbf{u}_c = (u_c, v_c)$ et $\mathbf{u}_e = (u_e, v_e)$ désignent le flot optique vrai et le champ de déplacement estimé respectivement. Elle permet une analyse quantitative sur l'erreur d'estimation. Cette mesure prend en compte à la fois l'erreur commise sur l'orientation et l'amplitude des vecteurs de vitesse pour chaque pixel de l'image. Traditionnellement, la moyenne et l'écart-type (ET) de l'erreur angulaire sur toute l'image caractérisent la précision de l'estimateur [4,6]. Récemment, un autre critère a été ajouté, c'est l'erreur moyenne absolue EPE (End Point Error) [74]:

$$EPE = \frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M \sqrt{(u_c^2 + v_c^2 + 1)(u_e^2 + v_e^2 + 1)} \quad (3.43)$$

3.4.2. Représentation du flot optique

De nombreuses solutions sont possibles pour représenter visuellement le flot optique calculé. La plus couramment utilisée est le tracé du champ des vecteurs vitesses superposé à l'image initiale, l'inconvénient étant que l'on ne visualise pas le champ dense, et donc que les points aberrants risquent d'être masqués ou au contraire mis en valeur, conduisant à une interprétation biaisée. La solution la plus intéressante est l'utilisation d'une carte de couleurs, comme montrée dans la figure 3.6 qui permet de représenter la direction du flot ainsi que son intensité de manière dense [74].

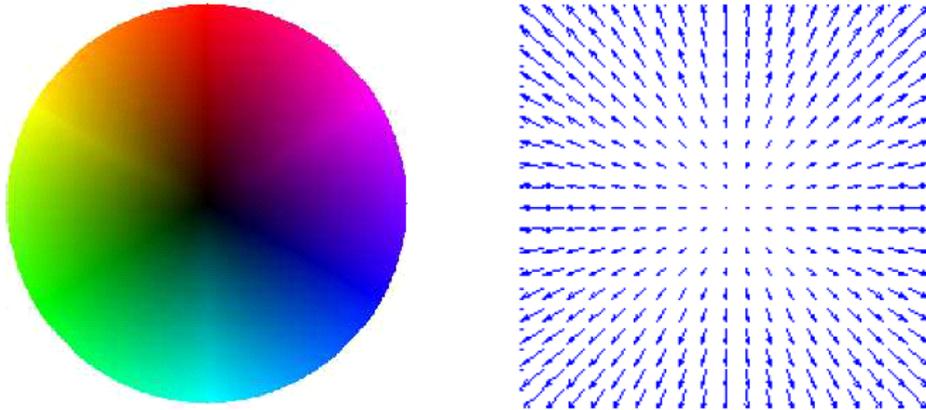


Figure 3.6. Colormap et affichage du même champ de vitesse sous forme de vecteurs.

Les vecteurs vitesses sont représentés par les couleurs contenues dans ce cercle. Chaque vecteur vitesse est codé par la couleur qu'il indique en plaçant son origine au centre du cercle. En fonction de l'angle, le point prend une couleur différente, et son intensité varie du noir à la couleur complète selon la norme du vecteur. Ce type de représentation permet une représentation dense (une couleur pour chaque pixel), et permet de visualiser très rapidement

la cohérence des résultats, la sensibilité au bruit de la méthode testée et l'aspect (lissage, discontinuités ...) du flot.

3.4.3. Sélection des paramètres

La méthode d'estimation nécessite trois paramètres à ajuster : le nombre de niveaux J , le nombre d'itérations dans chaque niveau et la taille de la fenêtre d'analyse. Ces paramètres sont choisis de façon à minimiser l'erreur angulaire moyenne EAM. Pour voir l'effet de ces paramètres sur la qualité des résultats, on a pris comme séquence de test 'Yosemite'. Dans le premier cas, nous fixons le nombre de niveaux à 1 et la taille de fenêtre à 15×15 . La figure (3.7.a) montre que quatre itérations sont suffisantes pour avoir une bonne performance, donc il est inutile d'augmenter le nombre d'itérations pour diminuer le temps d'exécutions. Pour l'effet de la taille de la fenêtre, on a fixé le nombre d'itérations à 4 et on a gardé le niveau 1. La taille optimale est entre 15×15 et 21×21 comme la montre la figure (3.7.b). Pour le nombre de niveaux, la figure (3.7.c) montre qu'il est inutile d'aller au-delà de deux niveaux de résolution.

Dans la suite de cette section, nous fixons le nombre de niveaux à 2, la taille de la fenêtre à 15×15 et le nombre d'itérations à 4.

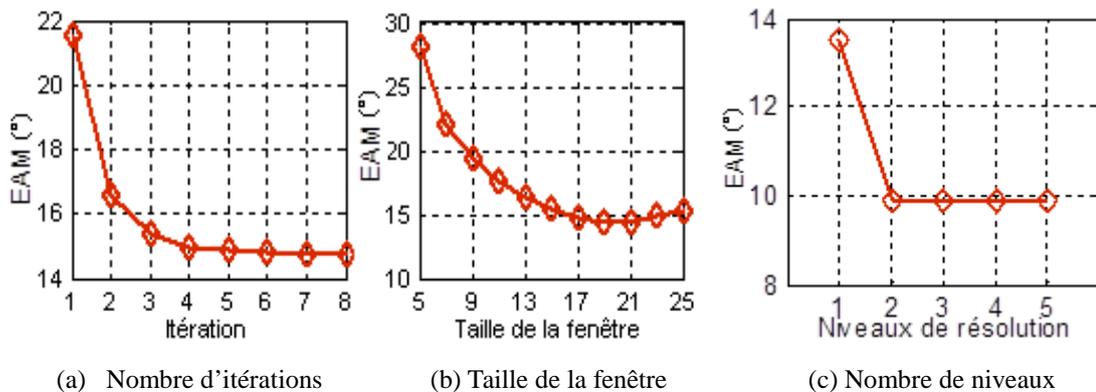


Figure 3.7. Influence des paramètres.

3.4.4. Stratégie 1 (masque d'interpolation)

La figure (3.8.a) représente un zoom avant de la séquence 'arbre en divergence' par rapport au centre de l'image, la figure (3.8.b) montre le flot optique réel et les figure (3.8.c) et 3.8.d représentent le module et la phase du vecteur de mouvement de la section représentée par la ligne diagonale sur la figure (3.8.b). Nous remarquons que le vecteur de mouvement estimé par la méthode proposée suit le vecteur réel du point de vue module ou direction.

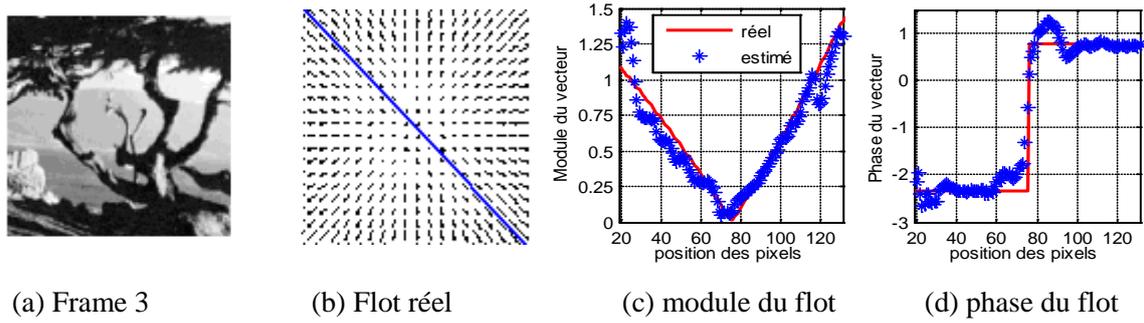


Figure 3.8. Résultats d’une section diagonale de la séquence ‘arbre divergente’.

Pour la séquence Yosemite, la figure (3.9) montre que la méthode proposée estime bien le flot à l’exception du mouvement importante dans la zone frontière entre le ciel et les montagnes (EAM=9.8851° et ET= 14.3368°).

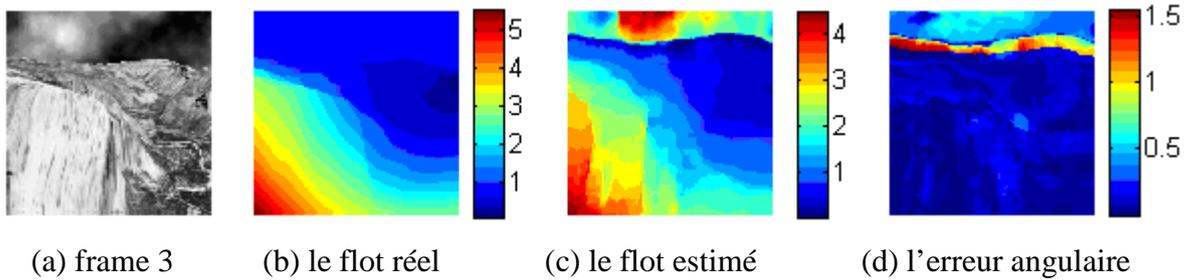


Figure 3.9. Performance de la méthode proposée pour la séquence ‘Yosemite’.

La méthode proposée a été aussi appliquée sur les autres séquences d’images synthétiques pour plusieurs niveaux de résolution, le tableau (3.1) résume les résultats obtenus. Les figures (3.10), (3.11) et (3.12) illustrent le flot réel et le meilleur flot estimé de chaque séquence à partir du tableau (3.1), nous remarquons que le flot estimé et le flot réel ont la même allure. Ces résultats sont comparés avec ceux obtenus avec la méthode originale et sont résumés dans le tableau (3.2). On voit que la méthode modifiée est plus avantageuse de point de vu performance et rapidité. Concernant le temps de calcul, les deux méthodes ont été simulées sous Matlab 7.7.0 par un PC 3GHz et 1Go de RAM.

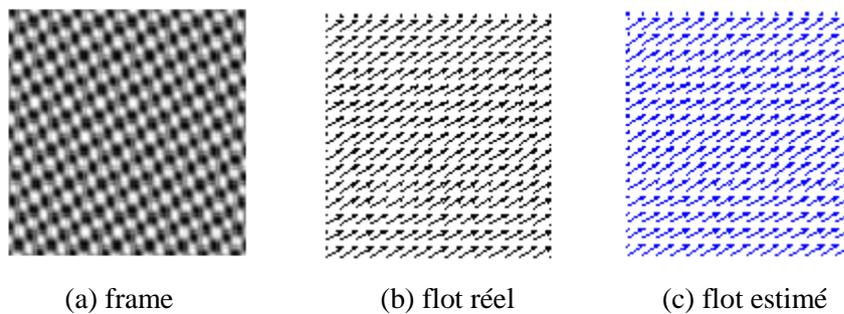


Figure 3.10. Performance de la méthode pour la séquence ‘Sinusoïde’ (EAM=0.5661° et ET= 0.0239°).

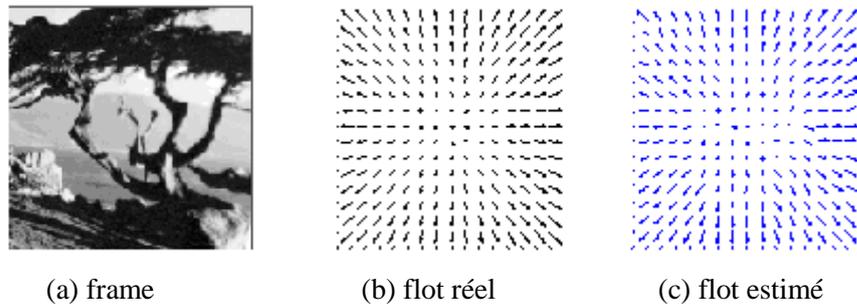


Figure 3.11. Performance de la méthode pour la séquence ‘arbre divergent’ (EAM=6.7731° et ET= 3.2573°).

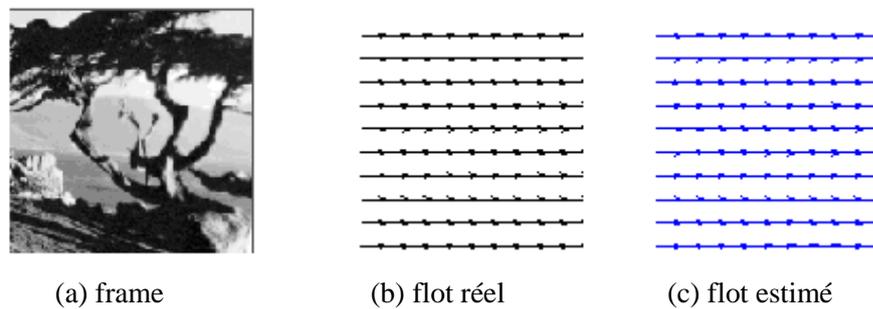


Figure 3.12. Performance de la méthode pour la séquence ‘arbre en translation’ (EAM=0.9668° et ET= 0.2287°).

Séquences	Sinusoïdale		Yosemite		Arbre en translation		Arbre divergente	
	EAM (°)	ET (°)	EAM (°)	ET (°)	EAM (°)	ET (°)	EAM (°)	ET (°)
Niveau 1	0.5661	0.0239	13.5145	19.6653	1.9466	6.6968	6.7731	3.2573
Niveau 2	0.5661	0.0239	9.8851	14.3368	0.9668	0.2287	6.7731	3.2573
Niveau 3	49.5678	45.2690	9.8810	14.3310	0.9668	0.2287	6.7731	3.2573

Tableau 3.1. Performance de la méthode modifiée.

	M	originale		modifiée	
	éthode	Temps (s)	EAM (°)	Temps(s)	EAM (°)
Sinusoïdale	1	3.34	1.42	3.01	0.5661
	2	4.21	1.52	3.73	0.5661
	3	4.43	54.57	4.14	9.5678
Yosemite	1	20.59	16.17	19.20	13.5145
	2	25.57	13.48	23.59	9.8851
	3	26.48	15.48	25.62	9.8810

Tableau 3.2. Comparaison de la performance entre la méthode originale et la méthode modifiée.

3.4.5. Stratégie 2 (Réseaux de neurones)

La méthode proposée utilisant le réseau de Hopfield (HNN) ou le réseau de Zhang (DZNN) a été testée sur les séquences Yosemite et arbre en translation. La convergence des deux réseaux est montrée sur les deux figures (3.13) et (3.14). Nous observons que le réseau de Hopfield nécessite des dizaines de milliers d'itérations pour la convergence. Cependant, deux ou trois itérations sont suffisantes pour le réseau de Zhang. L'erreur de convergence du réseau de neurones est donné par : $\|\varepsilon(k)\|_2 = \|\tilde{G}V(k) - \tilde{d}\|_2$

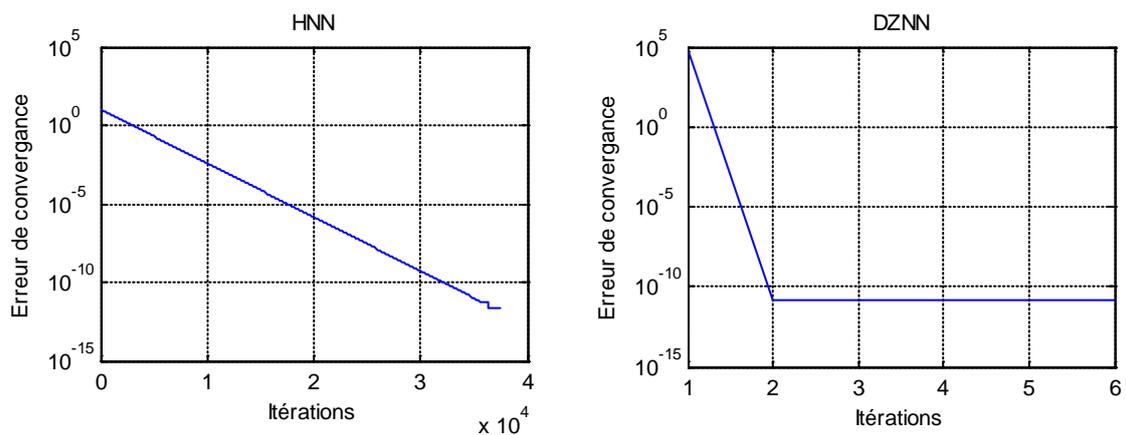


Figure 3.13. Courbe de convergence pour la séquence 'Yosemite'.

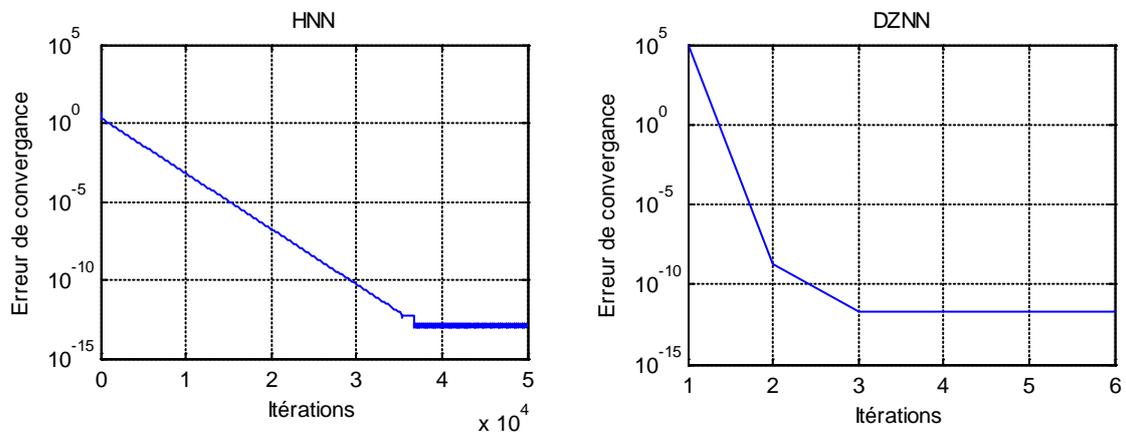


Figure 3.14. Courbe de convergence pour la séquence 'arbre en translation'.

Le flot estimé n'est pas représenté dans cette section puisqu'il est similaire à celui vu dans la stratégie 1.

Pour mieux évaluer la méthode proposée, elle est comparée avec des travaux récents qui utilisent la séquence 'Yosemite' avec et sans nuages et la séquence 'arbre en translation'. Nous avons comparé les résultats obtenus avec la méthode de Srinivasan *et al.* [75], la méthode de Arredondo *et al.* [76], la méthode de Liu *et al.* [77] et quelques méthodes rapides obtenues à partir du site 'Middlebury optical flow evaluation'

<http://vision.middlebury.edu/flow/eval/>. Le tableau 3.3 donne les résultats obtenus sur les deux séquences artificielles ‘Yosemite avec et sans nuages’, et ‘arbre en translation’. Les performances de notre algorithme sont comparables à celles citées dans cette section.

Séquence	Méthode	EAM (°)	ET (°)
Yosemite avec nuages	Proposée	9.88	14.33
	Srinivasan <i>et al.</i> [75]	8.94	10.63
	Liu <i>et al.</i> [77].	13.18	13.24
	Arredondo <i>et al.</i> [76]	12.87	15.87
Yosemite sans nuages	Proposée	3.21	3.15
	Cassisa <i>et al.</i> [78]	2.97	2.88
	Lempitsky <i>et al.</i> [81]	4.55	4.37
	Werlberger <i>et al.</i> [79]	2.83	2.09
	Rhemann <i>et al.</i> [80]	4.16	3.86
Arbre en translation	Proposée	0.96	0.22
	Srinivasan <i>et al.</i> [75]	0.61	0.26
	Liu <i>et al.</i> [77].	1.34	1.11
	Arredondo <i>et al.</i> [76]	4.24	7.29

Tableau 3.3. Comparaison quantitative avec d’autres méthodes.

3.4.6. Stratégie 3 (débruitage par TC)

Dans cette section on a évalué les performances de la méthode proposée pour des séquences d’images auxquelles on ajoute un bruit de 10%.

La figure (3.15) présente une décomposition en curvelettes d’une frame de la séquence Hydrangea sans et avec bruit. Alors que la figure (3.16), elle montre les résultats de débruitage du frame bruitée utilisant la transformée en ondelettes et la transformée en curvelettes. On remarque que le filtrage par la TC fournit un meilleur $PSNR = 35.25dB$ par rapport au filtrage par TO $PSNR = 33.61dB$.

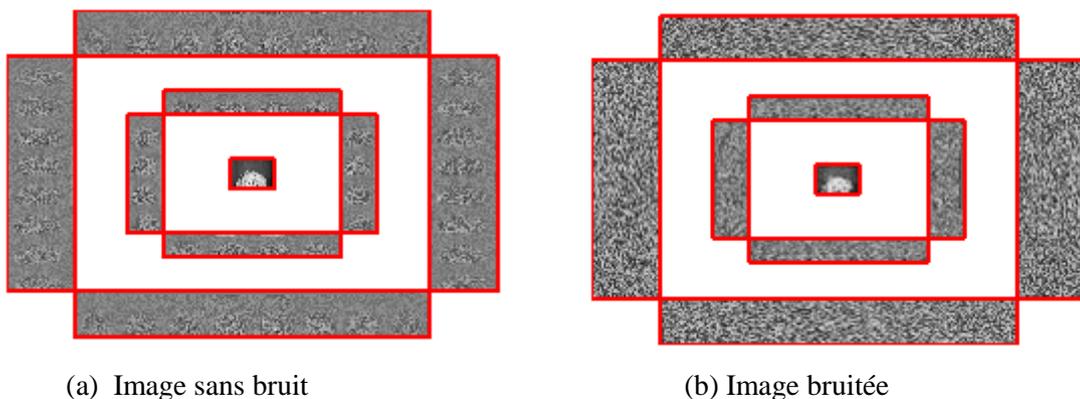
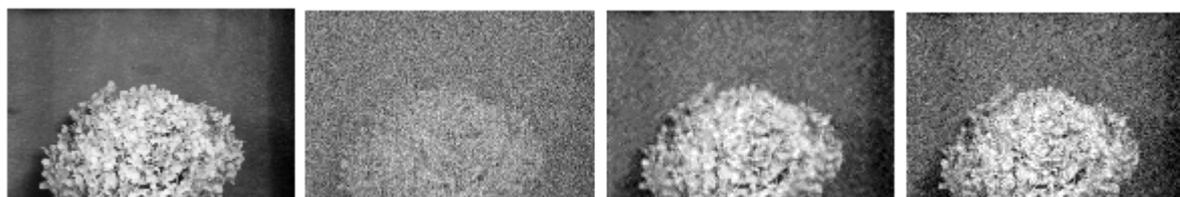


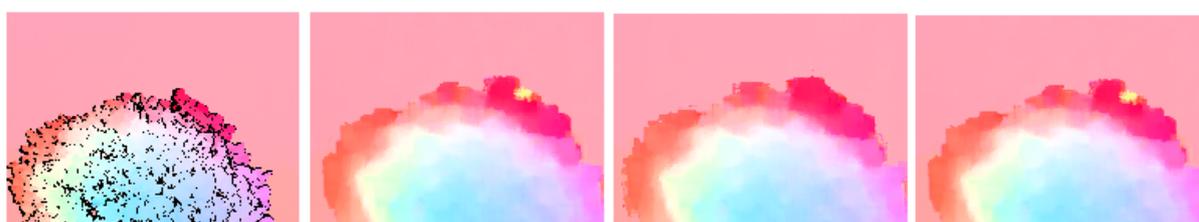
Figure 3.15. Décomposition en curvelettes de l’image Hydrangea en 2 niveaux.



(a) image originale (b) image bruitée (c) image débruitée (TC) (d) image débruitée (TO)

Figure 3.16. Débruitage de l'image 'Hydrangea'.

La figure (3.17) montre le flot optique estimé de la séquence sans filtrage, avec filtrage par la TC et avec filtrage par la TO. On constate que les flots estimés de la séquence sans filtrage et avec filtrage par la TO sont presque similaires, alors que le flot optique estimé avec filtrage par la TC est meilleur pour la séquence bruitée et non bruitée. La figure (3.18) met en évidence ce résultat.



(a) Flot réel (b) Flot estimé (c) Flot estimé par TC (f) Flot estimé par TO

Figure 3.17. Résultat du flot optique de la séquence 'Hydrangea' sans bruit.

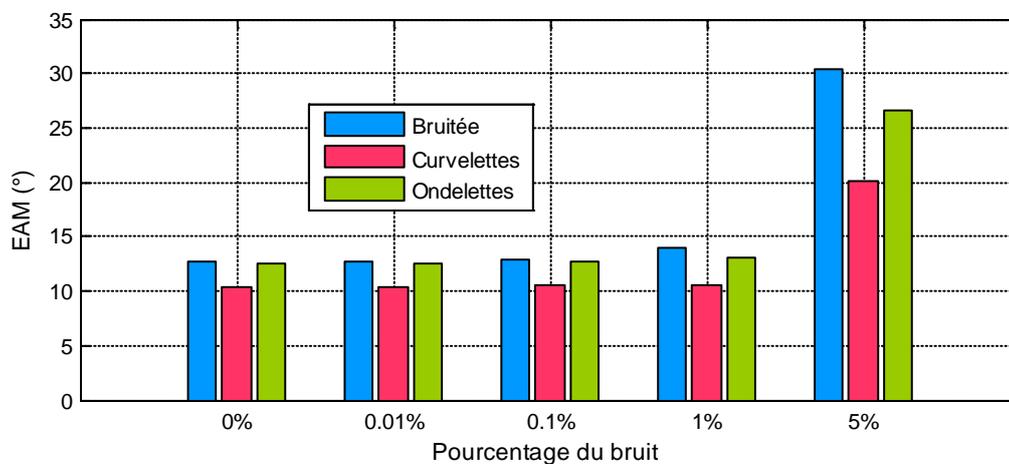


Figure 3.18. Erreur angulaire de la séquence 'Hydrangea'.

Dans le deuxième test, nous répétons la même procédure, mais avec la séquence Yosemite dont la version bruitée et le flot réel sont représentés dans (3.19).

La figure (3.20) montre le flot estimé de la séquence sans filtrage, avec filtrage par TC et avec filtrage par TO de la séquence sans bruit.



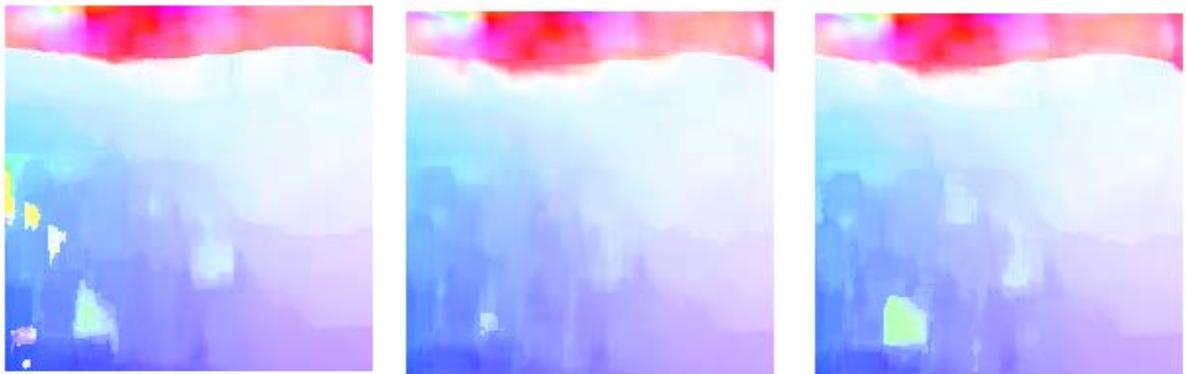
(a) Frame 11 (b) Frame bruité (c) Flot réel

Figure 3.19. Frame de la séquence ‘Yosemite’ et son flot optique réel.



