

THÈSE

pour l'obtention du Grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
Faculté des Sciences Fondamentales et Appliquées
Diplôme National - Arrêté du 7 Août 2006

École Doctorale : **Sciences et Ingénierie pour l'Information**
Secteur de Recherche : **Traitement du signal et des images**

Présentée par :
Olivier KIHIL

Modélisations polynomiales hiérarchisées applications à l'analyse de mouvements complexes

Dirigée et encadrée par :
Benoit TREMBLAIS
Bertrand AUGEREAU
Majdi KHOUDEIR

Thèse soutenue le 17/02/2012
devant la Commission d'Examen composée de :

J. BENOIS-PINEAU	Professeur, LABRI - Université de Bordeaux	Rapporteur
F. MERIAUDEAU	Professeur, LE2I - Université de Bourgogne	Rapporteur
A. BENSRAIR	Professeur, LITIS - INSA de Rouen	Examineur
M KHOUDEIR	Professeur, XLIM-SIC - Université de Poitiers	Directeur
B. TREMBLAIS	Maître de Conférences, XLIM-SIC - Université de Poitiers	Co-directeur
B. AUGEREAU	Maître de Conférences, XLIM-SIC - Université de Poitiers	Co-directeur

« À Alexandra et Clément »

REMERCIEMENTS

Tout d'abord, je souhaite remercier l'ensemble des membres du jury, qui ont accepté d'évaluer mes travaux. Je les remercie pour la pertinence de leurs remarques et de leurs questions.

Je tiens également à remercier MAJDI, BENOIT et BERTRAND d'avoir accepté de diriger ma thèse. Je remercie particulièrement BENOIT et BERTRAND, pour la qualité de leur encadrement et leur disponibilité tout au long de ma thèse. Je suis très heureux d'avoir travaillé avec vous et d'avoir appris tant de choses.

Je remercie le département EEA de l'université de Poitiers de m'avoir accueilli pour effectuer mon monitorat. Merci en particulier à PHILIPPE, CHRISTINE, OLIVIER A., CLENCY, ANNE-MARIE, RODOLPHE et SYLVIE. Je remercie également LAURENT qui m'a beaucoup appris lors de mon passage à l'ENSEA ainsi que PHILIPPE-HENRI, INBAR, DAVID, ANTONIA et VERONICA pour leur accueil à l'ENSEA et leur soutien pendant cette année passée à Cergy.

Je remercie aussi tous les collègues et amis du SIC pour tout ce que l'on a partagé ensemble, les broyés du Poitou, les cafés, les potins discussions d'informations non officielles, les défaites du foot en salle (il y a aussi eu des victoires, non ?), les conseils en photographie, et beaucoup de bons moments... Je tiens à remercier VIRGINIE et FRANÇOISE qui nous facilite la vie au quotidien, et qui sont aussi de très bonnes confidentes. J'adresse des remerciements spéciaux et très chaleureux à ceux qui ont été obligés de partager mon bureau plus ou moins longtemps, WINDU, SYLVAIN, KARIMA, DENIS et OLIVIER... Merci les amis. Merci à DENIS, KARIMA, OLIVIER, JF de s'être prêté à des séances de mouvements pour moi. Je remercie aussi WASSIM, ANNE-SOPHIE, ROMU, BORIS, THOMAS et ANTOINE pour les bons moments. J'oublie certainement du monde, vraiment merci à tous, venir au SIC a été (et reste) un plaisir pour moi.

Je souhaite remercier PHILIPPE CARRÉ qui m'a apporté beaucoup depuis la licence 3. Merci de m'avoir initié au traitement du signal ainsi qu'à la recherche par des mini-projets sur mesure. Tu as fait pour moi bien plus que tu ne le penses...

Je remercie aussi tous les membres de ma famille (belle famille comprise) pour leur soutien. Ces dernières années ont été longues et difficiles, on ne s'est pas vu souvent mais vous m'avez tous beaucoup soutenus. Je remercie "les cousins" d'essayer à chaque fois de comprendre ce que je fais. Maintenant vous pourrez lire le manuscrit si vous voulez... Merci à OCÉANE et CHLOÉ, mes petites princesses, vos sourires me réchauffe le coeur. Vous avez beaucoup grandi pendant ces années de thèse...

Enfin, je remercie de tout mon cœur ALEX qui enchante ma vie. Tu as tout fait pour me faciliter la vie pendant les "presques" derniers mois de ma thèse... Les tous derniers mois ont été bien occupés et chargés en émotions. En effet, deux mois avant de soutenir ma thèse nous avons (enfin surtout toi) mis au monde un petit ange, Clément. Maintenant, un nouveau chapitre de notre vie commence, mais ça c'est une autre histoire...