(d) Flot estimé sans débruitage (e) Flot estimé par TC (f) Flot estimé par TO

Figure 3.20. Flot optique estimé de la séquence ‘Yosemite’ sans bruit.



(d) Flot estimé sans débruitage (e) Flot estimé par TC (f) Flot estimé par TO

Figure 3.21. Flot optique estimé de la séquence ‘Yosemite’ bruitée.

La figure (3.21) montre le flot estimé de la séquence sans filtrage, avec filtrage par curvelettes et par ondelettes de la séquence bruitée (ajout d’un bruit de 1%). Dans les deux tests on remarque que le flot estimé après un prétraitement par la transformée en curvelettes est meilleur que le flot estimé sans prétraitement aussi bien qu’après un prétraitement par la transformée en ondelettes. La figure (3.22) présente l’erreur angulaire moyenne pour différents bruits pour les séquences Yosemite. Elle prouve l’efficacité du prétraitement par la transformée en curvelettes pour l’estimation du mouvement.

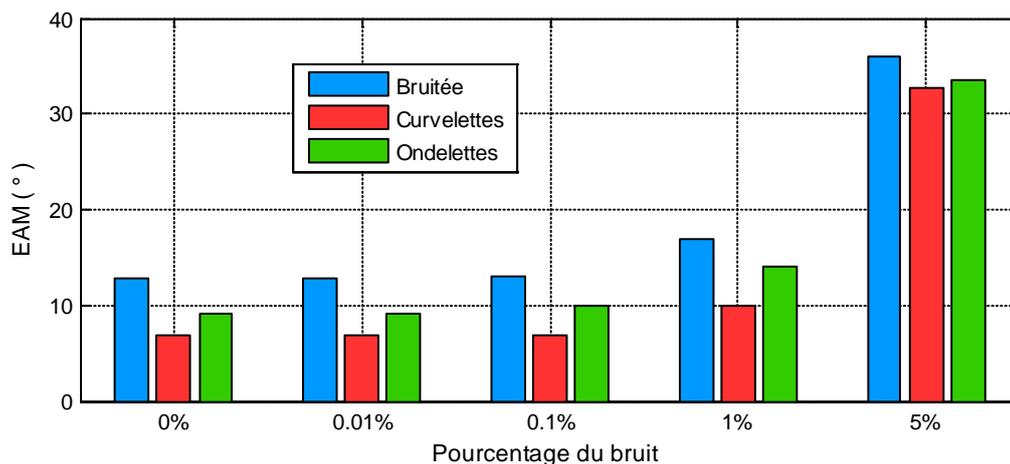


Figure 3.22. Erreur angulaire de la séquence 'Yosemite'.

3.4.7. Stratégie 4 (approche multi-résolutions)

La méthode d'estimation nécessite trois paramètres à ajuster : le nombre de niveaux J , le nombre d'itérations dans chaque niveau et la taille de la fenêtre d'analyse. Ces paramètres sont choisis de façon à minimiser l'erreur angulaire moyenne EAM. Pour voir l'effet du nombre de niveaux de résolution; on a fixé le nombre d'itérations à 3 et la taille de fenêtre à 15×15 . Les figures (3.23 et 3.24) montrent l'erreur angulaire et l'erreur moyenne absolue pour différents niveaux de résolution et pour différentes séquences synthétiques. On remarque que pour la séquence Dimentrodon et Grover 2, trois niveaux sont suffisants pour avoir une bonne performance. Cependant les séquences Hydrangea et Grover 3, elles nécessitent quatre. Donc il est inutile d'augmenter le nombre d'itérations pour diminuer le temps d'exécutions.

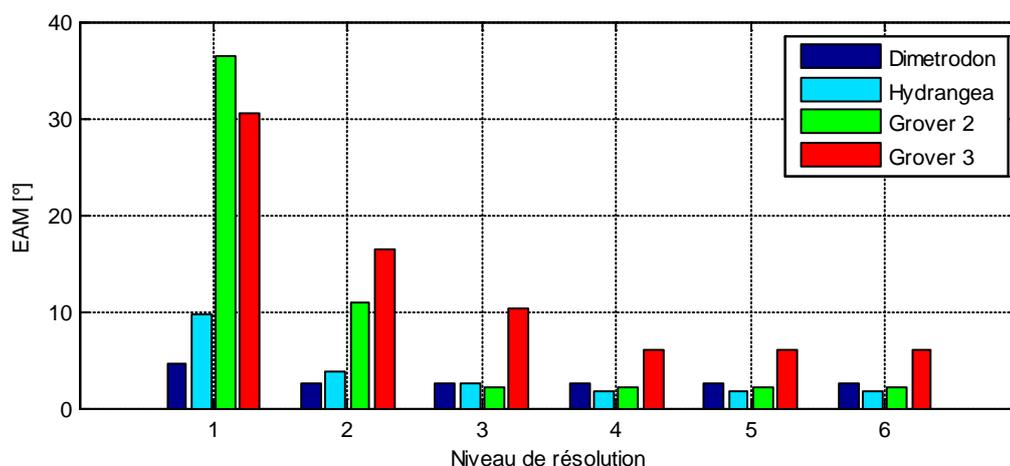


Figure 3.23. Erreur angulaire pour différents niveaux de résolution.

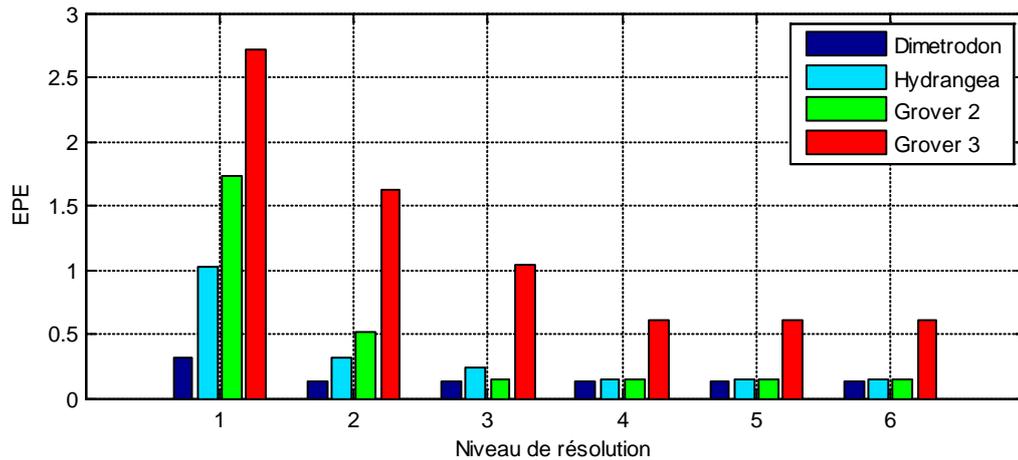


Figure 3.24. Erreur absolue pour différents niveaux de résolution.

3.4.7.1. Comparaison quantitative avec d'autres algorithmes

On a comparé les résultats obtenus avec la méthode multi-échelles proposée par Flopis et Sachez [82]. Le tableau (4.8) résume l'erreur angulaire moyenne et l'erreur absolue moyenne sur des séquences synthétiques de la base de Middlebury. On constate une nette amélioration de la méthode proposée due à l'utilisation de la transformée en curvelettes.

Séquence	Dimetrodon		Hydrangea		Grover 2		Grover 3	
	EPE	AAE [°]	EPE	AAE [°]	EPE	AAE [°]	EPE	AAE [°]
Flopis et Sachez[82]	0.150	2.767	0.327	3.364	0.219	3.105	0.842	7.608
Méthode proposée	0.128	2.555	0.24	2.636	0.14	2.042	0.615	5.963

Tableau 3.4. Comparaison quantitative avec la méthode de Flopis et Sachez.

3.4.7.2. Applications aux séquences réelles

Comme on a vu précédemment, la méthode proposée présente une supériorité pour les séquences synthétiques. Dans cette section, on teste son efficacité pour les séquences réelles.

La séquence Taxi de Hambourg : cette séquence montrée dans la figure (3.25.a) représente la surveillance d'un carrefour par une caméra statique placée en hauteur. Cette séquence contient 4 objets en déplacement : un taxi arrivant de la droite et tournant vers sa droite, un van arrivant de la droite et se dirigeant vers la gauche (ce véhicule est légèrement masqué par la présence d'un arbre), une voiture arrivant de la gauche et se dirigeant vers la droite, et un piéton difficilement discernable en haut à gauche. Les vitesses de ces objets dans le repère image sont respectivement de 1.0, 3.0, 3.0 et 0.3 pixels par image. Le point d'observation de la scène étant statique, les mouvements apparents sur cette séquence ne sont pas modifiés par le mouvement propre du capteur puisqu'il est nul.

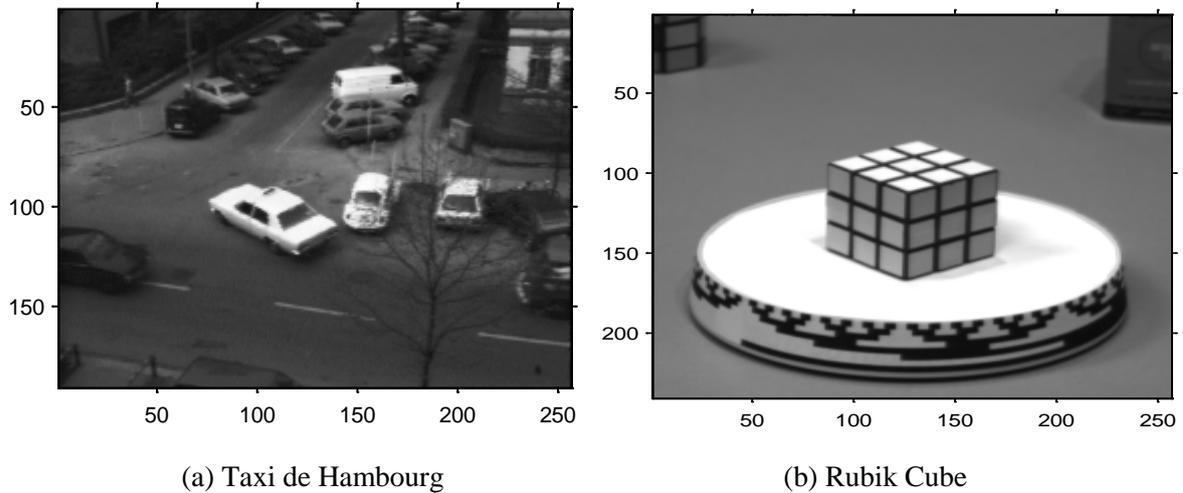


Figure 3.25. Séquences réelles de test.

La séquence Rubik Cube : La figure (3.25.b) montre une frame de la séquence Rubik Cube. Elle représente un cube placé au centre d'un plateau circulaire, lequel tourne sur lui-même à une vitesse constante dans le sens antihoraire. Les vitesses de déplacement sont comprises entre 1.2 et 1.4 pixels par image pour le plateau tournant, et comprises entre 0.2 et 0.5 pour le cube. Dans ce cas aussi, le point d'observation est statique, il n'y aura donc aucune modification des mouvements apparents induite par le capteur.

Les figures (3.26) et (3.27) montrent le flot optique estimé par la méthode de Lucas & Kanade multi-échelles (LKM) et la méthode multi-échelles par la transformée en curvelettes pour les deux séquences réelles. On remarque que pour la séquence, 'Taxi de Hambourg', le flot optique estimé par les deux méthodes fait apparaître les trois véhicules, alors que le mouvement du piéton n'a pas été mesuré. De plus, la méthode proposée détecte et préserve bien les contours des objets en mouvement. La même remarque pour la séquence Rubik Cube.

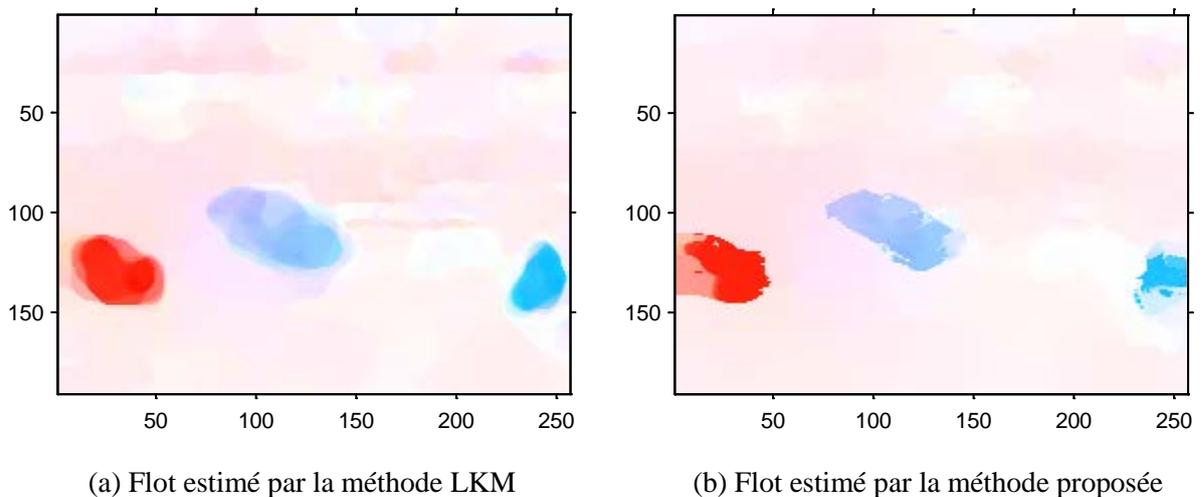
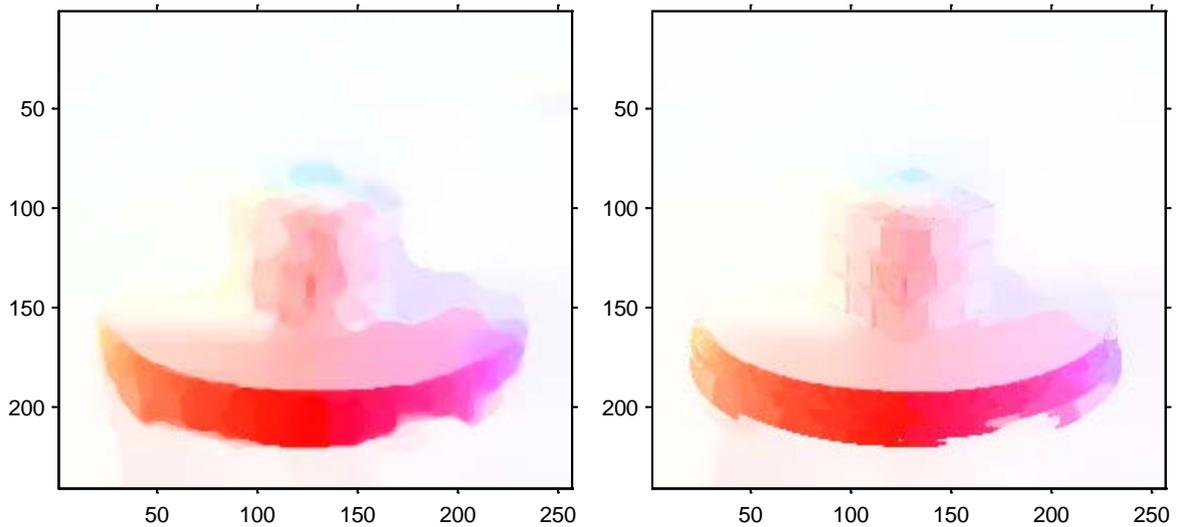


Figure 3.26. Flot estimé de la séquence 'Taxi de Humbourg'.



(a) Flot estimé par la méthode LKM

(b) Flot estimé par la méthode proposée

Figure 3.27. Flot estimé de la séquence 'Rubic Cube'.

3.5. Conclusion

Pour le problème d'estimation de mouvement dans une séquence d'images, deux performances critiques sont à améliorer, le temps de calcul et la précision des résultats. La méthode itérative multi-échelles de Lucas & Kanade est l'une des meilleures méthodes dans ce domaine. Dans ce chapitre nous avons introduit deux outils pour l'estimation de mouvement. Pour améliorer la performance de la méthode multi-échelles de Bougier, nous avons proposé d'utiliser la transformée en ondelettes géométriques soit pour le débruitage des images de la séquence, soit pour l'estimation multi-échelles de la méthode de LK. Pour accélérer l'algorithme du calcul du flot optique, deux réseaux de neurones (réseau de Hopfield et le réseau de Zhang) ont été utilisés. Les résultats de simulation pour les séquences synthétiques ou réelles montrent que les méthodes proposées apportent une nette amélioration de la méthode itérative multi-échelles de Lucas & Kanade.

Chapitre 4

Implémentation parallèle de la méthode de Horn & Schunk

4.1. Introduction

Motivés par la rapidité et la robustesse du réseau de Zhang [83-88], nous proposons d'utiliser une version discrète de ce réseau pour le calcul rapide du vecteur de mouvement au lieu de la méthode originale de Horn & Schunk. L'approche proposée consiste à décrire 3 types de réseaux discrets à partir de la version continue. Les deux premières versions dont la dynamique du réseau est de forme explicite et implicite servent à inverser la matrice contenant les dérivées spatio-temporelles. Le flot optique peut être également estimé directement en résolvant le système d'équations. Ceci fait l'objet de la troisième version du réseau proposée.

4.2. Méthode de Horn & Schunk

La méthode différentielle de Horn & Schunk est basée sur la minimisation d'une fonction d'énergie sur le domaine Ω [3]:

$$E = \int_{\Omega} \left((I_t + uI_x + vI_y)^2 + \lambda^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right) dx dy \quad (4.1)$$

où I_x , I_y et I_t sont les dérivés partielles de l'intensité $I(x, y, t)$ d'une image de taille N par rapport à x , y et t respectivement, u, v sont les composantes du vecteur de mouvement, $\nabla u, \nabla v$ leurs dérivées spatiales et λ est un paramètre de pondération. La solution de l'équation (4.1) peut être élaborée en utilisant les équations de Euler Lagrange :

$$\begin{cases} I_x^2 u + I_x I_y v + I_x I_t = \lambda^2 \nabla^2 u \\ I_x I_y u + I_y^2 v + I_y I_t = \lambda^2 \nabla^2 v \end{cases} \quad (4.2)$$

Par une discrétisation approximative, il s'en suit que pour tous les pixels d'indices $i \in \{1, 2, \dots, N = n \times m\}$ [19]:

$$\begin{cases} I_{xi}^2 u_i + I_{xi} I_{yi} v_i + I_{xi} I_{ti} - \lambda^2 \sum_{j \in N_i} (u_j - u_i) = 0 \\ I_{xi} I_{yi} u_i + I_{yi}^2 v_i + I_{yi} I_{ti} - \lambda^2 \sum_{j \in N_i} (v_j - v_i) = 0 \end{cases} \quad (4.3)$$

où $u_i, v_i, I_{xi}, I_{yi}, I_{ti}$ sont les valeurs de u, v, I_x, I_y, I_t au point i respectivement, et N_i est l'ensemble des indices des points voisins de i . L'expression (4.3) peut être réécrite sous la forme [19]:

$$\begin{cases} (I_{xi}^2 + \lambda^2 c_i) u_i + I_{xi} I_{yi} v_i - \lambda^2 \sum_{j \in N_i} u_j = -I_{xi} I_{ti} \\ I_{xi} I_{yi} u_i + (I_{yi}^2 + \lambda^2 c_i) v_i - \lambda^2 \sum_{j \in N_i} v_j = -I_{yi} I_{ti} \end{cases} \quad (4.4)$$

où $c_i = \text{card}(N_i)$ est le nombre des points voisins du pixel i . L'équation (4.4) peut être également réécrite sous la forme matricielle:

$$Az = b \quad (4.5)$$

où z, b sont deux vecteurs réels de longueur $2N$ ayant coordonnées:

$$\begin{cases} z_{2i-1} = u_i \\ z_{2i} = v_i \end{cases} \quad (4.6)$$

$$\begin{cases} b_{2i-1} = -I_{xi} I_{ti} \\ b_{2i} = -I_{yi} I_{ti} \end{cases} \quad (4.7)$$

avec $i \in \{1, \dots, N\}$

A est une matrice réelle de taille $2N \times 2N$, symétrique et définie positive dont les éléments:

$$\begin{cases} A_{2i-1, 2i-1} = I_{xi}^2 + \lambda^2 c_i \\ A_{2i, 2i} = I_{yi}^2 + \lambda^2 c_i \\ A_{2i-1, 2i} = A_{2i, 2i-1} = I_{xi} I_{yi} \\ A_{2i-1, 2j-1} = A_{2i, 2j} = -\lambda^2, \quad j \in N_i \end{cases} \quad \text{pour } i \in \{1, \dots, N\} \quad (4.8)$$

tous les autres éléments sont égaux à zéro.

Le système d'équations linéaires (4.5) peut être résolu quand A est non singulière comme :

$$z = A^{-1}b \quad (4.9)$$

Le système (4.5) peut être aussi résolu itérativement en utilisant la méthode de Jacobi [19]:

$$z^{k+1} = Pz^k + d \quad (4.10)$$

où d est un vecteur réel avec coordonnées :

$$\begin{cases} d_{2i-1} = -I_{xi}I_{ii}/(I_{xi}^2 + I_{yi}^2 + \lambda^2 c_i) \\ d_{2i} = -I_{yi}I_{ii}/(I_{xi}^2 + I_{yi}^2 + \lambda^2 c_i) \end{cases}, i \in \{1, 2, \dots, N\} \quad (4.11)$$

P est une matrice de Jacobi avec éléments :

$$\begin{cases} P_{2i,2i} = P_{2i-1,2i-1} = 0 \\ P_{2i,2j} = (I_{xi}^2 + \lambda^2 c_i)/c_i (I_{xi}^2 + I_{yi}^2) + \lambda^2 c_i^2 \\ P_{2i-1,2j} = P_{2i,2j-1} = -(I_{xi} \cdot I_{yi})/c_i (I_{xi}^2 + I_{yi}^2) + \lambda^2 c_i^2 \\ P_{2i-1,2j-1} = (I_{yi}^2 + \lambda^2 c_i)/c_i (I_{xi}^2 + I_{yi}^2) + \lambda^2 c_i^2 \end{cases} \forall i, j \in \{1, \dots, N\}, \text{ et } j \in N_i \quad (4.12)$$

Tous les autres éléments sont nuls.

Dans ce qui suit, nous proposons d'utiliser le réseau de Zhang pour résoudre le système (4.5). Deux stratégies seront utilisées, la première consiste à inverser la matrice A , alors que la deuxième, elle consiste à résoudre directement le système [89-90].

4.3. Inversion de la matrice A par le réseau de Zhang

Récemment Y. Zhang proposait une variété de réseaux de neurones pour inverser une matrice quelconques (à éléments fixes ou variable dans le temps). Suivant la forme explicite ou implicite de la dynamique du réseau, deux modèles du réseau peuvent être considérés le réseau DZNN1-1 et le réseau DZNN1-2.

4.3.1. Réseau de neurones à dynamique explicite

4.3.1.1. Principe

Soit $A \in \mathbb{R}^{L \times L}$ où $L = 2N$ est une matrice à inverser, le réseau de Zhang est conçu de façon à résoudre l'équation suivante [83-87] :

$$AX(t) - I = 0 \quad (4.13)$$

où $X(t)$ représente l'inverse de la matrice instantanée et $I \in \mathbb{R}^{L \times L}$ dénote la matrice identité.

La matrice $X(t)$ converge vers la matrice théorique A^{-1} quand le réseau atteint un état stable.

En prenant la dérivée de la matrice-erreur $\varepsilon(t) = AX(t) - I$ à minimiser :

$$\frac{d\varepsilon(t)}{dt} = -\Gamma f(\varepsilon(t)) \quad (4.14)$$

où $\Gamma \in \mathbb{R}^{L \times L}$ est une matrice des paramètres utilisée pour régler la cadence de la convergence du réseau et $f(\cdot) : \mathbb{R}^{L \times L} \rightarrow \mathbb{R}^{L \times L}$ dénote la matrice-fonction d'activation.

Dans l'expression (4.13), les matrices A et I sont fixes, d'où :

$$\dot{\boldsymbol{\varepsilon}}(t) = A\dot{X}(t) \quad (4.15)$$

En remplaçant cette expression ainsi que l'expression de $\boldsymbol{\varepsilon}(t)$ dans (4.14) nous obtenons :

$$A\dot{X}(t) = -\Gamma f(AX(t) - I) \quad (4.16)$$

Pour assurer que tous les éléments de la matrice $X(t)$ converge avec la même cadence et en même temps, il suffit que les valeurs propres de la matrice Γ soient proches. Pour la simplification nous prenons $\Gamma = \gamma \cdot I$ avec $\gamma > 0$. Cette proposition nous conduit à la formule:

$$A\dot{X}(t) = -\gamma \cdot f(AX(t) - I) \quad (4.17)$$

4.3.1.2. Discrétisation du réseau

La version discrète du réseau de Zhang est obtenue en utilisant la méthode d'Euler :

$$\dot{X}(t) = (X(k+1) - X(k)) / h \quad (4.18)$$

où h dénote le pas d'échantillonnage, $X(k)$ est le $k^{\text{ème}}$ échantillon de $X(t)$ pris à l'instant $t = kh$. Le modèle discret du réseau de Zhang est donné par [16]:

$$AX(k+1) = AX(k) - \tau f(AX(k) - I) \quad (4.19)$$

avec $\tau = \gamma \cdot h > 0$.

En posant l'hypothèse que la matrice A^{-1} existe, le modèle discret de Zhang (DZNN1-1) peut être réécrit sous la forme :

$$X(k+1) = X(k) - \tau A^{-1} f(AX(k) - I) \quad (4.20)$$

Comme A^{-1} est inconnue, et $X(k)$ peut être très proche de A^{-1} après quelque itérations, dans ces conditions A^{-1} dans l'équation (4.20) peut être remplacée par $X(k)$. Par conséquent, nous pouvons réécrire l'équation (4.20) comme suit [16]:

$$X(k+1) = X(k) - \tau X(k) f(AX(k) - I) \quad (4.21)$$

4.3.1.3. Implémentation du réseau DZNN1-1

L'équation (4.21) nous permet d'implémenter aisément le modèle discret du réseau de Zhang (DZNN1-1) représenté sur la figure (4.1).

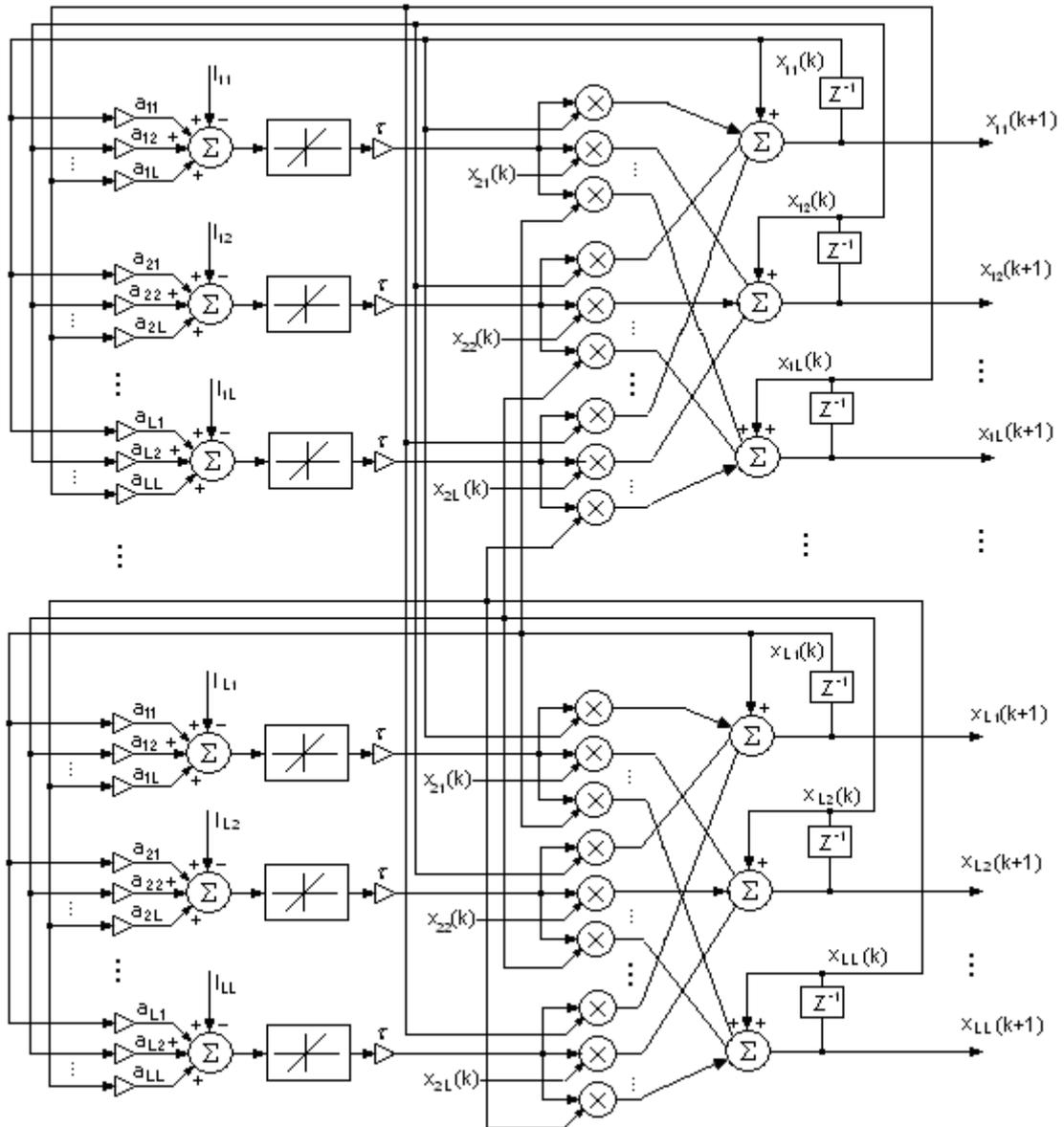


Figure 4.1. Architecture du réseau DZNN1-1

Nous remarquons que le réseau RNZD1-1 est composé de L sous-réseaux similaires dont chacun est composé L neurones [88]. Si nous définissons la complexité du réseau comme le nombre d'opérateurs utilisés, nous constatons que le réseau contient $2L^3 + L^2$ multiplicateurs, $2L^2$ additionneurs, L^2 opérateurs de retard et L^2 fonctions d'activation. Comme tous les sous blocs opèrent simultanément, donc le temps de réponse du réseau est celui du sous réseau. De plus, les neurones eux aussi sont parallèles, ceci conduit à dire que le temps de réponse du réseau est égal au temps de propagation de l'entrée vers la sortie à travers une fonction d'activation, $2L$ multiplications et 2 additions de L entrées.

4.3.2. Réseau de neurones à dynamique implicite

4.3.2.1. Principe

Dans cette section, l'inversion de la matrice X est équivalent à la solution d'un système de L équations. Si nous désignons $x^i = [x_1^i, x_2^i, \dots, x_L^i]^T$ le $i^{\text{ème}}$ colonne de la matrice X et I^i le $i^{\text{ème}}$ colonne de la matrice I , l'équation (4.13) peut être réécrite sous la forme [84] :

$$Ax^i = I^i \quad i = 1, \dots, L \quad (4.22)$$

Nous remarquons que l'équation (4.22) est similaire à l'équation (3.34). Donc nous pouvons déduire que le réseau de Zhang pour l'inversion de la matrice est composé de L sous-réseaux similaires dont chacun à pour rôle de calculer une seule colonne de la matrice inverse cherchée. Chaque sous-réseau est basé sur la solution de l'équation $Ax^i - I^i = 0$.

En suivant les mêmes démarches que dans la section (3.3), la dynamique de chaque réseau est régie par :

$$A\dot{x}^i(t) = -\mathcal{F}(Ax^i(t) - I^i) \quad (4.23)$$

4.3.2.2. Implémentation du réseau DZNN1-2

A partir de l'équation (4.23), nous pouvons exprimer l'équation dynamique du réseau sous la forme implicite [89] :

$$\dot{x}_j^i(t) = -\mathcal{F}\left(\sum_{k=1}^L a_{jk} x_k^i(t) - I_j^i\right) - \left(\sum_{k=1, k \neq j}^L a_{jk} \dot{x}_k^i(t)\right) + (1 - a_{jj}) \dot{x}_j^i(t) \quad (4.24)$$

où $x_j^i(t) = x(j, i)$ désigne le $j^{\text{ème}}$ élément de la $i^{\text{ème}}$ colonne de la matrice $X(t)$, $I_j^i = \delta(j, i)$ désigne le $j^{\text{ème}}$ élément de la $i^{\text{ème}}$ colonne de la matrice I .

La version discrète du réseau de Zhang (DZNN1-2) est obtenue en remplaçant $\dot{x}(t)$ de l'équation (4.24) par $(x(k+1)h) - x(kh) / h$.

L'architecture du réseau est illustrée dans la figure (4.2). Nous remarquons que le réseau est composé de L sous-réseaux similaires contenant des sommateurs, des opérateurs de retard, et des coefficients de pondération. Chaque sous-réseau doit performer $2L^2 + L$ multiplications, $2L$ additions de L nombres et $2L$ additions de 2 nombres par itération. La totalité des multiplications sont évaluées en parallèle nécessitant $2L^2 + L$ processeurs. Le calcul des $2L$ additions est fait sur $2L(L-1)$ sommateurs en $O(\log(L))$ cycle d'horloge [89] et les $2L$ dernières additions nécessitent $2L$ sommateur et un cycle d'horloge. Puisque tous les

sous réseaux opèrent simultanément et indépendamment, la complexité du réseau sera multipliée par L , tandis que le temps de calcul reste similaire à celui du sous-réseau.

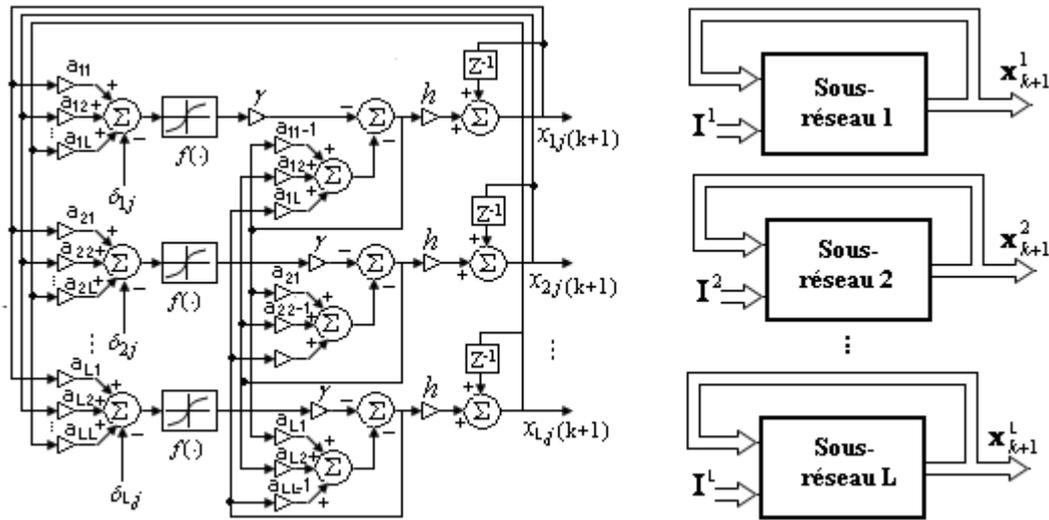


Figure 4.2. Architecture du réseau DZNN1-2

4.4. Résolution du système $Az = b$ par le réseau DZNN2

En suivant les mêmes démarches faites au chapitre 3, le réseau de Zhang pour résoudre l'équation $Az = b$ est montré dans la figure (4.3). La complexité du réseau est L fois inférieure à celle du réseau précédent.

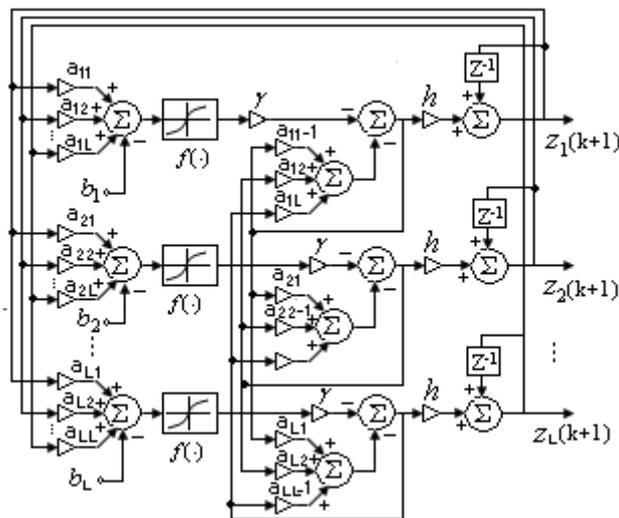


Figure 4.3. Architecture du réseau DZNN2 pour résoudre le système

4.5. Implémentation de l'approche proposée

Comme mentionné au début, le réseau DZNN performe l'inversion d'une matrice ou la résolution d'un système en temps réel, de ce fait, il constitue une bonne alternative pour résoudre le système d'équations (4.5). Nous n'avons qu'à considérer les éléments de la matrice A comme les poids du réseau. A partir d'un état initial aléatoire, le réseau de neurones itère tant qu'il n'a pas atteint une erreur de convergence présélectionnée. L'état final du réseau correspond à l'inverse de la matrice A cherchée [89-90].

La procédure de calcul du flot optique utilisant le réseau de Zhang consiste à calculer à partir de deux frame successives $I(x, y, t)$ et $I(x, y, t + 1)$, les dérivées spatio-temporelles I_x , I_y et I_t utilisant le filtre de Simoncelli après un simple préfiltrage (1/4,1/2,1/4) [Annexe B]. Nous construisons ensuite la matrice A et le vecteur b . La résolution du système (4.5) donnant le vecteur de mouvement est réalisée utilisant les deux réseaux DZNN1 ou DZNN2.

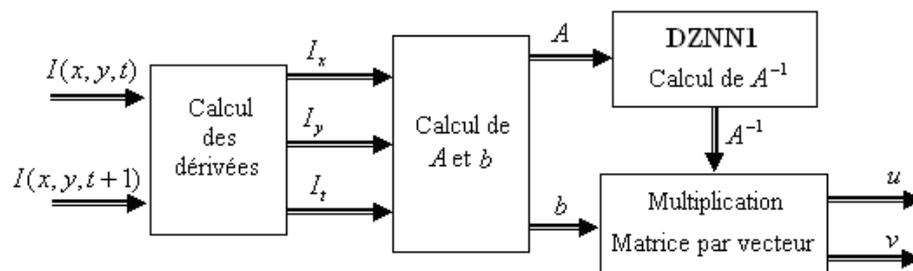


Figure 4.4. Schéma synoptique pour le calcul du flot optique par DZNN1-1

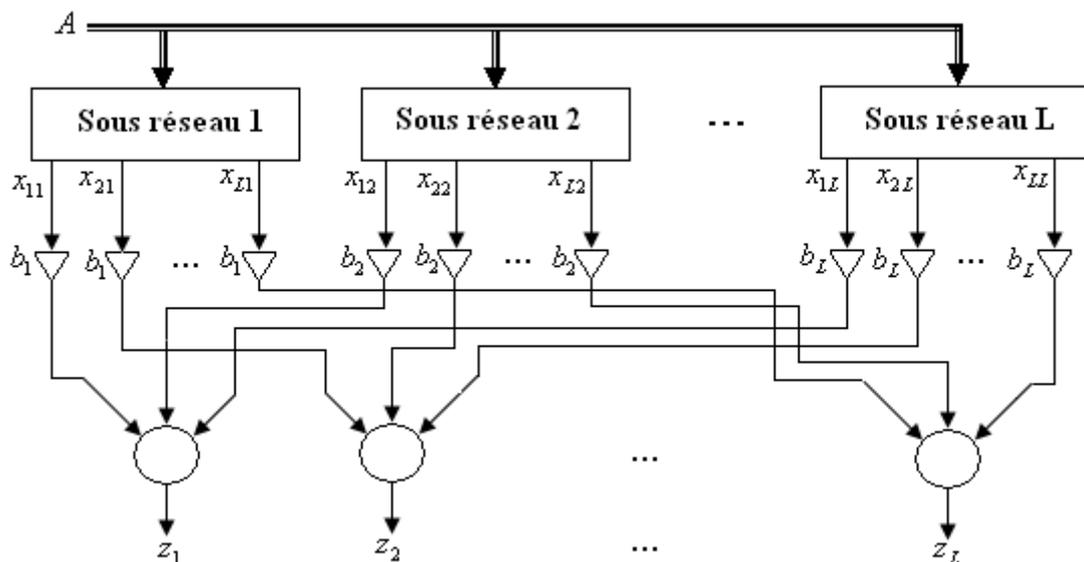


Figure 4.5. Schéma synoptique pour le calcul du flot optique par DZNN1-2

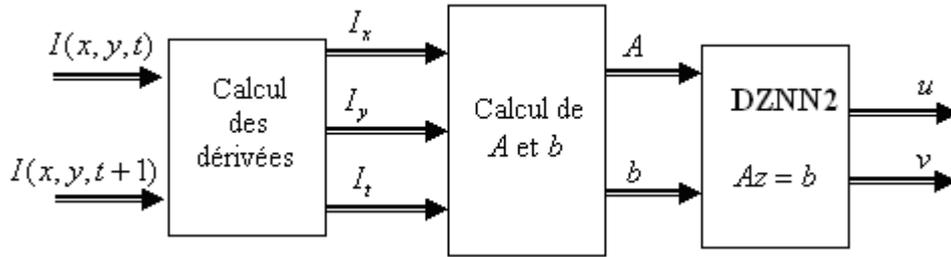


Figure 4.6. Schéma synoptique pour le calcul du flot optique par DZNN2

Les figures (4.4), (4.5) et (4.6) illustrent la procédure pour estimer le flot optique utilisant les réseaux DZNN1-1, DZNN1-2 et DZNN2.

Le système d'équations linéaires (4.5) est résolu soit en calculant l'inverse de la matrice A utilisant le réseau DZNN1-1 ou le réseau DZNN1-2 puis nous multiplions la sortie du réseau par le vecteur b , soit en résolvant directement le système d'équations utilisant le réseau DZNN2. Dans les deux premiers réseaux, il existe une multiplication matrice par vecteur qui consiste en L^2 multiplications scalaires et L additions de L nombres ce qui entraîne une complexité plus haute. Notons ici que l'approche proposée évalue le flot optique en parallèle pour tous les pixels de l'image.

4.6. Complexité de l'algorithme

Puisque la matrice A est composée de blocs tri-diagonaux (figure 4.7a), dont chacun possède une structure creuse simple, la majorité des éléments de la matrice des connexions sont nuls. De plus la matrice A est symétrique, donc, uniquement la moitié des éléments nécessitent la mémorisation. La figure (4.7.a) montre un exemple de la matrice A de taille 32×32 . La densité de la matrice A montrée dans la figure (4.7.b) est très petite ce qui montre que la matrice A est très creuse. Si nous désignons nnz comme le nombre des éléments non nuls de la matrice A , la complexité du réseau DZNN1 est réduite à $L(2nnz + L)$ multiplications comme montré dans la figure (4.7.c), tandis que la complexité du réseau DZNN2 est réduite à $(2nnz + L)$. Elle est illustrée dans la figure (4.7.d). Nous remarquons bien que la complexité du réseau DZNN2 est beaucoup inférieure devant la complexité du DZNN1 quand la taille de l'image augmente. Ceci est illustré par la figure (4.7.e) [89].

Pour calculer le flot optique, la méthode de Horn & Schunk nécessite $I_x, I_y, I_t, u(k), v(k), u(k+1)$ et $v(k+1)$; donc, $7N$ valeurs doivent être mémorisées. Avec l'approche utilisant le réseau DZNN, nous ne mémorisons que la moitié des nnz

éléments de la matrice A , et les N éléments du vecteur b . La figure (4.7.f) montre le nombre d'éléments à mémoriser dans l'approche proposée pour différentes tailles de l'image.

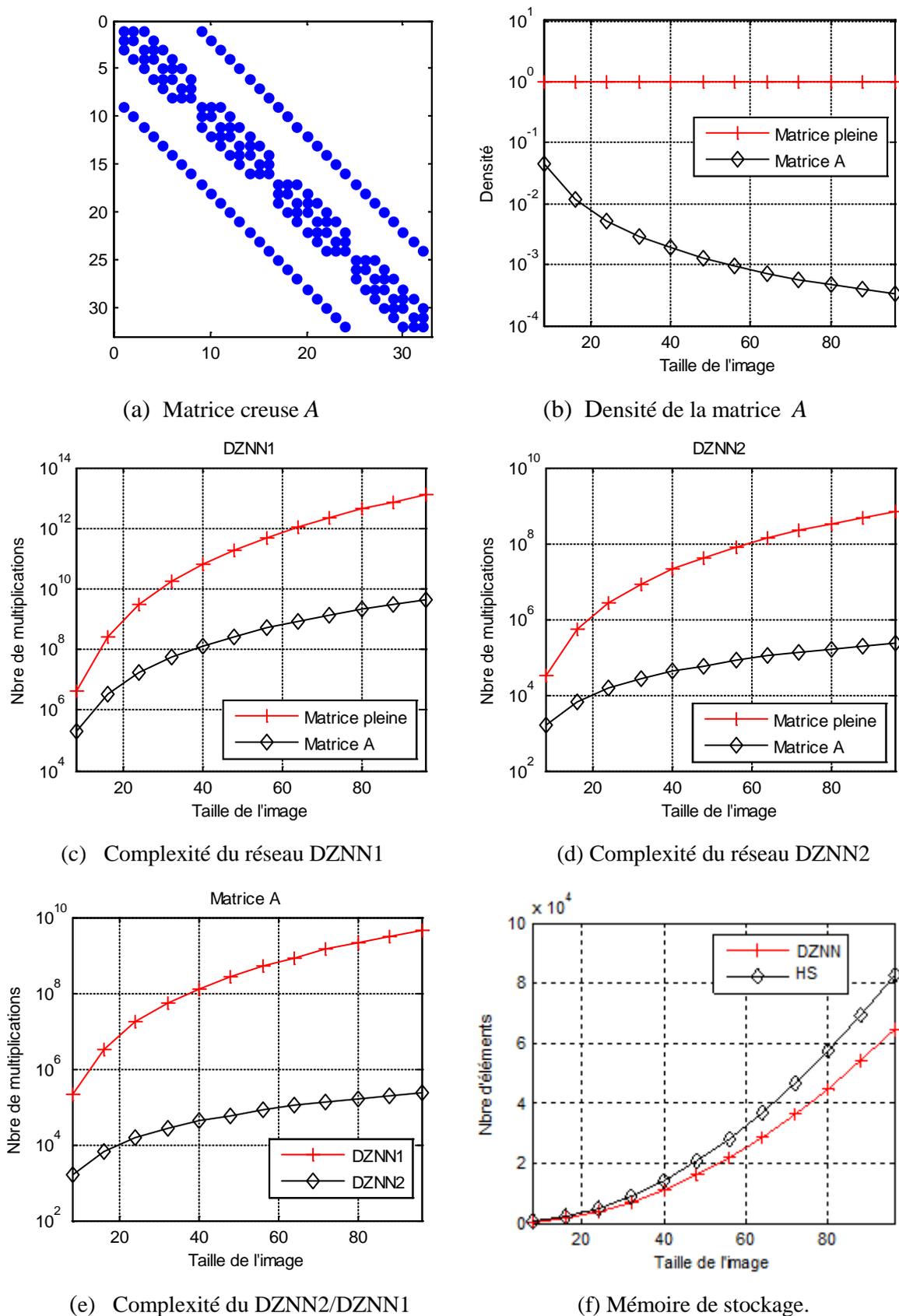


Figure 4.7. Complexité et mémoire de stockage.

Dans notre cas, chaque ligne de la matrice A contient 6 éléments non nuls au plus, par conséquent, il existe pas plus de 6 connexions par neurones. La totalité du réseau de neurones nécessite 11 cycles d'horloge indépendamment de la taille de la matrice A ce qui constitue un avantage majeur de l'approche proposée.

4.7. Résultats de simulation

Dans cette section, nous allons présenter et comparer les résultats de simulation obtenus par l'utilisation des trois types du réseau de Zhang ; DZNN1-1, DZNN1-2 et DZNN2 avec ceux obtenus par la méthode de Jaccobi et la méthode originale de HS. Pour démontrer les performances du réseau de Zhang pour l'estimation du flot optique, plusieurs types de séquence d'images ont été testés incluant les séquences synthétiques et les séquences réelles.

4.7.1. Schémas d'implantation des réseaux

Les trois réseaux sont implémentés dans l'environnement Simulink de MATLAB comme montré dans les figures (4.8), (4.9) et (4.10).

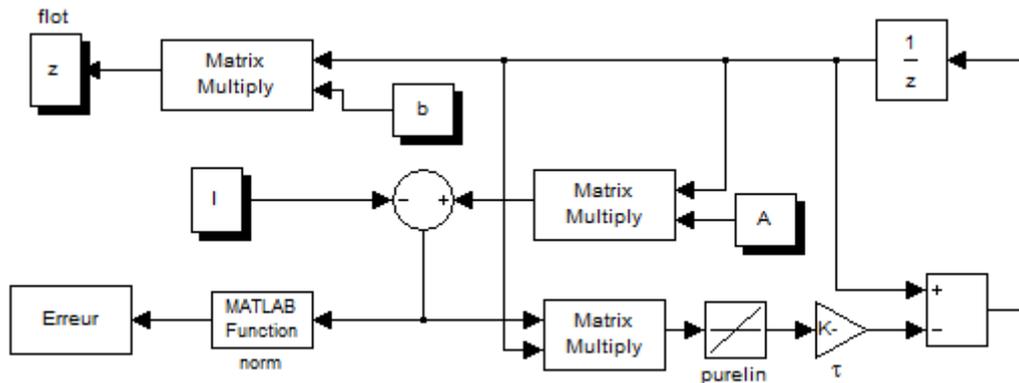


Figure 4.8. Schéma d'implémentation du réseau DZNN1-1.

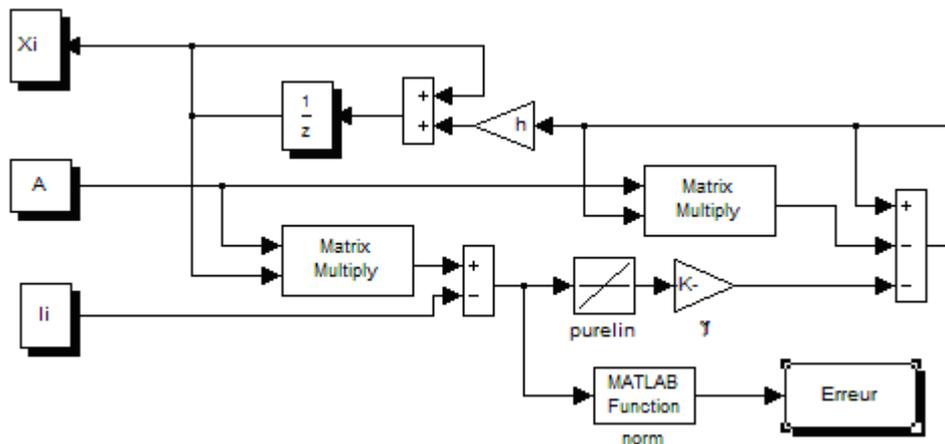


Figure 4.9. Schéma d'implémentation d'un bloc du réseau DZNN1-2.

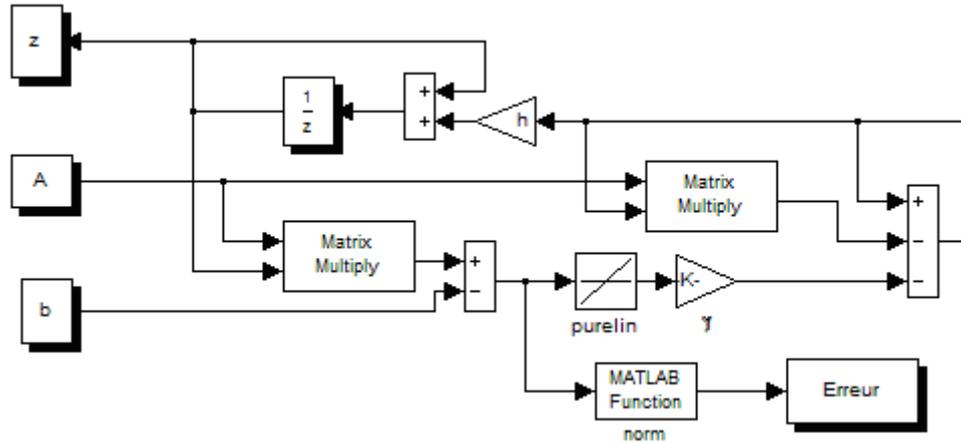
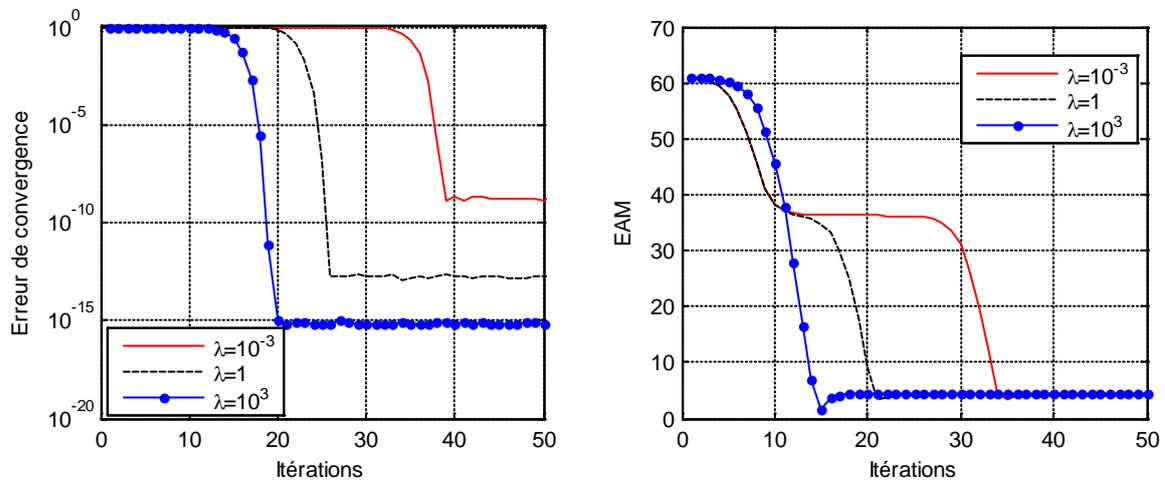


Figure 4.10. Schéma d'implémentation du réseau DZNN2.

Les réseaux présentés dans les figures 4.8, 4.10 évaluent la totalité du flot optique, par contre la figure (4.9) présente un bloc (sous réseau) du réseau DZNN1-2 qui sert à calculer une colonne X_i de la matrice A^{-1} . Tous les paramètres τ, γ et h sont égaux à 1.

4.7.2. Réglage du paramètre de lissage

Pour voir l'effet du paramètre de lissage sur la convergence des réseaux et sur la précision des résultats obtenus, la séquence 'sinusoïdale' de taille 128×128 a été utilisée. Les figures (4.11-15) présentent l'erreur de convergence et l'erreur angulaire en fonction de la contrainte de lissage pour les différentes approches DZNN1-1, DZNN1-2, DZNN2, Jacob et Horn & Schunk respectivement. Nous pouvons remarquer ici que les réseaux DZNN1-1 et DZNN1-2 sont sensibles au paramètre de lissage. Par contre les autres méthodes sont moins sensibles.



(a) Courbes de convergence

(b) Erreurs angulaires moyennes

Figure 4.11. Effet du paramètre de lissage sur le comportement du réseau DZNN1-1.

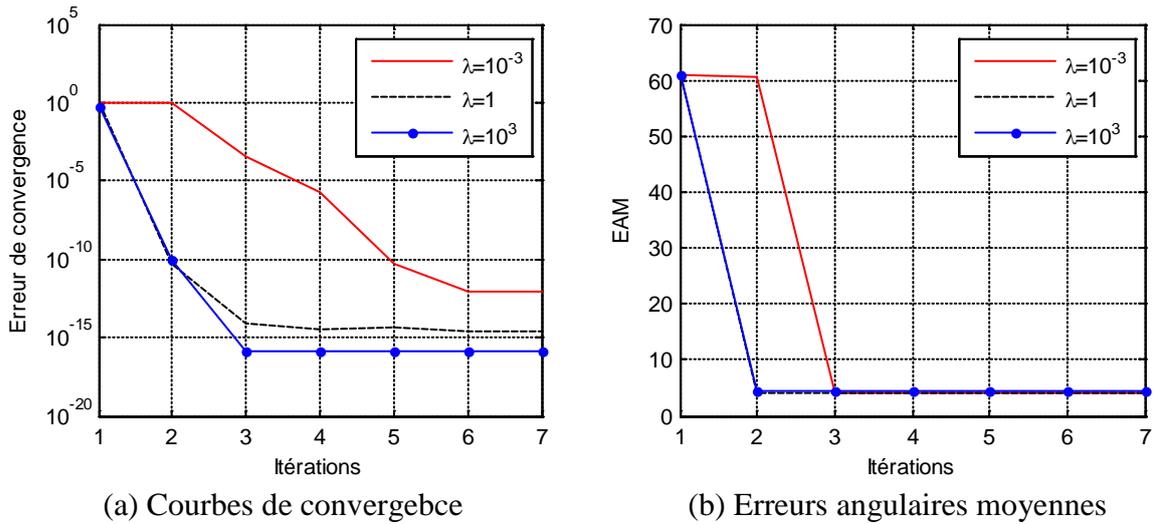


Figure 4.12. Effet du paramètre de lissage sur le comportement du réseau DZNN1-2.

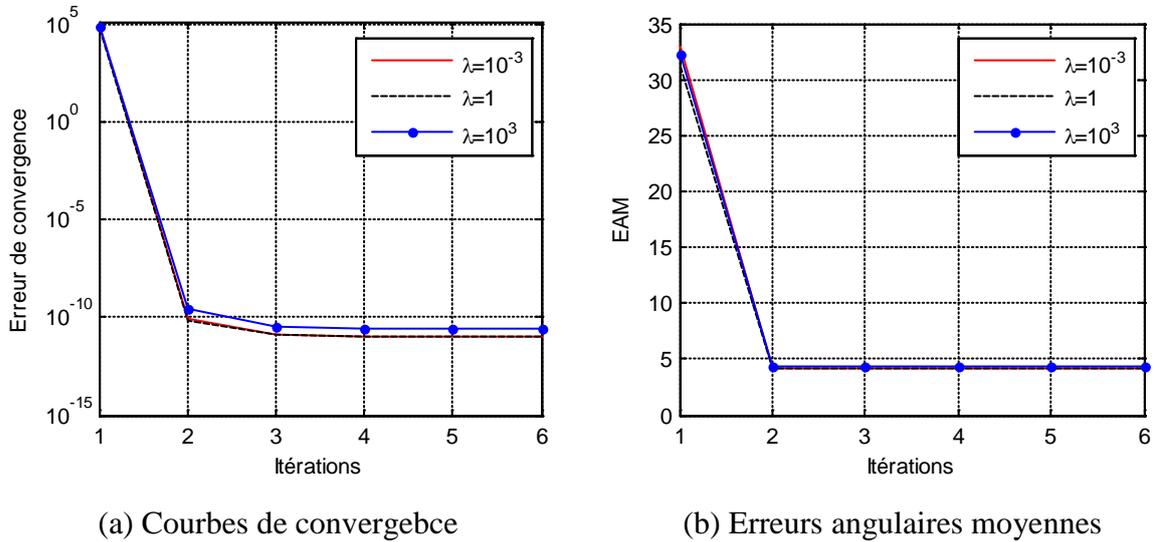


Figure 4.13. Effet du paramètre de lissage sur le comportement du réseau DZNN2.

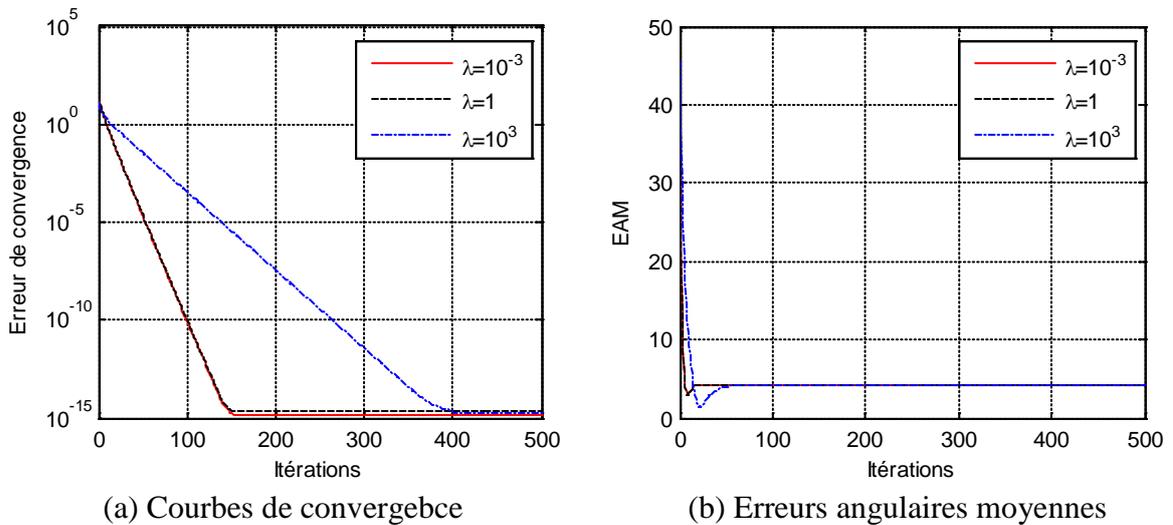


Figure 4.14. Effet du paramètre de lissage sur la méthode de Jacobi.

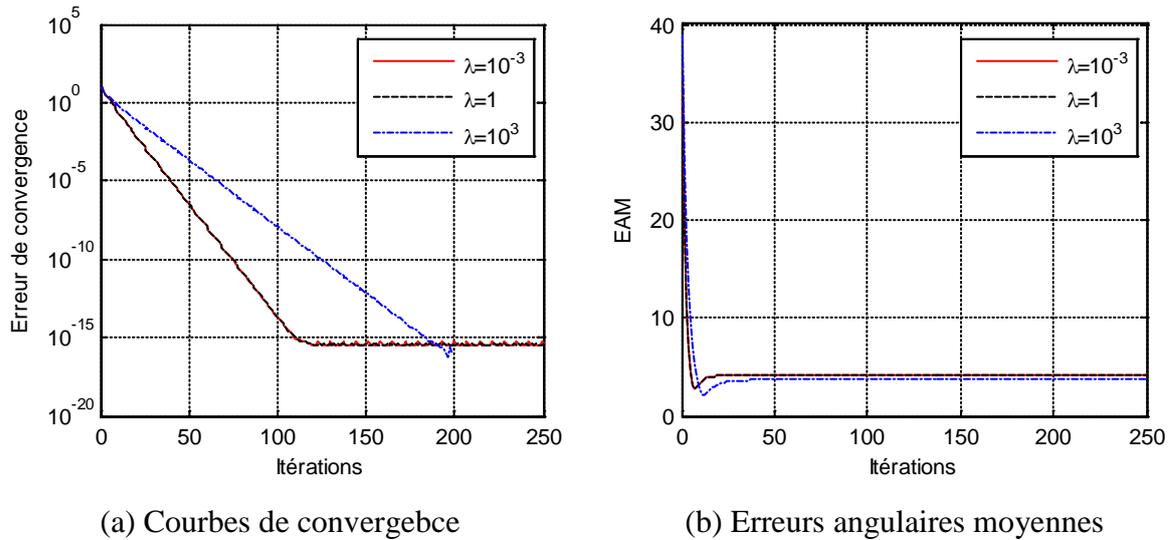


Figure 4.15. Effet du paramètre de lissage sur la méthode originale de Horn & Schunk.

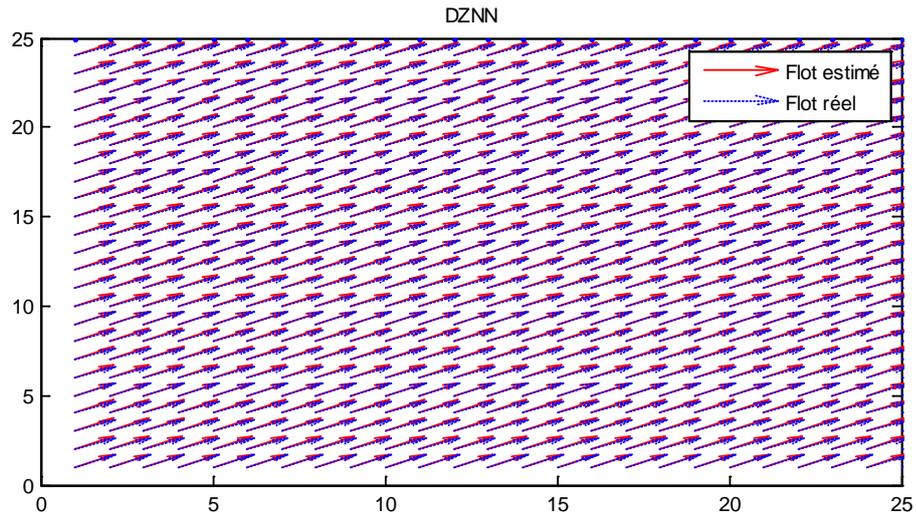
4.7.3. Résultats des séquences synthétiques

La figure (4.16) montre le flot optique obtenu par le réseau DZNN, Jacobi et Horn & Schunk. Notons ici que le flot optique estimé par les trois types de réseau est identique. Il est meilleur que ce lui obtenu par la méthode de Horn & Schunk comme montré dans la figure (4.16 c).

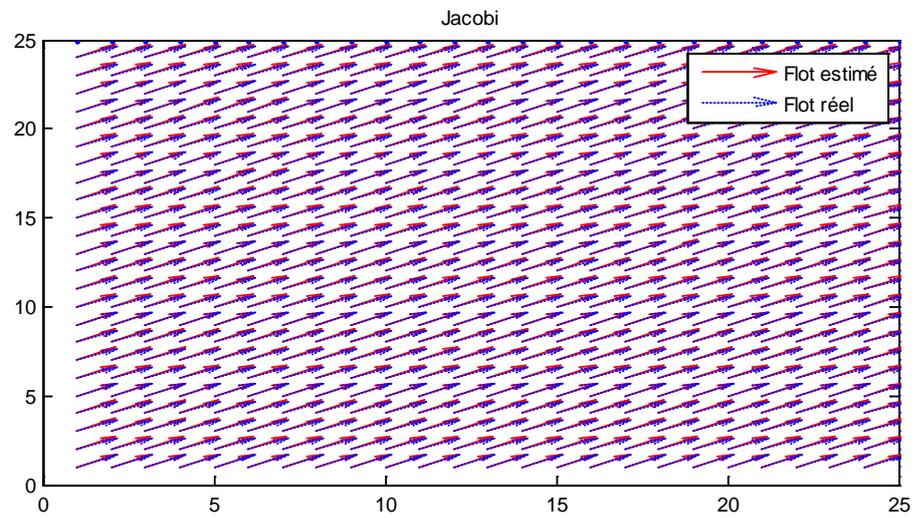
Dans la deuxième expérience, nous utilisons la célèbre séquence ‘Yosemite’ avec nuages. Les figures (4.17a,b) montrent une frame de cette séquence et le flot réel correspondant. Suite à la complexité modérée et l’efficacité du réseau DZNN2, ce dernier a été sélectionné parmi les trois types du réseau pour estimer le flot optique. Le vecteur de mouvement obtenu et l’erreur de convergence sont montrés sur les figures (4.17c,d).

Nous remarquons que l’approche proposée atteint une erreur minimale en 2 itérations. De plus, l’erreur angulaire obtenue 10.384° est plus petite que celle obtenue avec la méthode de HS modifiée (11.26°) [4].

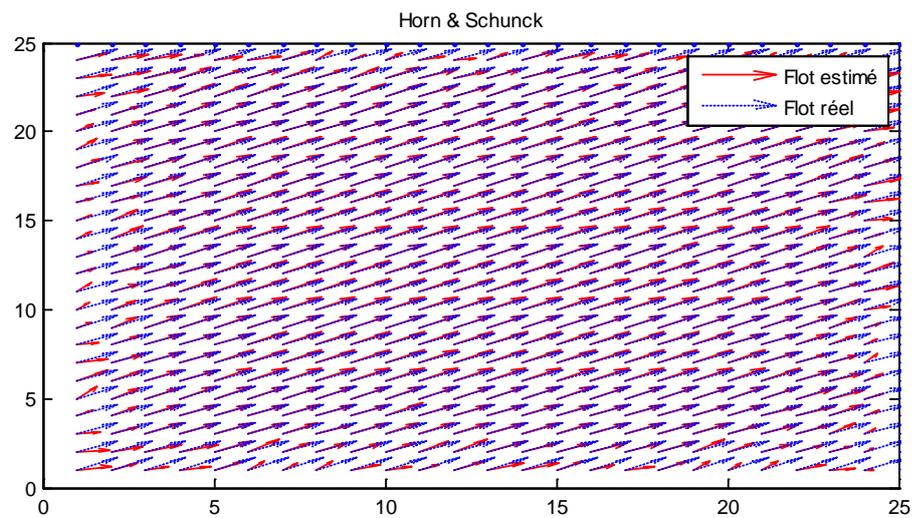
Pour mieux évaluer la méthode proposée, plusieurs tests ont été effectués utilisant des séquences d’images réelles dont le flot optique est connu disponibles dans la base des données de ‘Middlebury public’. La méthode de HS et le réseau DZNN ont été exécutés avec les mêmes paramètres ; $\lambda = 100$ et 30 itérations. Le tableau (4.1) résume les performances obtenues utilisant les deux méthodes pour les séquences Dimetrodon, Hydrangea, Grove 2 et Grove 3.



(a)



(b)



(c)

Figure 4.16. Flot optique estimé par les différentes méthodes.



(a) Frame (b) Flot optique réel (c) Flot optique estimé

Figure 4.17. Résultats de la séquence 'Yosemite' utilisant le réseau DZNN2.

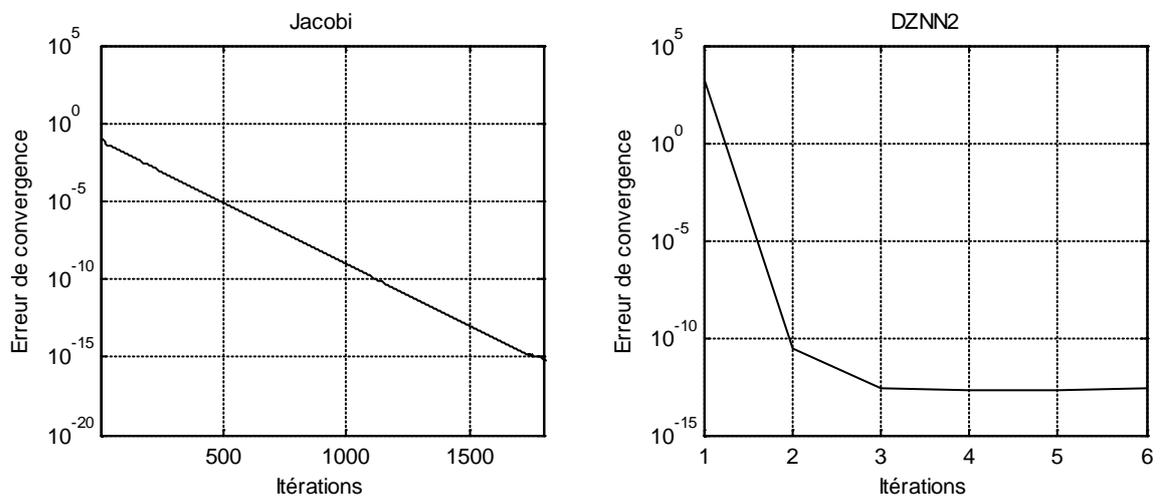
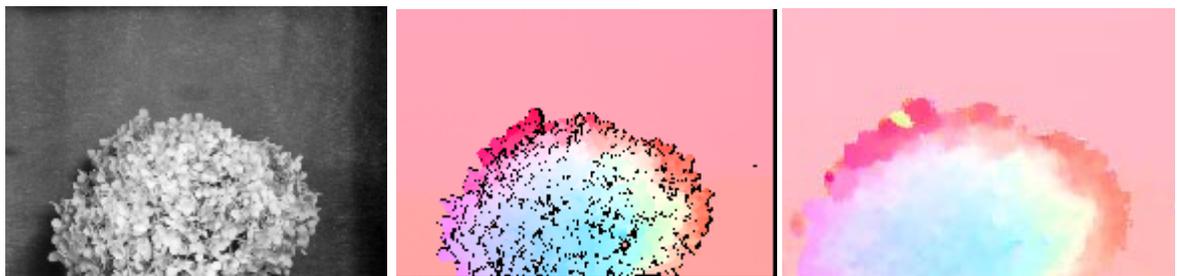


Figure 4.18. Erreur de convergence pour la séquence 'Yosemite'.



(a) Frame (b) Flot optique réel (c) Flot optique estimé

Figure 4.19. Résultats de la séquence 'Hydrangea' utilisant le réseau DZNN2

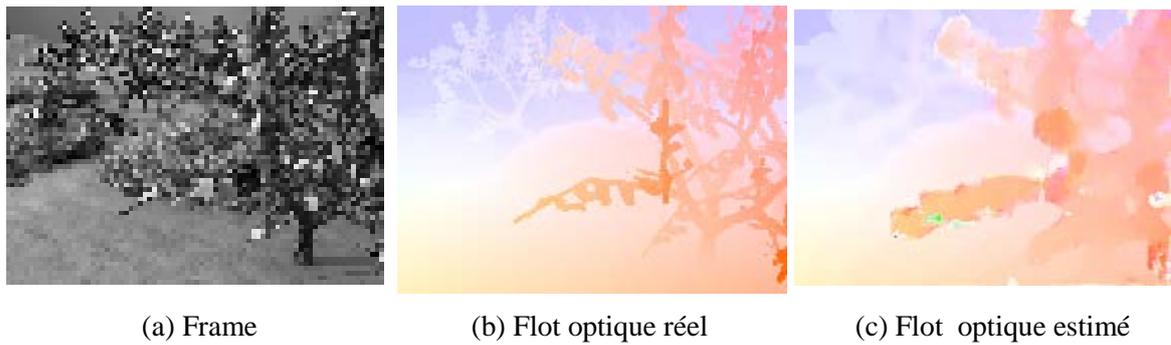


Figure 4.20. Résultats de la séquence ‘Grove3’ utilisant le réseau DZNN2

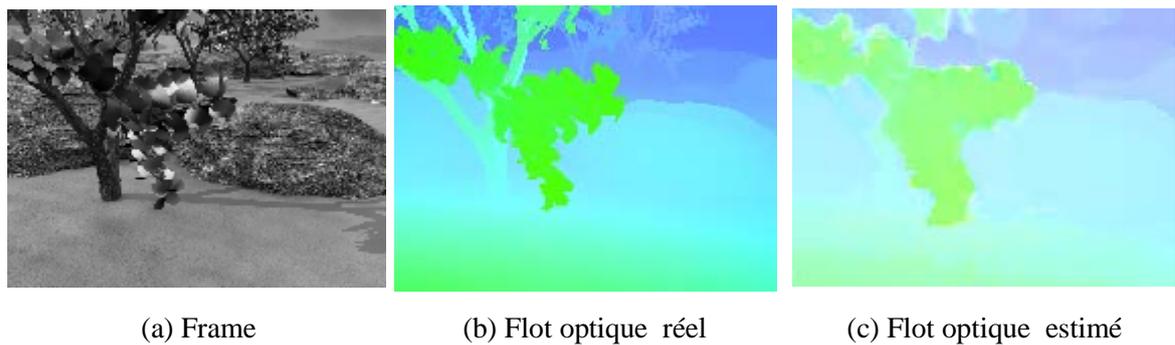


Figure 4.21. Résultats de la séquence ‘Grove2’ utilisant le réseau DZNN2

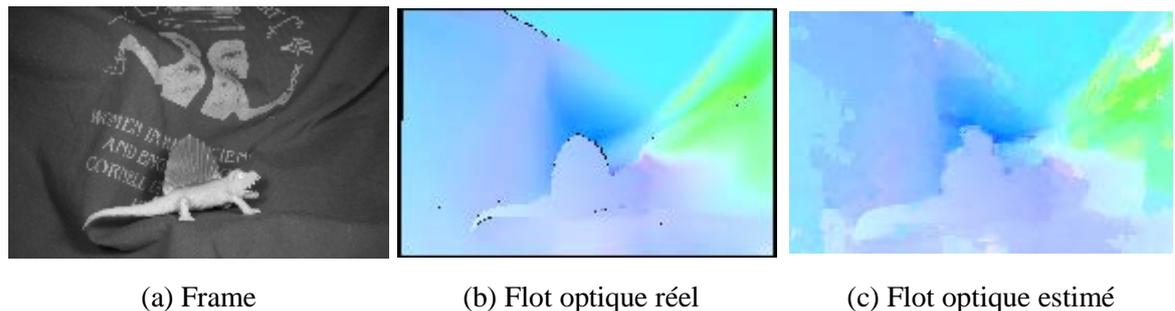


Figure 4.22. Résultats de la séquence ‘Dimetrodon’ utilisant le réseau DZNN2.

Nous remarquons que les performances des deux algorithmes pour les séquences Dimetrodon et Hydrangea sont identiques. Cependant pour les séquences Grove2 et Grove3 le réseau DZNN est plus performant. Nous devons noter ici que la précision de la méthode de HS est fortement liée au nombre d’itérations. Quand le nombre d’itération augmente suffisamment, les deux méthodes donnent les mêmes performances.

Séquences	Classical HS		DZNN2	
	AAE	EPE	AAE	EPE
Dimetrodon	4.938	0.255	4.131	0.211
Hydrangea	3.834	0.433	3.343	0.313
Grove 2	27.70	1.225	5.196	0.360
Grove 3	15.30	1.542	10.14	1.162

Tableau 4.1. EPE et AAE de quelques séquences de test de la base de données Middlebury

4.7.4 Résultats des séquences réelles

Finalement la le réseau DZNN2 a été testé avec la séquence ‘Hamburg Taxi’. Puisqu’il est difficile de déterminer le flot optique pour les séquences réelles, généralement un test qualitatif est utilisé. La validation du mouvement estimé dans ce cas peut être vérifiée en utilisant ces estimations dans la compensation du mouvement par interpolation bilinéaire.

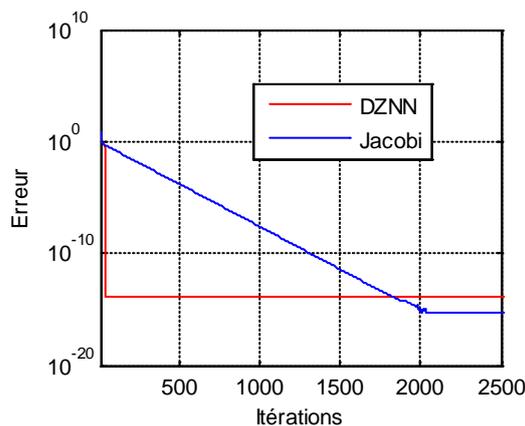


(a) Frame de la séquence ‘Taxi’

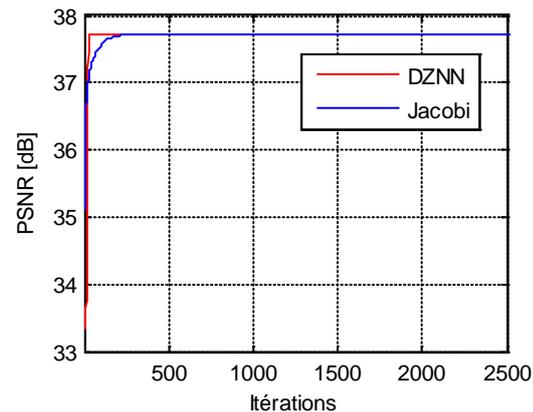


(b) Région d’intérêt

Figure 4.23. Frame de la séquence ‘Hamburg Taxi’



(a) Erreur de convergence



(b) PSNR

Figure 4.24. Performance du réseau pour la zone spécifiée

La figure (4.24) montre l’erreur de convergence et le PSNR de la méthode de Jacobi ainsi que du réseau DZNN2 pour $\lambda = 100$. Notons que le PSNR augmente de la même manière dans les deux méthodes. Mais le DZNN2 atteint le maximum du PSNR (37,3405dB) dans seulement 2 itérations, cependant la méthode de Jacobi nécessite 2040 itérations pour atteindre la même performance. La figure (4.25) résume les résultats de la zone du voiture. Nous pouvons voir facilement que les objets en mouvement sont clairement capturés et localisés.

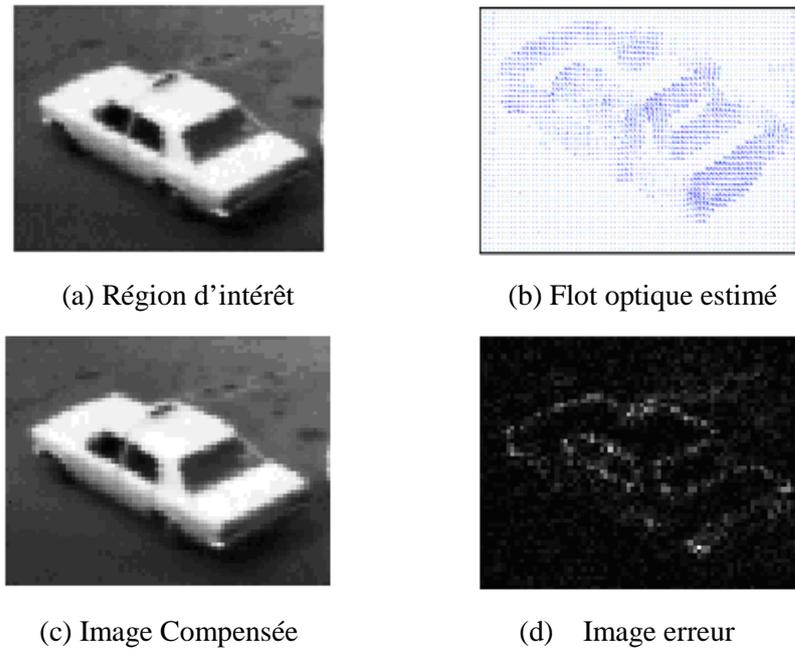


Figure 4.25. Résultats de la séquence 'Hamburg Taxi'.

4.8. Conclusion

Dans ce chapitre nous avons proposé une version discrète du réseau de Zhang pour l'estimation du flot optique en se basant sur la reformulation de Mitiche et Mansouri de la méthode différentielle de Horn & Schunk qui engendre un système large d'équations linéaires. Pour résoudre ce système, nous avons proposé trois types de réseaux de neurones DZNN1-1, DZNN1-2 et DZNN2. Ces trois réseaux sont caractérisés par leur rapidité de convergence sans altérer la précision des résultats. De plus, le réseau DZNN2 ayant la plus faible complexité ne nécessite que deux ou trois itérations pour la convergence ce qui le qualifie pour les applications en temps réel.

Chapitre 5

Extension à l'estimation de mouvement 3D

5.1. Introduction

L'estimation du mouvement 3D consiste à extraire un vecteur de mouvement 3D à partir d'une séquence volumique en faisant l'hypothèse que l'intensité (ou la couleur) est conservée au cours du déplacement.

En effet, les méthodes d'acquisition en trois dimensions s'étant développées, par exemple l'utilisation de multiples caméras (stéréoscopiques, triscopiques) ou encore la fusion d'informations provenant de radar-laser et de caméras. Elles ouvrent la voie à de nouvelles possibilités d'application des méthodes d'estimation du mouvement. Que ce soit le calcul du flot optique ou des méthodes de mise en correspondance, l'application de la méthode proposée dans le cadre d'une estimation du mouvement en 3D sera l'objectif de ce chapitre. Nous présenterons également l'extension de la méthode différentielles de Horn & Schunck et celle de Lucas & Kanade pour l'estimation de mouvement d'une séquence d'images volumiques [91-94].

5.2. Extension tridimensionnelle

Dans un premier temps, nous présentons les détails de la méthode de LK_3D et la méthode de HS_3D proposées dans [91]. Ceci nous permet d'introduire ensuite, les modifications que nous proposerons.

5.2.1. Equation du flot optique 3D

Une séquence de volumes peut être représentée par sa fonction de luminance $I(x, y, z, t)$. L'hypothèse de conservation de la luminance stipule que la luminance d'un point physique de la séquence d'image ne varie pas au cours du temps, c'est à dire [91]:

$$I(p, t) = I(p + V(p), t + dt) \quad (5.1)$$

avec $p = (x, y, z)^T$ et $V(p) = (u, v, w)^T$ le vecteur vitesse 3D associé au point p à l'instant t . Les composantes u , v et w sont respectivement la vitesse selon les directions x , y et z .

L'équation de la contrainte du flot optique peut être écrite sous forme suivante [91]:

$$I_x u + I_y v + I_z w + I_t = 0 \quad (5.2)$$

avec : I_x, I_y, I_z et I_t sont les dérivées spatiotemporelles tridimensionnelles dans un voisinage $n \times n \times n$ centré au voxel (x, y, z) .

5.2.2. Voisinage d'un voxel

Trois types de voisinages sont rencontrés, le voisinage 6-adjacents, 18-adjacents et 26-adjacents. Deux voxels sont dits 6-adjacents (adjacence par faces) si leurs coordonnées diffèrent de ± 1 sur exactement une coordonnée (figure 5.1.a).

Deux voxels sont dits 26-adjacents (adjacence par sommets) si leurs coordonnées diffèrent de ± 1 sur une, deux ou trois coordonnées (figure 5.1.b).

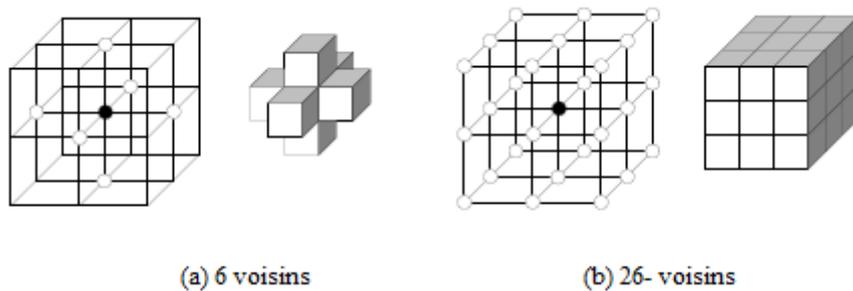


Figure 5.1 Représentation des voisinages d'un voxel.

5.3. Méthode de Lucas & Kanade 3D

Barron et Thacker [91] ont proposé une extension de la méthode originale de LK. Ils ont supposé que l'équation du flot optique 3D demeurerait constante dans de petits volumes de l'image Ω .

5.3.1. Principe

Cette méthode consiste à minimiser la quantité suivante :

$$E_{LK3D} = \sum_{p \in \Omega} w^2(p) (I_x(p, t)u + I_y(p)v + I_z(p)w + I_t(p))^2 \quad (5.3)$$

où $w(p)$ correspond à une fonction fenêtre isotropique ayant le rôle d'attribuer plus de poids au centre que sur les extrémités du voisinage. Ceci est réalisable par exemple par

l'application d'un filtre gaussien 3D de taille $5 \times 5 \times 5$ (le masque 1D est : $[0.0625, 0.25, 0.375, 0.25, 0.0625]$)

La solution de l'équation (5.3) est donnée par :

$$\vec{V} = [A^T W^2 A]^{-1} A^T W^2 b \quad (5.4)$$

avec:

$$W = \text{diag}[w(p_1), \dots, w(p_N)], A = (\nabla I(p_1), \dots, \nabla I(p_N))^T, b = -(I_x(p_1), \dots, I_x(p_N))^T \text{ et } N = n^3$$

qui est équivalent à:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y & \sum W^2 I_x I_z \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 & \sum W^2 I_y I_z \\ \sum W^2 I_z I_x & \sum W^2 I_z I_y & \sum W^2 I_z^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum W^2 I_x I_t \\ -\sum W^2 I_y I_t \\ -\sum W^2 I_z I_t \end{bmatrix} \quad (5.5)$$

Le problème, c'est que parfois la matrice $A^T W^2 A$ est singulière : l'estimateur obtenu a alors une variance très élevée. Nous pouvons contourner ce problème de singularité en ajoutant une constante à toutes les valeurs propres de la matrice $A^T W^2 A$, et en calculant :

$$V = [A^T W^2 A + \lambda I_3]^{-1} A^T W^2 b \quad (5.6)$$

où I_3 est la matrice identité de taille 3×3 et λ est un paramètre de régularisation positif.

5.3.2. Méthode de Lucas & Kanade 3D modifiée

L'équation (5.6) exige le calcul de la matrice inverse $[A^T W^2 A + \lambda I_3]^{-1}$. Dans ce chapitre nous proposons d'utiliser le réseau de Zhang DZNN2 pour résoudre le système directement sans passer par l'inversion de la matrice.

Le temps de calcul est également un avantage, en effet le nombre d'opérations nécessaires est limité au:

- calcul et multiplication des dérivées partielles I_x, I_y, I_z et I_t , puis leur sommation sur le voisinage Ω pour la construction des systèmes linéaires locaux.
- résolution d'un système 3×3 par voxel par le réseau DZNN2.

5.4. Méthode de Horn & Schunck 3D

5.4.1. Principe

La méthode de Horn & Schunck 3D est issue de l'extension de la fonction de coût à minimiser aux images volumiques [91].

$$E_{HS3D} = \int_{\Omega} \left((I_t + uI_x + vI_y + wI_z)^2 + \lambda^2 (\|\nabla u\|^2 + \|\nabla v\|^2 + \|\nabla w\|^2) \right) dx dy dz \quad (5.7)$$

où $(\nabla u, \nabla v, \nabla w)$ sont les dérivées spatiales de $V = (u, v, w)$ et λ^2 est un coefficient pondérant l'importance du lissage.

Après application du théorème d'Euler-Lagrange, nous obtenons l'ensemble d'équations suivant :

$$\begin{cases} I_x^2 u_x + I_x I_y v + I_x I_z w + I_x I_t = \lambda^2 \Delta^2 u \\ I_x I_y u + I_y^2 v + I_y I_z w + I_y I_t = \lambda^2 \Delta^2 v \\ I_z I_x u + I_z I_y v + I_z^2 w + I_z I_t = \lambda^2 \Delta^2 w \end{cases} \quad (5.8)$$

où $\nabla^2 v$, $\nabla^2 u$ et $\nabla^2 w$ sont les laplaciens des composantes u , v et w du flot optique V :

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}, \quad \nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \quad \text{et} \quad \nabla^2 w = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}$$

substituons une approximation du Laplacien :

$$\begin{cases} \nabla^2 u \approx u - \bar{u} \\ \nabla^2 v \approx v - \bar{v} \\ \nabla^2 w \approx w - \bar{w} \end{cases} \quad (5.9)$$

où \bar{u} , \bar{v} et \bar{w} sont les moyennes des composantes du flot optique dans un certain voisinage.

On obtient donc :

$$\begin{cases} (\lambda^2 + I_x^2)u + I_x I_y v + I_x I_z w = \lambda^2 \bar{u} - I_x I_t \\ I_y I_x u + (\lambda^2 + I_y^2)v + I_y I_z w = \lambda^2 \bar{v} - I_y I_t \\ I_z I_x u + I_z I_y v + (\lambda^2 + I_z^2)w = \lambda^2 \bar{w} - I_z I_t \end{cases} \quad \forall (x, y, z) \in \Omega \quad (5.10)$$

Avec ce système d'équations, il est possible d'obtenir une solution pour u , v et w à chaque position $(x, y, z) \in \Omega$, en utilisant la méthode de Cramer. Nous obtenons ainsi les équations de récursion:

$$\begin{cases} u^{it+1} = \bar{u}^{it} - I_x \frac{I_x \bar{u}^{it} + I_y \bar{v}^{it} + I_z \bar{w}^{it} + I_t}{\lambda^2 + I_x^2 + I_y^2 + I_z^2} \\ v^{it+1} = \bar{v}^{it} - I_y \frac{I_x \bar{u}^{it} + I_y \bar{v}^{it} + I_z \bar{w}^{it} + I_t}{\lambda^2 + I_x^2 + I_y^2 + I_z^2} \\ w^{it+1} = \bar{w}^{it} - I_z \frac{I_x \bar{u}^{it} + I_y \bar{v}^{it} + I_z \bar{w}^{it} + I_t}{\lambda^2 + I_x^2 + I_y^2 + I_z^2} \end{cases} \quad (5.11)$$

où it désigne le numéro de l'itération.

5.4.2. Méthode de Horn & Schunck 3D modifiée

5.4.2.1. Reformulation de la méthode de HS 3D

De la même manière que dans le cas 2D, nous discrétisons le domaine Ω sur une grille d'espace unitaire dont chaque point est indexé par $\{1, 2, \dots, N\}$ avec N le nombre de voxels dans l'image volumique. Une discrétisation de l'équation d'Euler-Lagrange est donnée par :

$$\begin{cases} I_{xi}^2 u_i + I_{xi} I_{yi} v_i + I_{xi} I_{zi} w_i + I_{xi} I_{ti} - \lambda^2 \sum_{j \in N_i} (u_j - u_i) = 0 \\ I_{xi} I_{yi} u_i + I_{yi}^2 v_i + I_{yi} I_{zi} w_i + I_{yi} I_{ti} - \lambda^2 \sum_{j \in N_i} (v_j - v_i) = 0 \\ I_{zi} I_{xi} u_i + I_{zi} I_{yi} v_i + I_{zi}^2 w_i + I_{zi} I_{ti} - \lambda^2 \sum_{j \in N_i} (w_j - w_i) = 0 \end{cases} \quad (5.12)$$

Pour $i \in \{1, 2, \dots, N\}$, ceci constitue un système d'équations creux avec $3N$ inconnus u_i, v_i et w_i :

$$Az = b \quad (5.13)$$

où z, b sont deux vecteurs réels de longueur $3N$ ayant les coordonnées pour $i \in \{1, \dots, N\}$:

$$\begin{cases} z_{3i-2} = u_i \\ z_{3i-1} = v_i \\ z_{3i} = w_i \end{cases} \quad (5.14)$$

$$\begin{cases} b_{3i-2} = -I_{xi} I_{ti} \\ b_{3i-1} = -I_{yi} I_{ti} \\ b_{3i} = -I_{zi} I_{ti} \end{cases} \quad (5.15)$$

A est une matrice de taille $3N \times 3N$, symétrique et définie positive, dont les éléments :

$$\begin{cases} A_{3i-2, 3i-2} = I_{xi}^2 + \lambda^2 c_i \\ A_{3i-1, 3i-1} = I_{yi}^2 + \lambda^2 c_i \\ A_{3i, 3i} = I_{zi}^2 + \lambda^2 c_i \\ A_{3i-2, 3i-1} = A_{3i-1, 3i-2} = I_{xi} I_{yi} \\ A_{3i-2, 3i} = A_{3i, 3i-2} = I_{xi} I_{zi} \\ A_{3i-1, 3i} = A_{3i, 3i-1} = I_{yi} I_{zi} \\ A_{3i-2, 3j-1} = A_{3i-1, 3j-1} = A_{3i, 3j} = -\lambda^2 \end{cases} \quad \text{pour } i \in \{1, \dots, N\} \text{ et } j \in N_i \quad (5.16)$$

Tous les autres éléments sont égaux à zéro.

Pour appliquer le réseau de Zhang pour résoudre itérativement le système (5.13), la matrice doit être définie positive pour assurer la convergence.

5.4.2.2. Convergence de l'algorithme proposé

La matrice A est définie positive et symétrique, nous allons vérifier que $z^T A z > 0$ pour tout $z \in R^{3N}$, $z \neq 0$. Nous avons:

$$\begin{aligned}
z^T A z &= \sum_{i=1}^N \left((I_{xi}^2 + \lambda^2 c_i) u_i + I_{xi} I_{yi} v_i + I_{xi} I_{zi} w_i - \lambda^2 \sum_{j \in N_i} u_j \right) u_i \\
&+ \sum_{i=1}^N \left(I_{xi} I_{yi} u_i + (I_{yi}^2 + \lambda^2 c_i) v_i + I_{yi} I_{zi} w_i - \lambda^2 \sum_{j \in N_i} v_j \right) v_i \\
&+ \sum_{i=1}^N \left(I_{xi} I_{zi} u_i + I_{zi} I_{yi} v_i + (I_{zi}^2 + \lambda^2 c_i) w_i - \lambda^2 \sum_{j \in N_i} w_j \right) w_i \\
&= \sum_{i=1}^N \left((I_{xi}^2 u_i^2 + I_{yi}^2 v_i^2 + I_{zi}^2 w_i^2 + 2I_{xi} I_{yi} u_i v_i + 2I_{xi} I_{zi} u_i w_i + 2I_{yi} I_{zi} v_i w_i) \right. \\
&\quad \left. - \lambda^2 c_i (u_i^2 + v_i^2 + w_i^2) - \lambda^2 \sum_{j \in N_i} u_j u_i - \lambda^2 \sum_{j \in N_i} v_j v_i - \lambda^2 \sum_{j \in N_i} w_j w_i \right) \\
&= \sum_{i=1}^N (I_{xi} u_i + I_{yi} v_i + I_{zi} w_i)^2 + \lambda^2 \sum_{i=1}^N (c_i (u_i^2 + v_i^2 + w_i^2) - \sum_{j \in N_i} u_j u_i - \sum_{j \in N_i} v_j v_i - \sum_{j \in N_i} w_j w_i)
\end{aligned}$$

D'autre part, nous avons :

$$\begin{aligned}
&\sum_{i=1}^N (c_i (u_i^2 + v_i^2 + w_i^2) - \sum_{j \in N_i} u_j u_i - \sum_{j \in N_i} v_j v_i - \sum_{j \in N_i} w_j w_i) \\
&= \sum_{i=1}^N \left(\sum_{j \in N_i, j > i} (u_i^2 + u_j^2 - 2u_i u_j) + \sum_{j \in N_i, j > i} (v_i^2 + v_j^2 - 2v_i v_j) + \sum_{j \in N_i, j > i} (w_i^2 + w_j^2 - 2w_i w_j) \right) \\
&= \sum_{i=1}^N \sum_{j \in N_i, j > i} ((u_i - u_j)^2 + (v_i - v_j)^2 + (w_i - w_j)^2)
\end{aligned}$$

Ceci implique :

$$z^T A z = \sum_{i=1}^N (I_{xi} u_i + I_{yi} v_i + I_{zi} w_i)^2 + \lambda^2 \sum_{i=1}^N \sum_{j \in N_i, j > i} ((u_i - u_j)^2 + (v_i - v_j)^2 + (w_i - w_j)^2) \quad (5.17)$$

Pour $z \neq 0$, nous avons $z^T A z = 0$ si et seulement si les termes dans les deux sommes dans la partie droite de l'équation (5.17) sont nulles. Les termes dans la deuxième somme s'annulent si et seulement si V est constant sur tout le volume Ω . Les termes de la première somme s'annulent si et seulement si le vecteur constant de vitesse est orthogonal au gradient spatial de l'image pour tous les voxels dans Ω [19]. Nous concluons que $z^T A z > 0$ pour tout $z \neq 0$.

5.5. Expérimentations et résultats

5.5.1. Séquences de test

5.5.1.1. Séquence synthétique

La séquence synthétique de taille $81 \times 78 \times 17$ est composée de deux images 3D. La figure (5.2) présente le volume à l'instant t , alors que les figures (5.3) et (5.4) présentent les 12 coupes horizontales et les deux coupes selon les axes x et y de chaque volume.

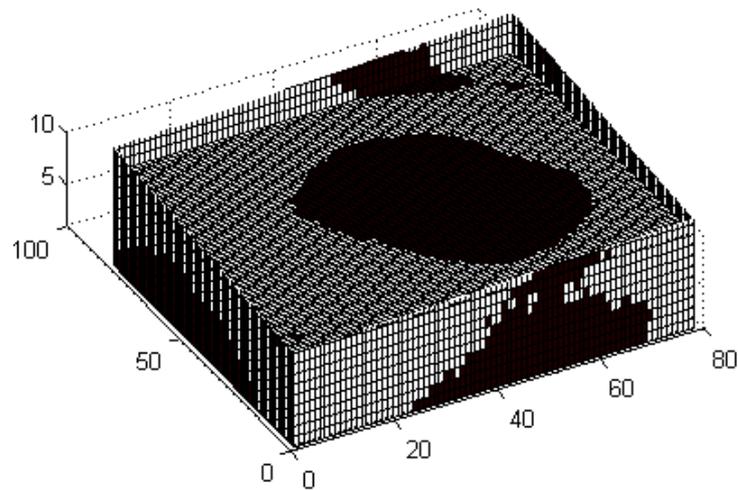


Figure 5.2. Image 3D de la séquence synthétique.

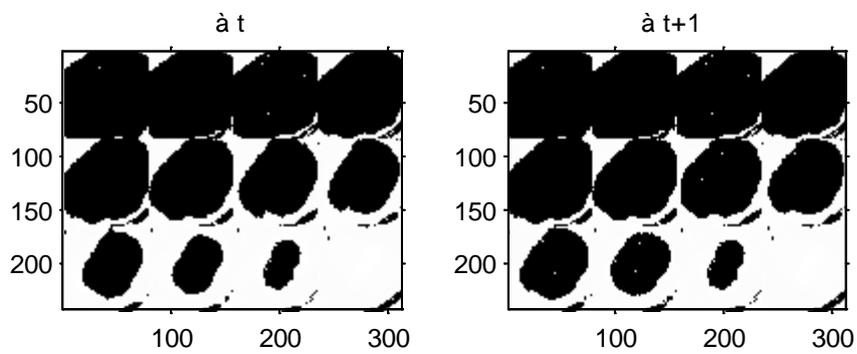


Figure 5.3. 12 coupes horizontales de la séquence synthétique.

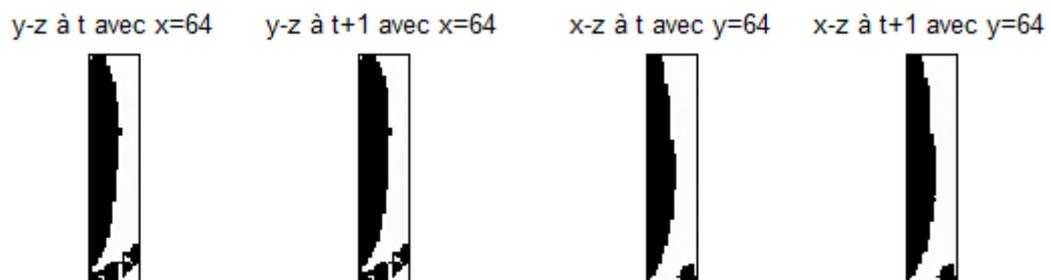


Figure 5.4. Coupes frontales selon x et y de la séquence synthétique.

5.5.1.2. Séquence réelle médicale

La deuxième séquence est une séquence de 20 images cardiaques à résonance magnétique du cœur de taille $256 \times 256 \times 12$ obtenue à partir du lien www.cse.yorku.ca/~mridataset/ [95].

Les figures (5.5-9) présentent les 12 coupes horizontales et les deux coupes frontales selon x et y de chaque volume.

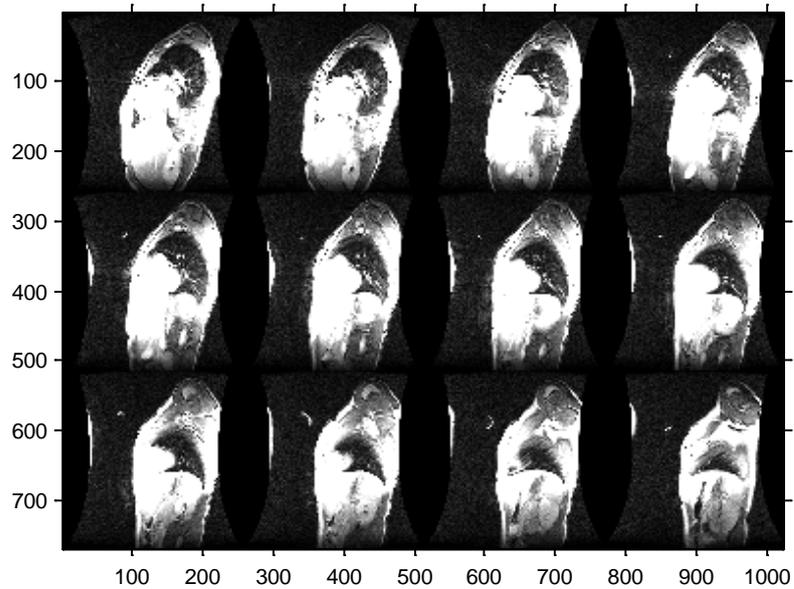


Figure 5.5. 12 coupes horizontales de la séquence réelle à l'instant $t = 2$.

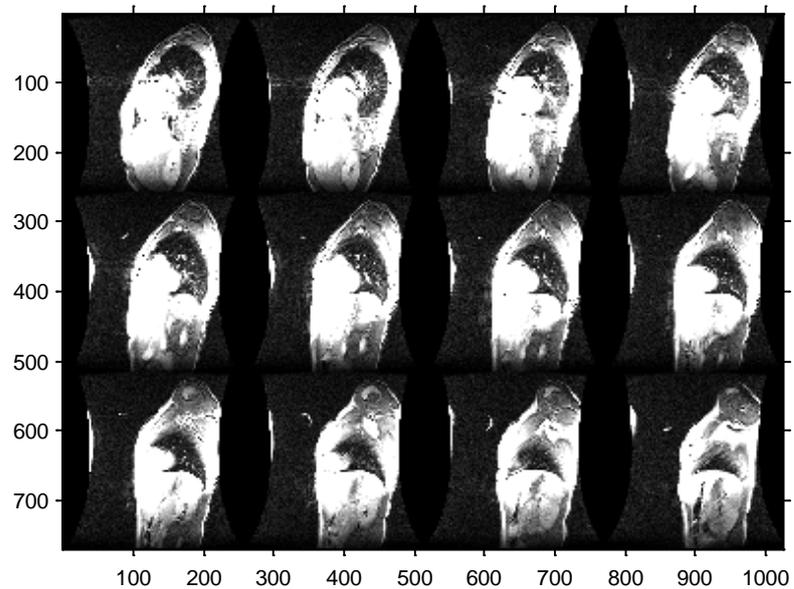


Figure 5.6. 12 coupes horizontales de la séquence réelle à l'instant $t = 3$.

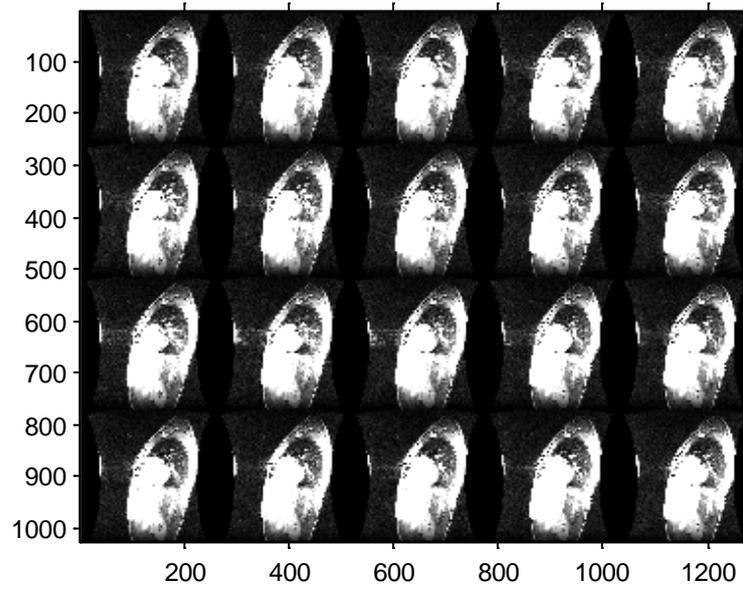


Figure 5.7. Coupe horizontale 3 pour différents instants.

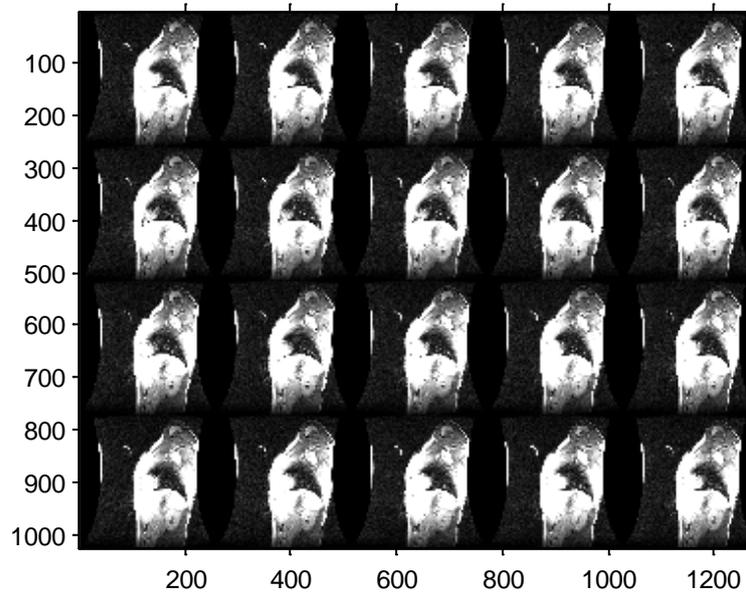


Figure 5.8. Coupe horizontale 10 pour différents instants.

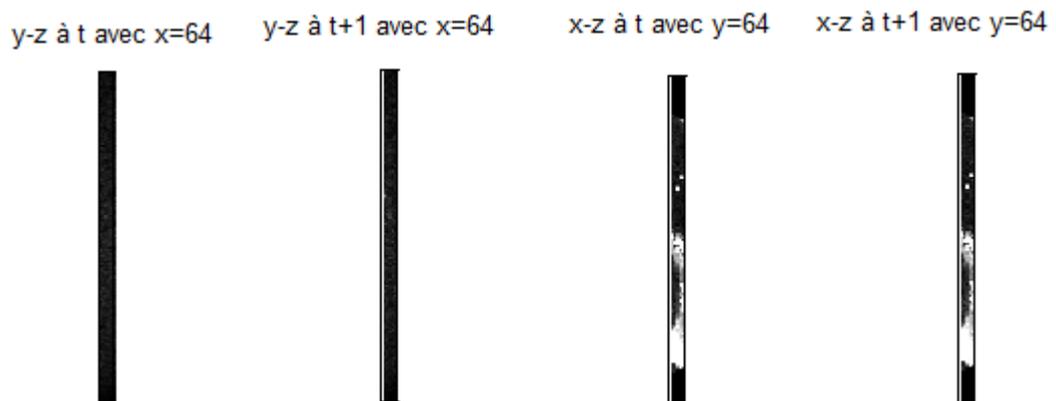


Figure 5.9. Coupes frontales selon x et y de la séquence réelle

Pour mieux comprendre le mouvement de la séquence, nous ajoutons les contours des projections selon les trois axes de quelques coupes comme illustrés par la figure (5.10).

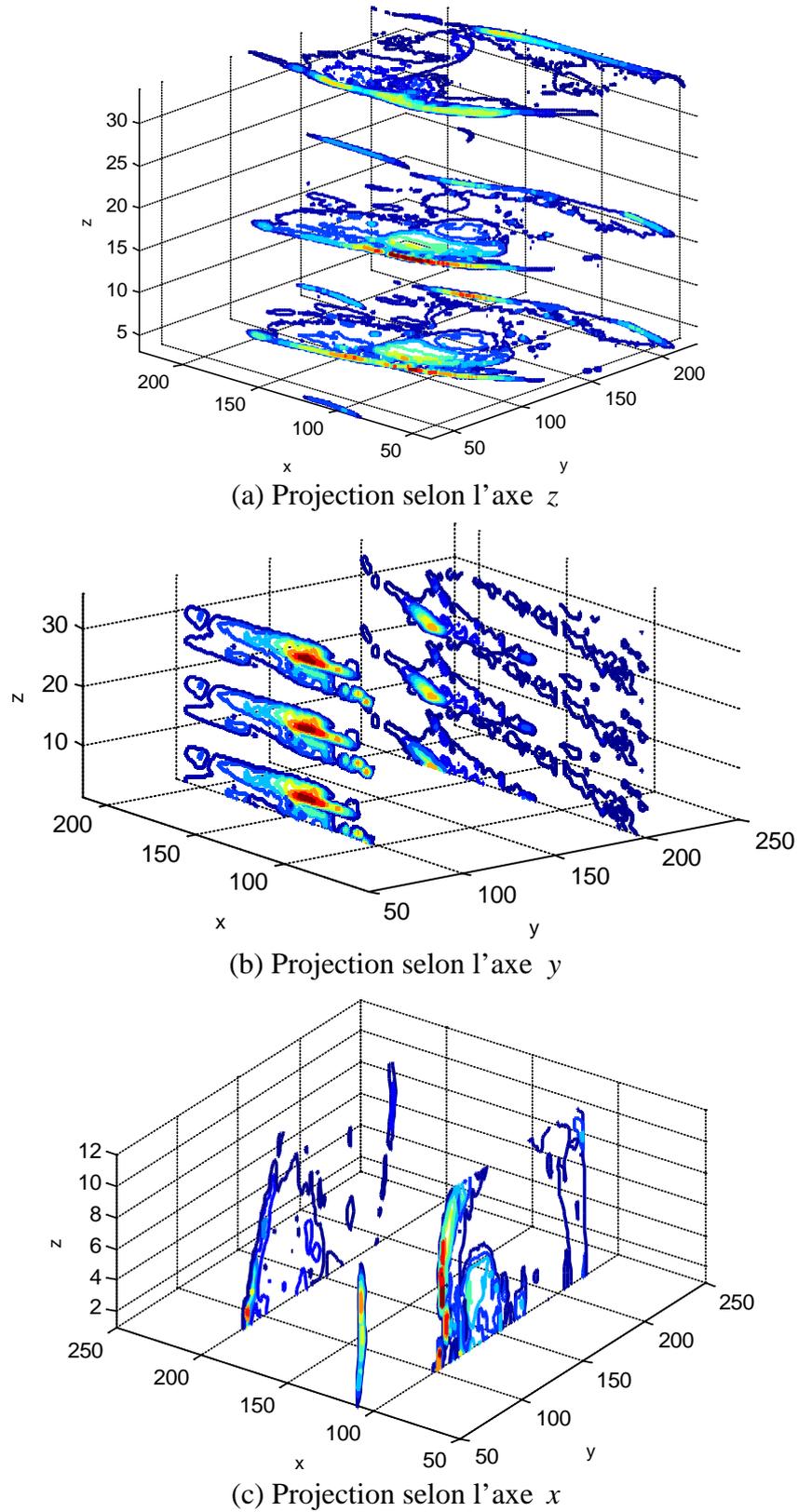


Figure 5.10. Contours des projections selon les axes x , y et z .

5.5.2. Dérivées spatio-temporelles

Le calcul du flot optique par les méthodes différentielles est basé sur une relation différentielle. Il est donc nécessaire de calculer les différentes dérivées présentées dans les relations (5.5) et (5.13).

Il existe plusieurs méthodes d'estimation du gradient dont les plus couramment utilisées dans le cas des séquences 3D sont [93] :

$$I_x = M_x * (I(p, t) + I(p, t + 1))$$

$$I_y = M_y * (I(p, t) + I(p, t + 1))$$

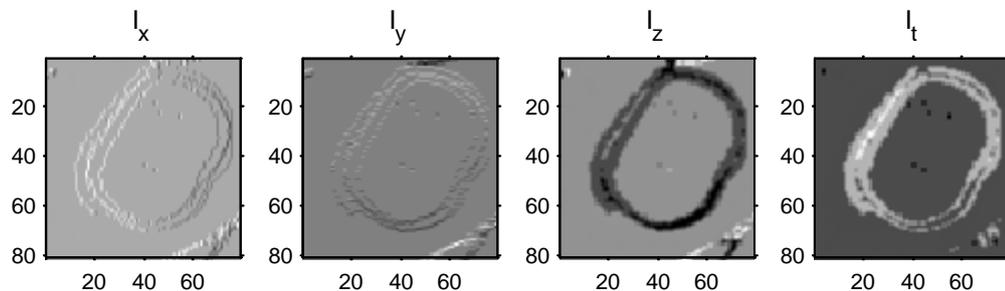
$$I_z = M_z * (I(p, t) + I(p, t + 1))$$

$$I_t = M_t * (I(p, t + 1) - I(p, t))$$

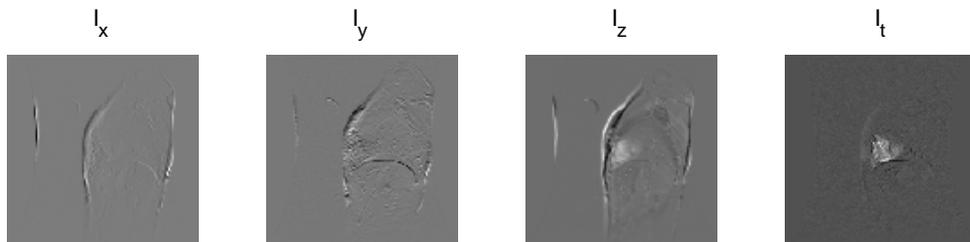
$$\text{avec } M_x = \frac{1}{8} \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}, M_y = \frac{1}{8} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}, M_z = \frac{1}{8} \begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\text{et } M_t = \frac{1}{8} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

La figure (5.11) présente les dérivées spatiotemporelles des deux séquences utilisées.



(a) Séquence synthétique



(a) Séquence réelle

Figure 5.11. Dérivées spatiotemporelles

5.5.3. Évaluation du flot optique estimé

5.5.3.1. Séquence synthétique

Dans cette section, nous présentons les résultats des deux méthodes pour la séquence synthétique. Pour la méthode de Lucas & Kanade modifiée LKM3D, le réseau de Zhang est utilisé pour résoudre le système d'équations pour chaque voxel. La figure (5.12) présente les courbes de convergence du réseau DZNN2. Nous pouvons noter que la quasi-totalité des courbes convergent en 2 itérations.

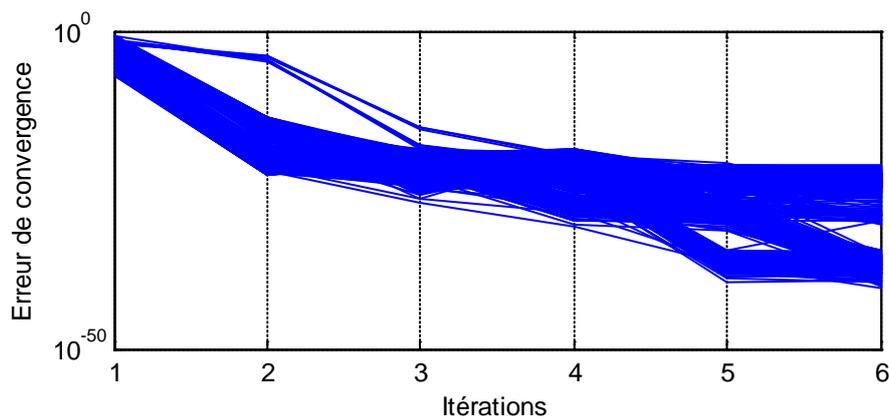


Figure 5.12. Courbes de convergence du DZNN2-LKM3D pour la séquence synthétique.

La figure 5.13 montre le flot optique tridimensionnel estimé par la méthode de Lucas & Kanade modifiée LKM3D et les figure (5.14) et (5.15) présentent les trois projections selon les axes z , y et x .

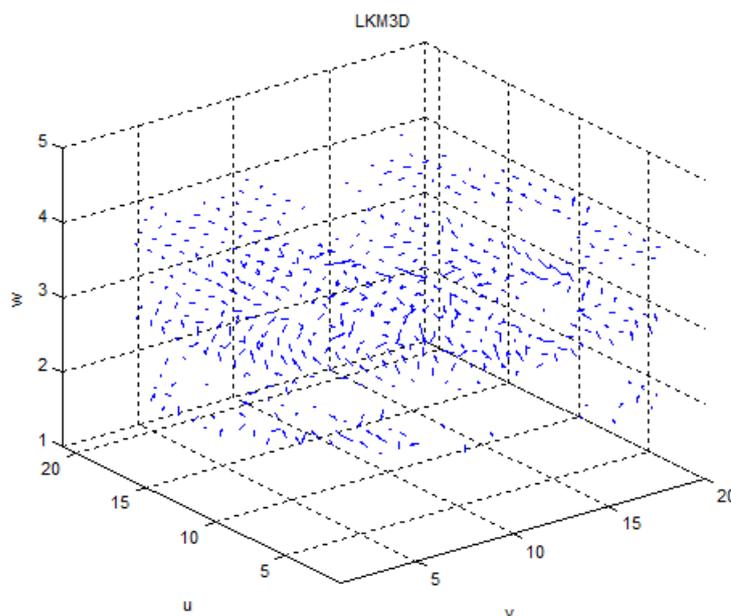


Figure 5.13. Flot optique 3D estimé par la méthode de LKM3D pour la séquence synthétique.

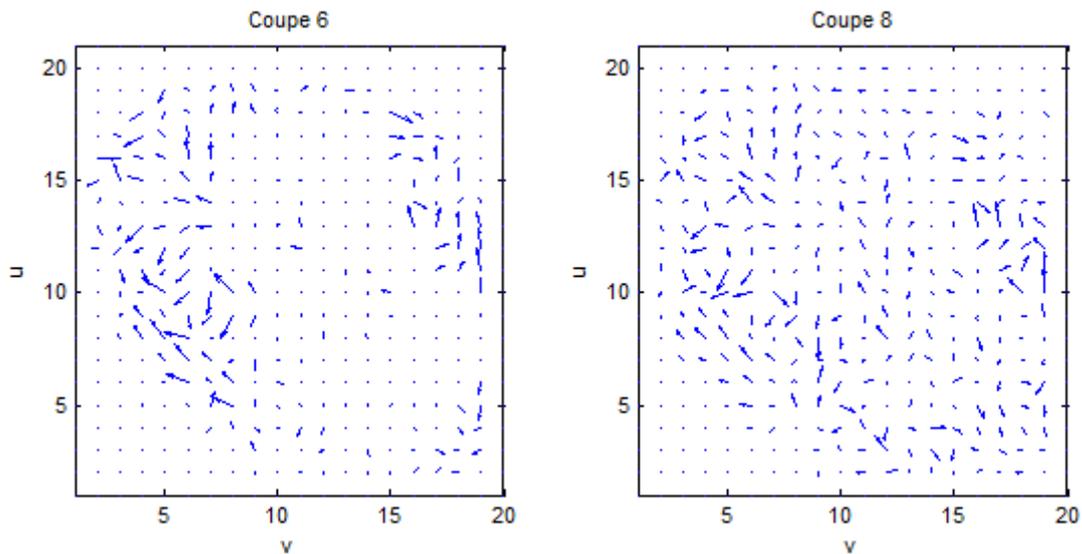


Figure 5.14. Flot optique estimé des coupes horizontales 6 et 8 par la méthode de LKM3D pour la séquence synthétique

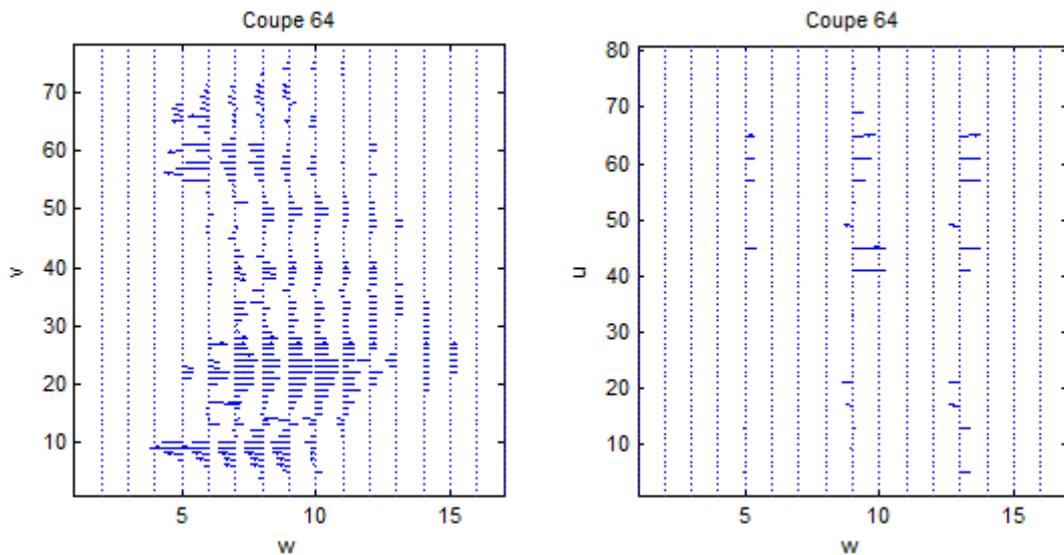


Figure 5.15. Projection du flot optique estimé par la méthode de LKM3D selon l'axe x et l'axe y pour la séquence synthétique.

Pour la méthode de Horn & Schunck modifiée HSM3D, le réseau de Zhang est utilisé pour résoudre le système d'équations pour tous les voxels. La figure 5.16 présente la courbe de convergence du réseau DZNN2. Nous remarquons que le réseau converge en 2 itérations. La figure (5.17) montre le flot optique tridimensionnel estimé par la méthode de HSM3D et les figure (5.18) et (5.19) présentent les trois projections selon les axes z , y et x .

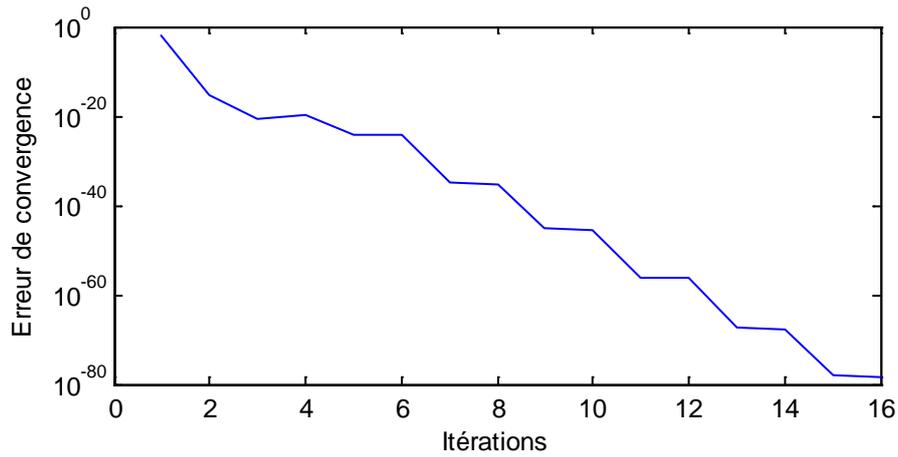


Figure 5.16. Courbes de convergence du DZNN2-HSM3D pour la séquence synthétique.

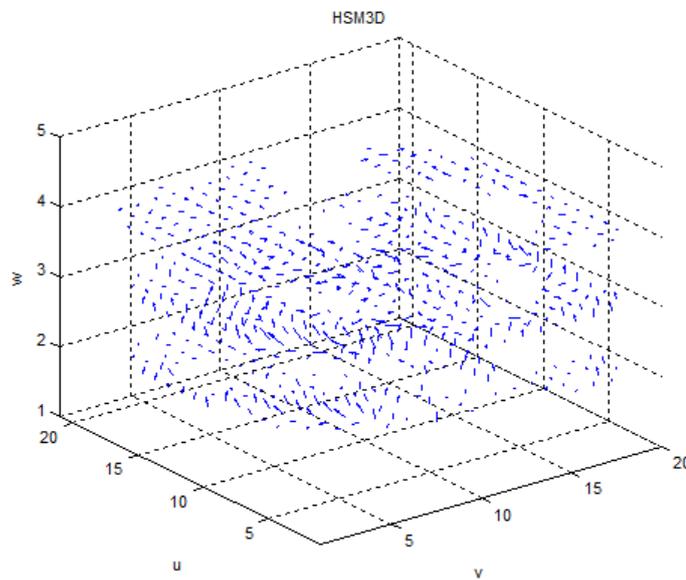


Figure 5.17. Flot optique 3D estimé par la méthode de HSM3D pour la séquence synthétique.

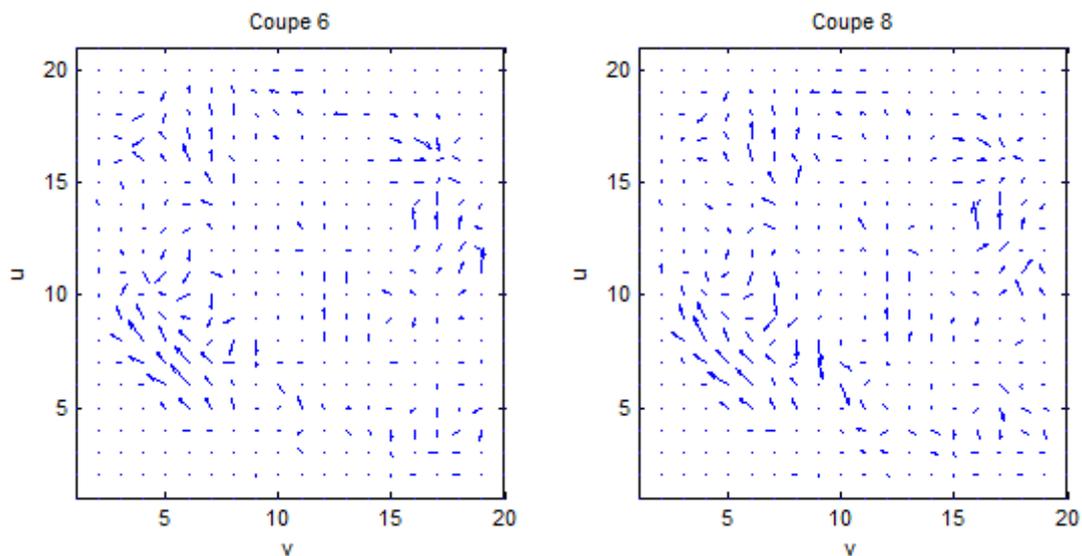


Figure 5.18. Flot optique estimé des coupes horizontales 6 et 8 par la méthode de HSM3D pour la séquence synthétique

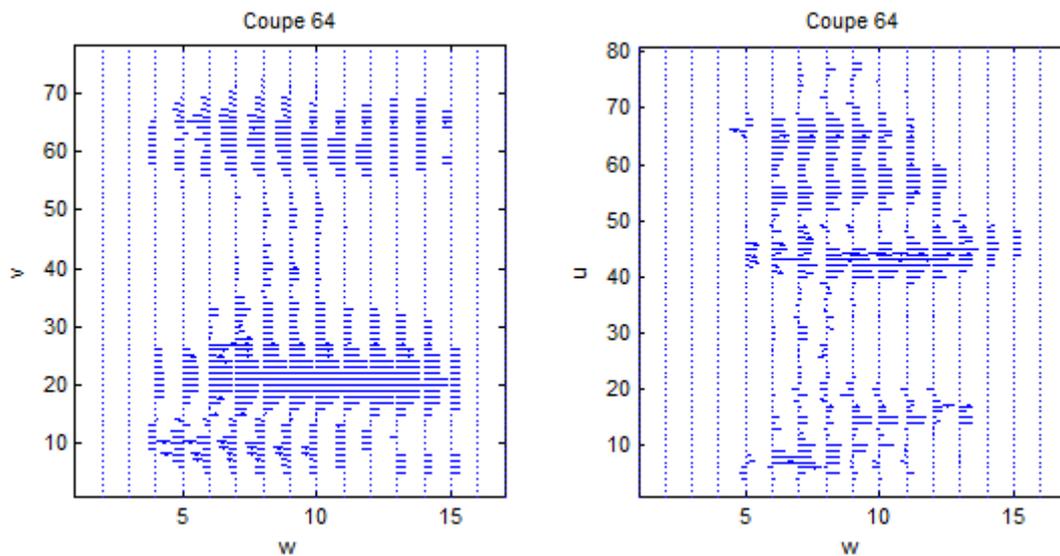


Figure 5.19. Projection du flot optique estimé par la méthode de HSM3D selon l'axe x et l'axe y pour la séquence synthétique.

D'après ses résultats nous pouvons remarquer que :

Le réseau DZNN2 converge rapidement pour les deux méthodes proposées.

La méthode HSM3D estime mieux le flot optique que la méthode LKM3D.

5.5.3.2. Séquence réelle

Dans cette section, nous présentons les résultats des deux méthodes pour la séquence volumique réelle. Nous suivons les mêmes démarches que celles de la section précédente.

La figure (5.20) [(5.24)] présente les courbes de convergence du réseau DZNN2-LKM3D (DZNN2-HSM3D) pour la méthode LKM3D (HSM3D) de la séquence réelle.

La figure (5.21) [(5.25)] montre le flot optique tridimensionnel estimé par la méthode de LKM3D (HSM3D).

La figure (5.22) [(5.26)] présente le flot optique estimé des coupes 3 et 10 horizontales par la méthode de LKM3D (HSM3D).

La figure (5.23) [(5.27)] présente la projection du flot estimé par la méthode de LKM3D (HSM3D) selon l'axe x et l'axe y pour la séquence réelle.

D'après ces résultats du flot optique sur la séquence réelle, nous remarquons que le réseau DZNN2 converge rapidement pour les deux méthodes proposées.

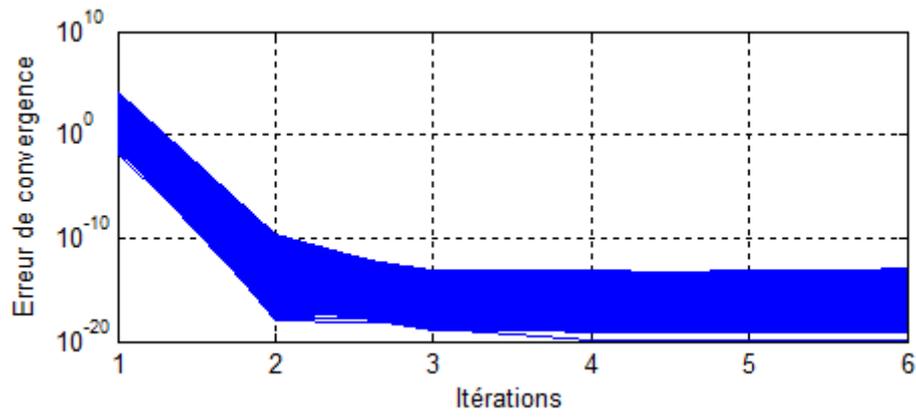


Figure 5.20. Courbes de convergence du DZNN2-LKM3D pour la séquence réelle

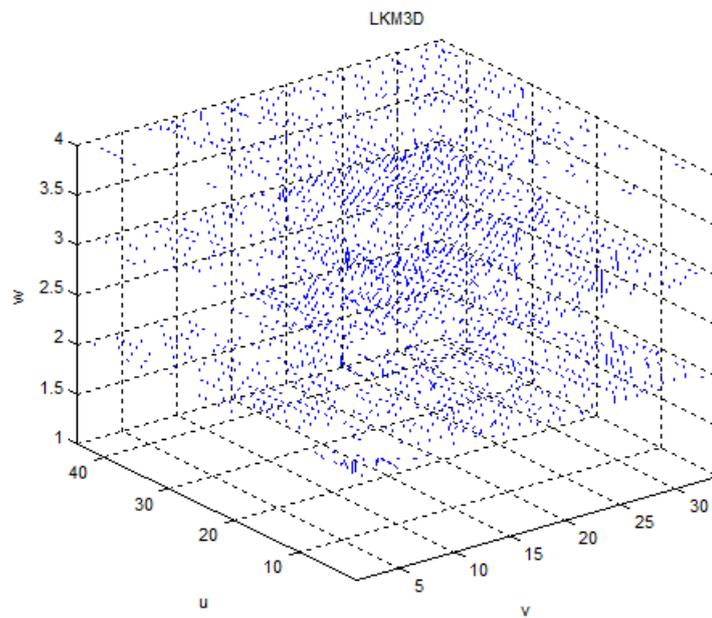


Figure 5.21. Flot optique 3D estimé par la méthode de LKM3D pour la séquence réelle.

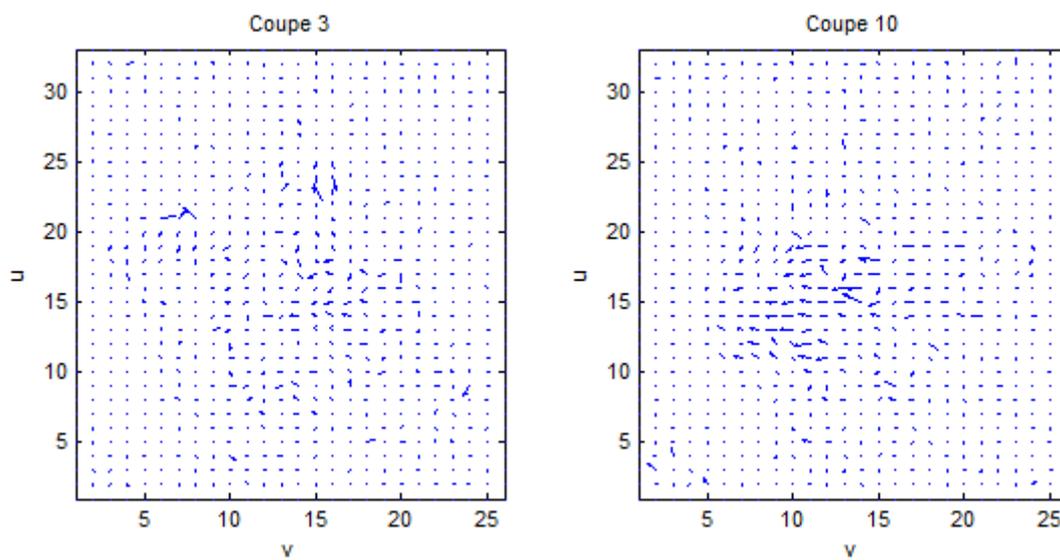


Figure 5.22. Flot optique estimé des coupes horizontales 3 et 10 par la méthode de LKM3D pour la séquence réelle

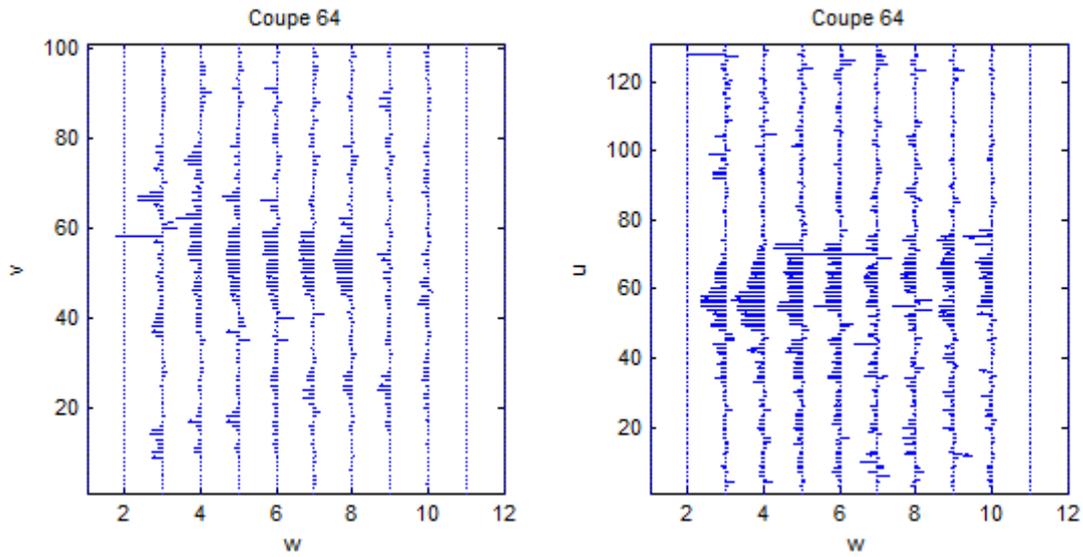


Figure 5.23. Projection selon x et y du flot optique estimé par la méthode de LKM3D pour la séquence réelle.

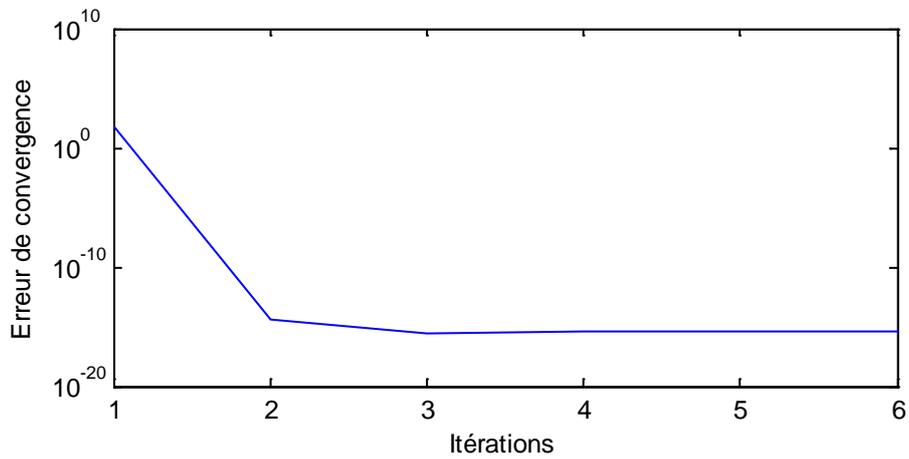


Figure 5.24. Courbes de convergence du DZNN2-HSM3D pour la séquence réelle

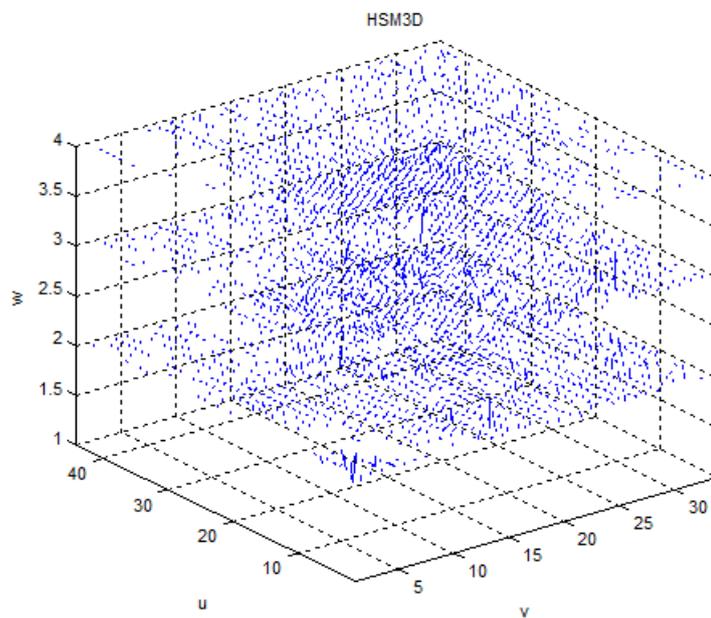


Figure 5.25. Flot optique 3D estimé par la méthode de HSM3D pour la séquence réelle.

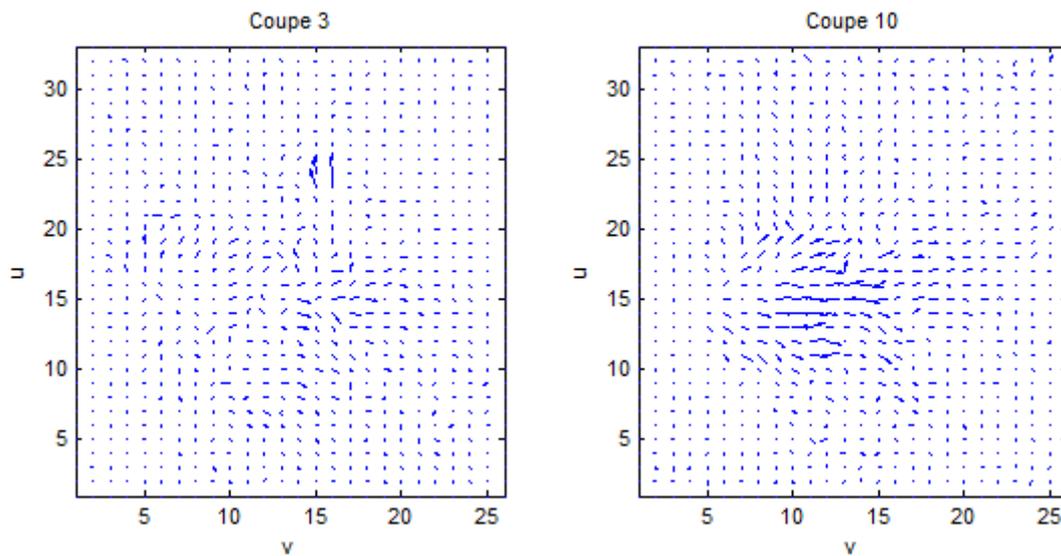


Figure 5.26. Flot optique estimé des coupes horizontales 3 et 10 par la méthode de HSM3D pour la séquence réelle.

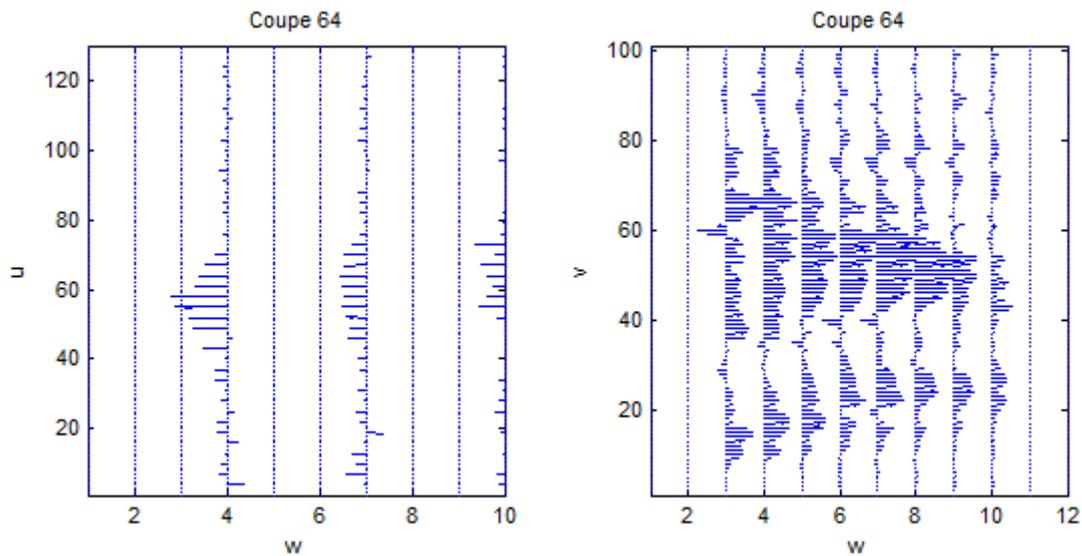


Figure 5.27. Projection du flot optique estimé par la méthode de HSM3D selon l'axe x et l'axe y pour la séquence réelle.

Suite à la non-disponibilité du flot réel, nous avons utilisé le procédé de la reconstruction inverse utilisant l'interpolation bilinéaire pour évaluer la qualité du champ estimé. Le critère PSNR entre l'image réelle et l'image compensée a été utilisé. La figure 5.28 montre le PSNR des deux méthodes pour différentes coupes. Nous pouvons remarquer que la méthode de Horn & Schunck modifiée est légèrement meilleure que la méthode de Luca & Kanade pour les séquences synthétiques et réelles.

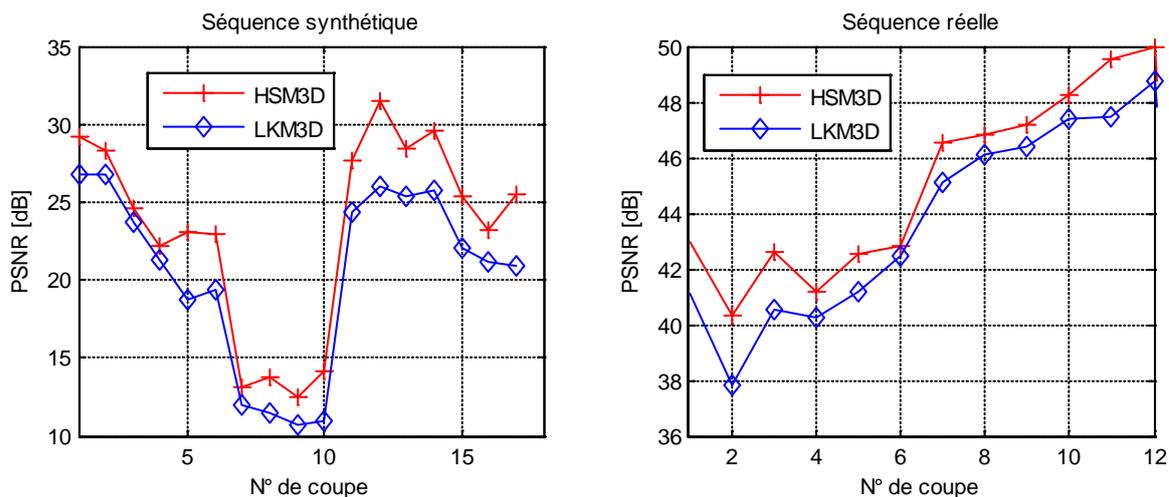


Figure 5.28. Évaluation du PSNR pour différentes coupes des deux méthodes proposées.

5.6. Conclusion

Généralement, il est admis que la méthode globale de Horn & Schunck est plus robuste et fournit des résultats de meilleure qualité que l'approche locale de Lucas & Kanade aux prix d'un coût de calcul plus important.

Dans ce chapitre nous avons proposé une nouvelle formulation de l'extension de la méthode de Horn & Schunck pour l'estimation du mouvement dans les séquences tridimensionnelles. Cette méthode a été significativement accélérée par l'introduction du réseau de Zhang pour le calcul du flot. De plus, nous avons montré que deux itérations sont largement suffisantes pour avoir une erreur de convergence presque nulle avec la conservation de la précision de la méthode. Ce réseau a été également utilisé dans les deux méthodes tridimensionnelles de Lucas & Kanade et Horn & Schunck.

Conclusion générale

L'objectif fixé au début de cette thèse était d'élaborer des méthodes d'estimation de mouvement précises, robustes et rapides. L'estimation de mouvements représente un intérêt majeur en vision par ordinateur étant donné le champ des applications possibles; contrôle du trafic routier, guidage de robots, compression de séquences d'images, diagnostic médical Dans cette thèse nous avons contribué à l'amélioration des méthodes différentielles existantes (par régularisation globale ou locale) en introduisant deux nouveaux outils ; la transformée en curvelettes et le réseau de neurones récurrents.

Principales contributions

Notre première contribution a consisté à améliorer les performances de la méthode itérative de Lucas & Kanade en introduisant les outils suivants:

- La reformulation de la méthode pour l'estimation simultanée du mouvement de tous les pixels en utilisant deux types de réseaux de neurones récurrents ; le réseau de Hopfield et le réseau de Zhang.
- L'introduction d'une transformation géométrique récente pour l'estimation de mouvement multi-échelles au lieu de la pyramide Laplacienne.
- Le prétraitement des images utilisant les transformations géométriques.

La seconde contribution a consisté de proposer une implémentation parallèle via les réseaux de neurones de la méthode de Horn & Schunk dans sa reformulation proposée par Mitiche & Mansouri donnant naissance à un système large d'équations [19]. Trois versions du réseau de Zhang ont été proposées pour résoudre ce système :

- Une version à dynamique explicite pour inverser la matrice de données (DZNN1-1).
- Une version à dynamique implicite pour inverser la matrice de données (DZNN1-2).
- Une version à dynamique implicite pour résoudre le système d'équations (DZNN2).

Les résultats obtenus sur des séquences synthétiques et réelles montrent que les méthodes proposées nous permettent une nette amélioration de point de vue précision et rapidité. Cette amélioration se résume dans les points suivants :

- Les deux réseaux de Zhang DZNN2 et DZNN1-2 convergent en seulement deux itérations sans altérer la précision de l'estimation.
- Le réseau de Hopfield bien qu'il nécessite un grand nombre d'itérations, il est moins sensible au conditionnement de la matrice.

- L'estimation de mouvement après débruitage des images par la transformée en curvelettes présente une erreur angulaire inférieure à celle obtenue sans débruitage ou après un débruitage par ondelettes.
- L'estimation de mouvement multi-échelles par la transformée en curvelettes présente des bons résultats comparables aux méthodes récentes.

En définitive, les résultats étant convaincants et d'un intérêt scientifique certain. Ils nous permettent de réaliser les publications et les communications suivantes :

- F. Charif, N. Djedi, A. Benchabane, “ On Parallel Implementation of Horn and Schunk Motion estimation Method”, International journal of computer and applications. vol. 35,no.2, pp:79-85,2013.
- F. Charif, N. Djedi, A. Taleb-Ahmed, A. Benchabane, “A simplified Zhang neural network for solving a large Toeplitz linear systems and its application for AR parameters estimation”. *Pensee journal*. vol. 76, no.4,pp.354-360,2014.
- F. Charif, N. Djedi, A. Benchabane., A. Bennia, A. Taleb-Ahmed, “ Estimation Rapide du Champs de Mouvement par la Méthode Itérative Multi-échelles de Lucas & Kanade”. SETIT 2012, Tunisie.
- F. Charif , A. Benchabane, Djedi N., Taleb-Ahmed A. “ Horn & Schunk meets a discrete Zhang neural network for computing 2D optical flow”. ICEO13, Ouargla, Algérie.
- F. Charif, N. Djedi, A. Benchabane, A. Taleb-Ahmed, “ Estimation Neuronale du Flot Optique dans une Séquence d'Images Débruitées par la Transformée en Curvelettes”, ICA2IT , Ouargla, Algérie .Mars 2014.

Perspectives

La réussite des outils proposés nous permet d'ouvrir les perspectives suivantes :

- Introduire le prétraitement par la transformée en curvelettes dans des méthodes présentées récemment comme par exemple [96-97].
- Introduire la transformée en curvelettes tridimensionnels aux méthodes d'estimation de mouvement 3D proposées.

Annexe A

Aperçu sur les réseaux de neurones artificiels

A.1. Introduction

Inspirés du fonctionnement du cerveau humain, les réseaux de neurones artificiels (RNA) occupent aujourd'hui une place prépondérante dans plusieurs domaines des sciences de l'ingénieur [15,98]. Le fonctionnement des systèmes de neurones, caractérisé par le principe de parallélisme intrinsèque, permet d'exécuter un grand nombre de calculs dans un temps limité et offre une alternative intéressante pour résoudre des problèmes complexes.

A.2. Neurone formel

Le neurone formel est une modélisation mathématique qui reprend les principes du fonctionnement du neurone biologique. C'est une est une fonction paramétrée, non linéaire, de plusieurs variables d'entrée appelées entrées du neurone; la valeur de la fonction est disponible en sortie du neurone [15,98]. La figure (A.1) montre le modèle du neurone artificiel.

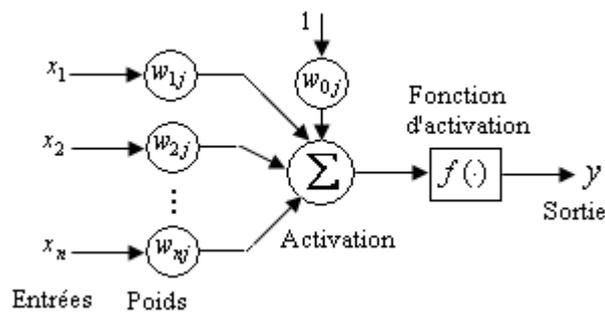


Figure A.1. Modèle de neurone formel

Pour un neurone j , le calcul de la fonction est effectué en deux étapes; le calcul d'une somme pondérée par les poids synaptiques $(w_{1j}, w_{2j}, \dots, w_{nj})$ de ses entrées (x_1, x_2, \dots, x_n) . Le résultat obtenu est appelé le potentiel du neurone. Il est donné par la relation suivante :

$$v_j = \sum_{i=1}^n w_{ij} x_i - w_{0j} \quad (\text{A.1})$$

avec w_{0j} est un paramètre de seuil.

Puis le calcul d'une fonction non linéaire du potentiel, souvent appelée fonction d'activation est effectué. La sortie du neurone est alors la valeur de cette fonction :

$$y = f(v_j) \quad (\text{A.2})$$

A.3. Réseaux de neurones artificiels

Un réseau de neurones artificiels est un ensemble de neurones formels associés en couches et fonctionnent en parallèle. Chaque couche fait un traitement indépendant des autres et transmet le résultat de son analyse à la couche suivante. L'information donnée au réseau va donc se propager de la couche d'entrée à la couche de sortie, en passant au choix, par aucune, une ou plusieurs couches intermédiaires appelées couches cachées. La figure (A.2) montre un réseau de neurone mono-couche.

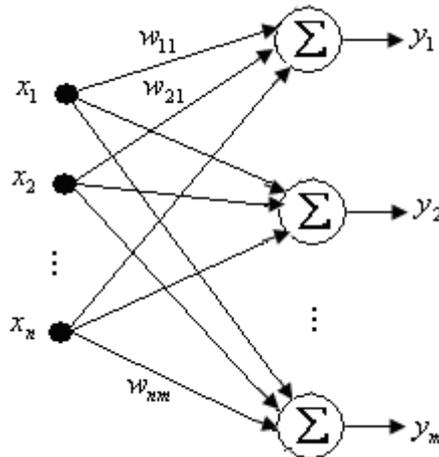


Figure A.2. Schéma d'un réseau de neurones

A.4. Principaux types des réseaux de neurones

Il existe une multitude de réseaux de neurones différents les uns des autres par leurs architectures et leurs méthodes d'apprentissages. Ils peuvent être classés en deux types ; les réseaux non récurrents et les réseaux récurrents. La Figure (A.3) présente les réseaux de neurones les plus utilisés dans la littérature.

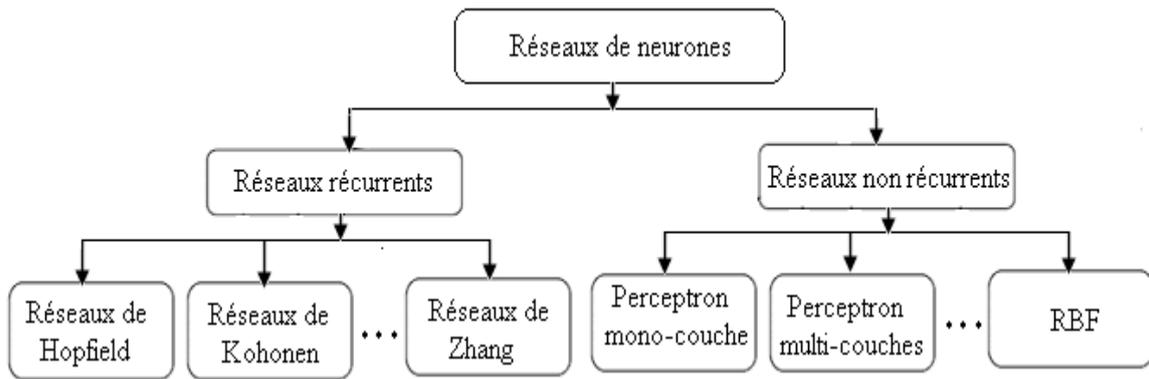


Figure A.3. Réseaux de neurones les plus utilisés.

A.5. Réseaux de neurones récurrents pour l'optimisation

Le fonctionnement des systèmes de neurones, caractérisé par le principe de parallélisme intrinsèque, permet d'exécuter un grand nombre de calculs dans un temps limité et offre une alternative intéressante pour résoudre le problème complexe d'optimisation, dont les réseaux de neurones de Hopfield est très appropriés.

A.6. Réseaux de neurones de Hopfield

Le réseau de Hopfield est un réseau dynamique mono-couche dont les neurones sont complètement connectés. La dynamique du réseau de Hopfield est donnée par [62] :

$$\begin{aligned} \tau \dot{y}_i(t) &= -y_i(t) + \sum_{j=1}^n w_{ij} u_j(t) + v_i \\ u_i(t) &= f(y_i(t)) \end{aligned} \quad (\text{A.3})$$

où $y_i(t)$ est l'état interne du neurone i , $u_i(t)$ est la sortie du neurone i , w_{ij} est le poids de connexion entre le neurone i et le neurone j , $v_i(t)$ est l'entrée du neurone i et τ est le taux de convergence. Le réseau de Hopfield peut être vu comme un système à dynamique non linéaire avec un vecteur d'entrée \mathbf{v} , un vecteur d'état $\mathbf{y}(t)$ et un vecteur de sortie $\mathbf{u}(t)$ comme montré dans la figure (A.4)

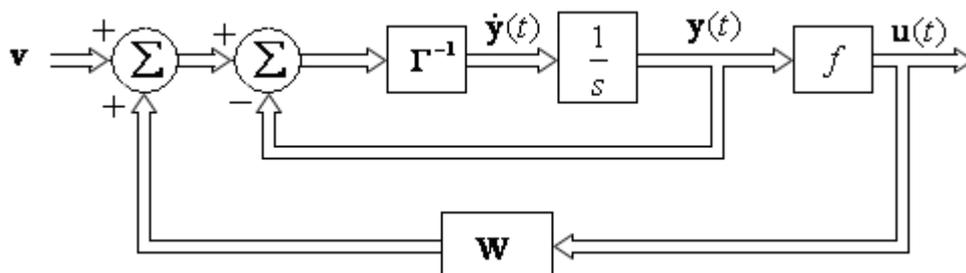


Figure A.4. Structure du réseau de Hopfield

Hopfield a utilisé une fonction d'énergie associée au réseau comme outil pour définir leur dynamique [62]. Cette fonction d'énergie est définie par Hopfield comme suit :

$$E(t) = -\frac{1}{2} \mathbf{u}(t)^T \mathbf{W} \mathbf{u}(t) - \mathbf{u}(t)^T \mathbf{v} \quad (\text{A.4})$$

A.7. Principes de fonctionnement du réseau de Hopfield pour l'optimisation

En 1985, Hopfield et Tank [62] ont appliqué ce type de réseaux de neurones à l'optimisation de la manière suivante: à partir d'un état initial, on laisse le réseau évoluer librement jusqu'à un attracteur, qui est généralement, pour les problèmes d'optimisation, un état stable indépendant du temps. On dit alors que le réseau a convergé : la convergence est atteinte lorsque les sorties des neurones n'évoluent plus. Les poids des connexions sont déterminés analytiquement à partir de la formulation du problème; cela est fait directement à partir de la fonction de coût associée au problème. De plus, les sorties des neurones, dans l'attracteur vers lequel converge le réseau, codent une solution au problème d'optimisation.

A.8. Conclusion

Nous avons présenté dans cette annexe un aperçu général sur les réseaux de neurones. Au début, nous avons présenté le neurone formel et son comportement. Puis on a donné un plus de détail sur le réseau de Hopfield et son utilisation dans les problèmes de l'optimisation.

Annexe B

Gradient de Simoncelli

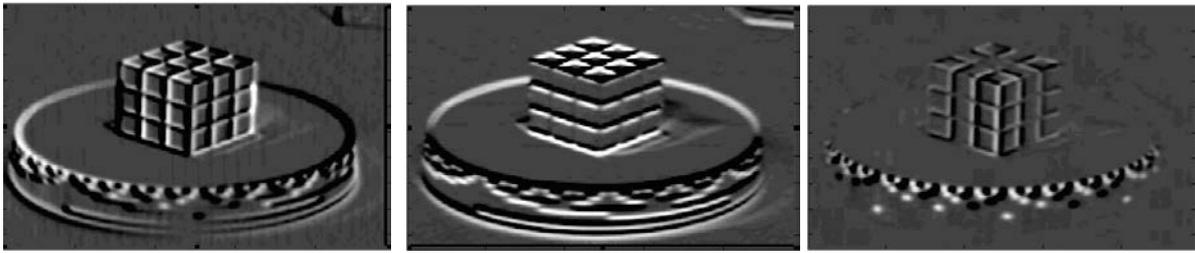
Simoncelli [99] s'intéresse à la représentation fréquentielle du signal pour estimer sa dérivée. Il utilise deux filtres différents. Le premier filtre (que l'on appellera p_5) consiste en un préfiltrage de l'image : c'est un filtre passe-bas. Le bruit de l'image étant principalement caractérisé par de légères variations brusques de la luminosité, ce filtre passe-bas atténue fortement ce bruit. De plus, l'opérateur de dérivation pose souvent des problèmes liés aux fortes variations de signal, d'autant plus que nous considérons des valeurs discrètes. Le préfiltrage aura pour conséquence de lisser l'image et d'atténuer ces fortes variations. Le second filtre (que l'on appellera d_5) permet de calculer la dérivée. Simoncelli a calculé ces filtres de manière à ce qu'ils répondent notamment aux propriétés suivantes :

1. Le préfiltre p_5 sur deux dimensions doit être séparable en deux pré-filtres sur une dimension, ce qui implique la même propriété sur d_5 .
2. Le préfiltre doit être symétrique, le filtre de dérivation est anti-symétrique.

La première de ces propriétés rend le calcul de la dérivée sur x indépendant de la dérivée sur y . Ces deux filtres sont structurellement différents des filtres de Sobel puisque Simoncelli propose des masques de convolutions sous forme de vecteurs. Cette configuration de masque est permise puisque les filtres sont séparables. La figure B.1 représente les résultats de l'estimation des dérivées suivant x et y . Ces résultats ont été obtenus avec les masques des filtres p_5 et d_5 définis dans la table B.1. Le calcul se fait sur sept images successives [100]. Avec la méthode de Simoncelli, les dérivées sur x et y paraissent beaucoup plus floues. Ceci est dû à la pondération sur le voisinage d'un maximum de cinq pixels.

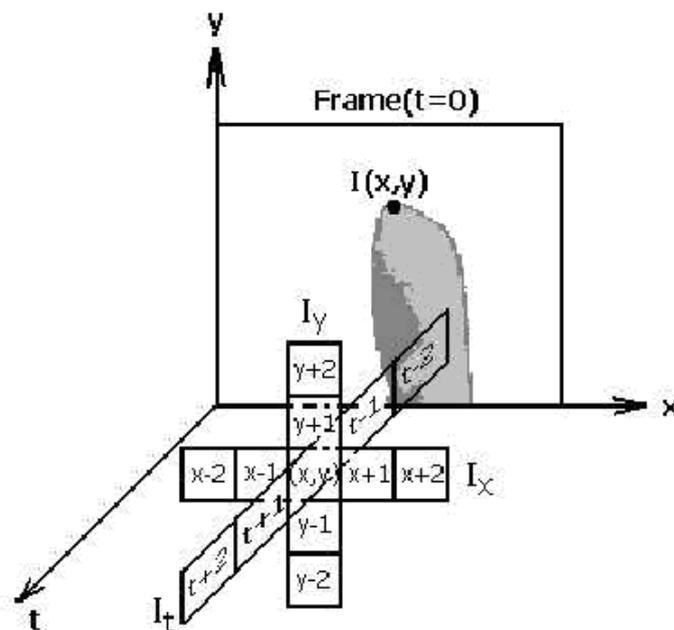
Filtre/ n	-2	-1	0	1	2
p_5	0.036	0.249	0.431	0.249	0.036
d_5	-0.108	-0.283	0.0	0.283	0.108

Tableau. B.1 – Masques proposés par Simoncelli dans [99]

(a) suivant x (b) suivant y (c) suivant t **Figure. B.1.** Gradient de Simoncelli

La dérivée temporelle laisse bien apparaître le mouvement des objets dans l'image. Le filtre de Simoncelli a l'avantage de posséder en entrée un filtre passe-bas qui diminue la composante haute fréquence du bruit. Ainsi les dérivées estimées se trouvent moins bruitées et plus lisses. La méthode de Simoncelli considère un voisinage plus important puisque qu'elle consiste à appliquer successivement deux filtres sous forme de vecteurs de cinq éléments. Ainsi, nous avons besoin de sept images pour calculer la dérivée temporelle à un instant précis (Fig. B.2)

Le filtre de dérivation d_5 nécessite cinq points pour être appliqué ; ces cinq points sont issus du filtrage passe-bas p_5 . Le filtrage passe-bas se calcule lui même sur cinq points. De même pour les dérivées spatiales, l'estimation s'effectue sur sept points de l'image originale, donc un voisinage comparable à quatre pixels. Par exemple, I_y est calculé en appliquant p_5 suivant t , puis en appliquant p_5 à ces résultats suivant x et finalement en appliquant d_5 à ces résultats suivant y . I_x et I_y sont calculés d'une façon appropriée et semblable.

**Figure. B.2.** Calcul des dérivées partielles de Simoncelli

Bibliographie

- [1] E. Memin, “Estimation du flot-optique: contributions et panorama de différentes approches”, Habilitation à diriger des recherches de l'Université de Rennes 1, 2003.
- [2] M. R. Fezzani, “Approche parallèle pour l’estimation du flot optique par méthode variationnelle”, thèse de doctorat, Université de Lille 1, 2000.
- [3] B. K. P. Horn, B. G. Schunck, “Determining optical flow”, *Artificial Intelligence*, vol. 17, pp.185–204, 1981.
- [4] J. L. Barron, D. J. Fleet, S. S. Beauchemin, T. A Burkitt, “Performance of Optical Flow Techniques”, *Department of Computer Science, University of Western Ontario*, July 1993.
- [5] S. Ambellouis, “Analyse du mouvement dans les séquences d’images par une méthode récursive de filtrage spatio-temporel sélectif ”, Thèse de doctorat, Université Pierre et Marie Curie, 2011.
- [6] E. Bruno, “De l’estimation locale à l’estimation globale de mouvement dans les séquences d’images ”, Thèse de doctorat, Université Joseph Fourier novembre 2001.
- [7] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” *In Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.
- [8] S. Baker, I. Matthews, “ Lucas-Kanade 20 Years On: A Unifying Framework ”, *Tech. report CMU-RI-TR-02-16, Robotics Institute, Carnegie Mellon University*, July, 2002.
- [9] T. Amiaza, E. Lubetzkyb, N. Kiryatia, “Coarse to over-fine optical flow estimation”, *Pattern Recognition* , vol.40,no. 9, pp. 2496-2503, 2007.
- [10] J.-Y. Bouguet, “Pyramidal Implementation of the Lucas Kanade Feature Tracker”, *Intel Corporation, Microprocessor Research Labs*, 2000.
- [11] S. Mallat. *Une exploration des signaux en ondelettes*. Les éditions de l’école Polytechnique, 2000.
- [12] E. Candès. “Ridgelets: Theory and Applications”, Thèse de Doctorat, Département of Statistics , Stanford University, 1998.
- [13] E. Candès and D. Donoho. “Curvelets: A surprisingly effective non adaptive representation of objects with edges”, *In Saint-Malo Proceedings*, pp. 1-10. Vanderbilt University Press, 2000.

- [14] J.-L. Starck, E.J. Candès, and D.L. Donoho. “The curvelet transform for image denoising”. *IEEE Trans. on Image Processing*, vol.11, no.6, pp. 131–141, 2002.
- [15] Y. H. Hu, J. N. Hwang, *Handbook of neural network signal processing*, CRC PRESS, 2002.
- [16] Y. Zhang, “Revisit the analog computer and gradient-based neural system for matrix inversion”, *In Int. sympo. on intelligent control, IEEE ISIC 2005*, pp.1411–1416, 2005.
- [17] C. Grava, A. Gacsádi, I. Buciu , “ A Homogeneous Algorithm for Motion Estimation and Compensation by Using Cellular Neural Networks” , *Int. J. of Computers, Communications & Control*, vol. 5, no. 5, pp. 719-726. 2010.
- [18] H. Li, J. Wang, “Computing optical flow with a recurrent neural network”, *Int. J. Pattern Recognit. Artif. Intell.*, vol.7 , no.4 ,pp.801-814,1993.
- [19] A. Mitiche, A. R. Mansouri, “On convergence of the Horn and Schunck optical-flow computation”, *IEEE, Image Processing*, 13, pp. 848-852, 2004.
- [20] S. Beauchemin, J. Barron, “The computation of optical flow”, *ACM Computing Surveys*, vol.27, pp. 433-467, 1995.
- [21] A. Bruhn, J. Weickert, “Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Method”, *IJVC*, vol. 61, no. 3, pp. 211-231.2005.
- [22] Q. Tang, B. Chiang, A. Akinyemi ; A. Zamyatin, B. Shi, S. Nakanishi , “ A combined local and global motion estimation and compensation method for cardiac CT ”, *Proc. SPIE 9033, Medical Imaging 2014: Physics of Medical Imaging, 903304* ,March 2014.
- [23] D. J. Fleet, A. D. Jepson, “Computation of Component Image Velocity from Local Phase Information”, *IJCV*, vol.5, no.1, pp .77-104, 1990.
- [24] D. J. Fleet, “Disparity from Local Weighted Phase- Correlation”, *IEEE Int. Conf. on Systems, Man and Cybernetics*, vol.1, pp. 48-54, 1994.
- [25] V. Argyriou and T. Vlachos, “A study of sub-pixel motion estimation using phase correlation”, *Centre for Vision, Speech and Signal Processing University of Surrey*, 2004
- [26] J.R. Bergen, P. Anandan, K.J. Hanna, and R.Hingorani, “Hierarchical Model-Based Motion Estimation”, *ECCV 1992*, pp. 237-252.1992.
- [27] A. Spinei, D. Pellerin, J. Herault, “ Estimation rapide de vitesse a base de triades de filtres de Gabor”, *16eme colloque GRETSI*, pp 905-908, 15-19 septembre 1997.
- [28] J. Weber and J. Malik, “Robust computation of optical flow in a multi-scale differential framework,” *Int. J. Comput. Vision*, vol.14, pp.67–81 .1995.

- [29] C. Bernard “Ondelettes et problèmes mal posés : la mesure du flot optique et l’interpolation irrégulière”, Thèse Centre de Mathématiques Appliquées, Ecole Polytechnique, 1999.
- [30] Y.T Wu, T. Kanade, J. Cohn, C-C. Li, “ Optical Flow Estimation Using Wavelet Motion Model”, *ICCV 98* ,pp 992-998, 1998
- [31] J. Magarey, N. Kingsbury, “Motion Estimation Using a Complex-Valued Wavelet Transform ”,*IEEE Trans. on signal processing*, vol. 46,no. 4, April 1998.
- [32] E. Bruno et D. Pellerin. “Global motion model based on B-spline wavelets: application to motion estimation and video indexing”, *In: The 2nd Int. Symp. on Image and Signal Processing and Analysis ISPA'01* , pp. 289-294.2001.
- [33] L. F. Chen, H. Y. M. Liao, J. C. Lin, “Wavelet-Based optical flow estimation” ,*IEEE Trans. on circuits and systems for video technology* , vol.12, pp 1–12. 2002
- [34] R. Szeliski and J. Coughlan. “Spline-based image registration”, *Int. J. of Computer Vision*, vol.22, no.3, pp. 199–218, 1997.
- [35] W. Li and E. Salari, “Successive Elimination Algorithm for Motion Estimation”, *IEEE Trans. on image processing*, vol. 4, no. 1, january 1995.
- [36] X. Jing, L. P Chaud, “An Efficient Three-Step Search Algorithm for Block Motion Estimation” ,*IEEE Transactions on multimedia*, vol 6, no. 3, june 2004.
- [37] Y. Nie, K. K. Ma, “Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation”, *IEEE Trans. on image processing* , vol.11, no. 12, december 2002.
- [38] F. Yu and A.N. Willson, Jr., “A flexible hardware-oriented fast algorithm for motion estimation”, *ICASSP97*, pp.268-288, 1997.
- [39] S. Zhu, K. Ma, “A new diamond search algorithm for fast block matching”, *IEEE Trans. on Circuits and Systems on Video Technology*, vol.9, no.2, pp287-290.2000.
- [40] X. L. S. Zhu and L. Chau, “Hexagon-based search pattern for fast block motion estimation”, *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 12, May 2002.
- [41] J. Gilles, “Décomposition et détection de structures géométriques en imagerie”, Thèse de doctorat, L'école normale supérieure de Cachan.2006.
- [42] E. Candès and D. Donoho. “Continuous curvelet transform, part II: Discretization and frames”, *Rapport technique*, Department of Statistics,Stanford University, 2003.
- [43] D. Donoho and M. Duncan. “Digital curvelet transform : strategy, implementation and experiments”. *Rapport technique*, Department of Statistics, Stanford University, 1999.

- [44] M. N. Do and M. Vetterli. "Pyramidal directional filter banks and curvelets". *IEEE Int. Conf. on Image Processing (ICIP), Thessaloniki, Greece, 2001*.
- [45] M. N. Do. "Contourlets and sparse image representations". *SPIE conf. on Wavelet Applications in Signal and Image Processing X, San Diego, USA, 2003*.
- [46] M. N. Do and M. Vetterli. "The contourlet transform : an efficient directional multiresolution image representation", *IEEE Trans. Image Processing*, vol.14, no.12, pp: 2091-2106, 2003.
- [47] D. D. Y. Po and M. N. Do. "Directional multiscale modeling of images using the contourlet transform". *IEEE Trans. Image Processing*, 2006.
- [48] S. Mallat. "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.11, no.7, pp. 674-693, 1989.
- [49] G. Lebrun, "Ondelettes géométriques adaptatives : vers une utilisation de la distance géodésique ", Thèse de doctorat , Université de Poitiers,2009.
- [50] J.-L. Starck, P. Abrial, Y. Moudden, and M. Nguyen, "Wavelets, ridgelets and curvelets on the sphere", *Astronomy and Astrophysics*, vol.446pp.1191–1204, 2006.
- [51] E. Le Pennec and S. Mallat. "Sparse geometrical image representation with bandelets", *IEEE Transactions on Image Processing*, vol.14,no.4, pp.423-438, 2005.
- [52] R. H. Bamberger and M.J.T. Smith. "A filter bank for the directional decomposition of images: Theory and design", *IEEE Trans. Signal Processing*, vol.40, no.4, pp .882-893, April 1992.
- [53] P.J. Burt and E.H. Adelson, "The Laplacian Pyramid as a Compact Image Code". *IEEE Trans. Commun.* , vol .31, no 4 ,pp. 532-540 ,1983.
- [54] D. Donoho, " De-noising by soft-thresholding", *IEEE Trans. on Information Theory* , vol.41, no. 3, pp. 613- 627,May 1995.
- [55] L. Breiman "Better Subset Regression Using the Nonnegative Garrote", *Technometrics*, vol. 37, no.4, pp.373-384. 1995.
- [56] A. Bruce and H. Y. Gao,"WaveShrink: Shrinkage Functions and Thresholds", *Proc. SPIE* , vol. 2569, pp. 270-281, 1995.
- [57] H.Y. Gao, "Wavelet Shrinkage Denoising Using the Non-Negative Garrote", *Journal of Computational and Graphical Statistics*, vol. 7, no.4, pp.469-488. 1998.
- [58] W. Fourati, M. S. Bouhlel , "Techniques de Débruitage d'Images", *5th Int. Conf. Sciences of Electronic, Technologies of Information and Telecommunications* , March 22-26, 2009

- [59] X.-P. Zhang, "Thresholding neural network for adaptive noise reduction", *IEEE Trans. Neural Networks*, vol.12, no.3, pp. 567–584.2001.
- [60] N. Mehdi, H. Nezamabadi "Image denoising in the wavelet domain using a new adaptive thresholding function neurocomputing", *Neurocomputing*, vol.72, pp.1012-1025. 2009.
- [61] S. N. Tamgade, V. R. Bora, "Motion vector estimation of video image by Pyramidal implementation of Lucas Kanade Optical flow," *Second Int. Conf. on Emerging Trends in Engineering and Technology, ICETET-09*, pp. 914-917, 2009.
- [62] J. Hofield, D. Tank, "Neural computation of decisions in optimization problems", *Biol. Cybernet.* vol.52, pp. 141-152, 1985.
- [63] D. Guo, C. Yi, Y. Zhang, "Zhang neural network versus gradient based neural network for time-varying linear matrix equation solving", *Neurocomputing*, vol. 74, no. 17, pp. 3708–3712, Oct. 2011.
- [64] C. Yi, Y. Zhang, "Analogue recurrent neural network for linear algebraic equation solving", *Electr. Lett.*, vol. 44, no. 18, pp. 1078-1079, August 2008.
- [65] Y. Zhang, Y. Shi, Y. Yang, Z.K, "Convergence Analysis of Zhang Neural Networks Solving Time-Varying Linear Equations but without Using Time-Derivative Information", *2010 8th IEEE Int. Conf. on Control and Automation Xiamen, China*, pp.1215-1220, June 2010.
- [66] F. Charif, N. Djedi, A. Benchabane, BENNIA A., TALEB-AHMED A, "Estimation Rapide du Champs de Mouvement par la Méthode Itérative Multi-échelles de Lucas & Kanade", *SETIT 2012, Tunisie.2012*.
- [67] E. H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, J.M. Ogden. "Pyramid methods in image processing". *RCA Engineer*, vol. 29, n6 ; pp. 33-41.1984.
- [68] F. Charif, N. Djedi, A. Benchabane, A. Taleb-Ahmed, " Estimation Neuronale du Flot Optique dans une Séquence d'Images Débruitées par la Transformée en Curvelettes", *ICA2IT, Ouargla, Algérie .Mars 2014*.
- [69] R. Acha and S. Peleg, "Lucas-kanade without iterative warping," *In International Conference on Image Processing (ICIP 2006)*, pp. 1097-1100, 2006.
- [70] F. Charif, N. Djedi, A. Benchabane, A. Taleb-Ahmed, "A simplified Zhang neural network for solving a large Toeplitz linear systems and its application for AR parameters estimation". *Pensee journal*. vol. 76, no.4, pp.354-360, 2014.

- [71] S. Li, S. Chen, B. Liu, “Accelerating a recurrent neural network to finite time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function”, *Neural Process. Lett.* vol 37, no. 2, pp. 189–205, 2013.
- [72] S. Li, Y. Li, Z. Wang, “A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application,” *Neural Netw.* Vol. 39, pp. 27–39.2013.
- [73] L. Tessens, A. Pižurica, A. Alecu, A. Munteanu, and W. Philips, “Context adaptive image denoising through modelling of curvelet domain statistics”, *Journal of Electronic Imaging*, vol. 17, pp.1-17, 2008.
- [74] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, “A database and evaluation methodology for optical Flow”, *Microsoft Research, Technical report MSR-TR-2009-179*, 2009. <http://vision.middlebury.edu/flow/>.
- [75] S. Srinivasan, R. Chellappa, “Optical Flow Using Overlapped Basis Functions for Solving Global Motion Problem”, *Proc. ECCV-B*, pp.288-304, 1998.
- [76] M. A. Arrendondo, K. Lebart, “Optical flow using textures”, *Pattern Recognition Letters*, vol 25, pp . 449–457, 2004.
- [77] H. Liu, R. Chellappa, and A. Rosenfeld, “Two-Frame Multi-Scale Optical Flow Estimation Using Wavelet Decomposition”, *Int. Conf. on Acoustics Speech and Signal Processing*, vol. 4, pp. 3588 -3591, 2002.
- [78] C. Cassisa, S. Simoens, V. Prinnet, “Two-Frame Optical Flow Formulation in an Unwarping Multiresolution Scheme”, *CIARP '09 Proc. of the 14th Iberoamerican Conf. on Pattern Recognition: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp.790–797, 2009.
- [79] M. Werlberger, T. Pock, and H. Bischof. “Motion estimation with non-local total variation regularization,” *In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [80] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. “Fast cost-volume filtering for visual correspondence and beyond,” *In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.
- [81] V. Lempitsky, S. Roth, and C. Rother. “Discrete-continuous optimization for optical flow estimation,” *In Proc. of the IEEE Conf.e on Computer Vision and Pattern Recognition* , 2008.
- [82] E. M. Flopis and J. Saechez, “Horn-Schunck Optical Flow with a Multi-Scale Strategy”. *Image processing on line*. 2012.

- [83] Y. Zhang ,S. S. Ge , “ Design and analysis of a general recurrent neural network model for time-varying matrix inversion”, *IEEE Trans. on Neural Network*. vol.16, no.6, pp. 1477-1490, 2005.
- [84] Y. Zhang, X. Guo, W. Ma, K. Chen, and B. Cai, “MATLAB Simulink modeling and simulation of Zhang neural network for online time-varying matrix inversion”, *Proc. IEEE Int. Conf. on Networking, Sensing and Control*, pp. 1480–1485, 2008.
- [85] Y. Zhang, Z. Chen, K. Chen, B. Cai, “Zhang neural network without using time-derivative information for constant and time-varying matrix inversion”, *Proceedings of IEEE Int. Joint Conf. on Neural Networks*, Hong Kong, pp. 142-146, June 2008.
- [86] Y. Zhang, D. Guo, C. Yi, L. Li, Z. Ke, “More Than Newton Iterations Generalized from Zhang Neural Network for Constant Matrix Inversion Aided with Line-Search Algorithm”, *2010 8th IEEE Int. Conf. on Control and Automation Xiamen, China, June , 2010*.
- [87] Y. Zhang, L. Jin, D. Guo, S. Fu, L. Xiao , “Three Nonlinearly-Activated Discrete-Time ZNN Models for Time-Varying Matrix Inversion”, *2012 8th Int. Conf. on Natural Computation (ICNC 2012)*, pp.163-167, 2012.
- [88] A. Benchabane, A. Bennia, F. Charif , A. Taleb-Ahmed., “Multi-Dimensional Capon Spectral Estimation Using Discrete Zhang Neural Networks. *Multidim Syst Sign Process.*, vol. 24, pp.583-598, 2013.
- [89] F. Charif, N. Djedi, A. Benchabane, “On Parallel Implementation of Horn and Schunk Motion estimation Method”, *Int. j. of computer and applications*. vol. 35, no.2, pp:79-85, 2013.
- [90] F. Charif, N. Djedi, A. Benchabane, A. Taleb-Ahmed,, “ Horn & Schunk meets a discrete Zhang neural network for computing 2D optical flow”, *ICEO13*, Ouargla, Algérie.2013.
- [91] J. L. Barron and N. A. Thacker. “Tutorial : Computing 2D and 3D optical flow ”. *Tina Memo No. 2004-012, Int. document of Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester*, 2004.
- [92] J. L. Barron, “Experience with 3D optical flow on Gated MRI cardiac datasets”, *Canadian Conf. on Computer and Robot Vision*, pp. 370-377, 2004.
- [93] T. Regert, B. Tremblais, and L. David. “Parallelized 3D optical flow method for fluid mechanics applications”, *In Elect. Proc. of the Fifth Int. Symp. on 3D Data Processing, Visualization and Transmission (3DPVT'10)*, Paris, France, May 17-20 2010.

- [94] N. Hata, A. Nabavi, W. Wells, S. Warfield, R. Kikinis, P. Black, F. Jolesz. “ Three dimensional Optical flow method for measurement of volumetric brain deformation from intraoperative magnetic resonance images”, *J Comput Assist Tomogr.* vol.24, pp. 531-538, 2000.
- [95] A. Andreopoulos, J. K. Tsotsos, “Efficient and Generalizable Statistical Models of Shape and Appearance for Analysis of Cardiac MRI”, *Medical Image Analysis*, vol. 12, no. 3, pp. 335-357. June 2008.
- [96] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, “ Highly Accurate Optic Flow Computation with Theoretically Justified Warping”, *International Journal of Computer Vision*, vol.67, no.2, pp.141-158, April 2006.
- [97] D. Sun, S. Roth, and M. Black. “Secrets of optical flow estimation and their principles”. *Computer Vision and Pattern Recognition*, pp. 2432 – 2439, 2010.
- [98] S. Zhang, *Neural Networks in Optimization*. Norwell, MA: Kluwer, 2000.
- [99] E. P. Simoncelli, “Design of multi-dimensional derivative filters”, *IEEE Int. Conf. Image Processing*, vol.1, pp.790-793, 1994.
- [100] Z.E. Baarir, F. Charif, “Fast Modified Horn & Schunck Method for the Estimation of Optical Flow Fields”, *IEEE-E- SiPS 2011*, pp. 283-288, 2011.