SOMMAIRE

I	Introduction et contexte	1
1	Introduction	1
1.1	Contexte	1
1.2	Motivation	2
1.3	Contributions	2
1.4	Organisation du document	4
2	Contexte général	7
2.1	Des séquences d'images aux champs de déplacements	7
2.1.1	Méthode de mise en correspondance de blocs	8
2.1.2	Méthodes fondées sur l'équation du flot optique	12
2.2	Utilisation des champs de déplacements	15
2.2.1	La compensation du mouvement	15
2.2.2	L'étude des mouvements de caméra	16
2.2.3	La segmentation d'images basée mouvement	17
2.2.4	La reconnaissance de mouvements	17
2.2.5	L'analyse des champs de déplacements en analyse de mouvement fluide	18
2.3	Outils de modélisation en traitement d'image	19
2.3.1	Décomposition en valeurs singulières	20
2.3.2	Séries et transformée de Fourier	21
2.3.3	La transformée en ondelettes	22
2.3.4	Les fonctions polynomiales	25
2.3.5	Conclusion	27
II	Approximation des champs de déplacements dans des bases polynomiales	29
3	Approximation globale des champs denses	33
3.1	Introduction et définitions	33
3.2	Construction de la base polynomiale	35
3.3	Approximation globale des champs denses	37
3.3.1	Utilisation des bases de degré 1	37
3.3.2	Utilisation de bases de degré supérieur à 1	39
3.3.3	Gain de compression des bases de polynômes	40
3.4	Conclusion sur la modélisation globale	43

4	Approximation hiérarchisée des champs denses	47
4.1	Subdiviser le champ de déplacements	47
4.2	Subdivision en grille régulière	48
4.3	Modélisation par quadtree	49
4.4	Subdivision par kd-tree	51
4.5	Modélisation obtenue par subdivision à partir de points singuliers du champ de déplacements	58
4.5.1	Les boîtes englobantes	59
4.6	Le diagramme de Voronoï	64
4.7	Conclusion	68
5	Approximation hiérarchisée des champs épars	73
5.1	Difficultés d'approximation des champs épars	73
5.2	Bases de polynômes orthogonaux définies sur un support épars	76
5.3	Identification	81
5.4	Approximation hiérarchique avec les bases éparses	87
5.4.1	Quadtree avec les bases éparses	87
5.4.2	Diagramme de Voronoï avec les bases éparses	91
5.5	Conclusion	91
6	Bases de polynômes A-complètes	93
6.1	Vers une approximation à erreur nulle	93
6.2	Du Polynôme d'interpolation de Lagrange aux bases de polynômes A-complètes	94
6.2.1	Polynôme d'interpolation de Lagrange	94
6.2.2	Les Bases A-complètes	94
6.2.3	Bases A-complètes en dimension 2	96
6.3	Bases A-complètes et approximation à erreur nulle	98
6.3.1	Comparaison avec la Transformée en Cosinus Discrète (DCT)	99
6.3.2	Comparaison avec la SVD	100
6.3.3	Multi-résolutions	103
6.4	Application aux champs de déplacements	114
6.5	Comparaison de l'approximation par base A-complète aux méthodes précédentes	116
6.6	Conclusion	118

III Applications **119**

7	Détection et classification des points singuliers	121
7.1	Introduction	121
7.2	Caractérisation mathématique des points singuliers	122
7.2.1	Définitions	122
7.2.2	Portraits de phase	123
7.2.3	Indice de Poincaré	126
7.3	État de l'art sur la détection de points singuliers	128
7.3.1	Méthodes fondées sur l'utilisation du portrait de phase	128
7.3.2	Estimation du champ de déplacements et détection de structures singulières simultanées	129
7.3.3	L'indice de Poincaré	130
7.3.4	Les fonctions de potentiels	131

7.3.5	Méthodes de détection des tourbillons	133
7.3.6	Synthèse	135
7.4	Polynômes orthogonaux multi-variables pour la détection de points singuliers .	136
7.4.1	Approximation du modèle affine	136
7.4.2	Schémas multi-résolutions	137
7.4.3	Détermination de la zone de linéarité autour de la singularité	140
7.5	Résultats de nos méthodes de détection et de classification de singularités . . .	140
7.5.1	Comparaison entre la détection à partir de notre estimateur et à partir de l'estimateur de Levenberg-Marquardt	140
7.5.2	Résultats des algorithmes multi-résolutions sur le champ 100 de la séquence <u>Cylindre-Carré</u>	146
7.5.3	Résultats de l'estimation de la zone de linéarité	151
7.5.4	Comparaison avec la détection par les fonctions de potentiels	151
7.5.5	Comparaison avec le critère λ_2	157
7.6	Conclusion	160
8	Reconnaissance du mouvement dans les vidéos	163
8.1	Introduction	163
8.2	État de l'art	163
8.2.1	Détecteur de points	164
8.2.2	Méthodes reposant sur un modèle d'apparence de mouvement	164
8.2.3	Méthodes reposant sur un descripteur de champ de déplacements	167
8.3	Notre descripteur de mouvement fondé sur la projection sur une base de polynômes	173
8.4	Algorithme de reconnaissance de mouvements humains	176
8.4.1	Présentation de l'algorithme	176
8.4.2	Résultats sur une application de type commande de jeu vidéo	180
8.5	Application à la classification d'une base de vidéos	183
8.5.1	Présentation de la base de WEIZMANN	184
8.5.2	Présentation de l'algorithme global de classification	184
8.5.3	Présentation de l'algorithme hiérarchique de classification	188
8.5.4	Résultats de classification sur la base de WEIZMANN	189
8.5.5	Résultats avec l'algorithme global	190
8.5.6	Résultats avec l'algorithme hiérarchique	194
8.5.7	Comparaison avec la littérature	197
8.6	Conclusion	201
IV	Conclusion et perspectives	205
9	Conclusion et perspectives	207
A	Squence Cylindre-Carré	211
B	Complexité et occupation mémoire des bases polynomiales denses	215
	Bibliographie	217
	Publications personnelles	231

TABLE DES FIGURES

2.1	Correspondance entre le mouvement réel et le mouvement apparent.	7
2.2	Représentation d’une carte de corrélation	9
2.3	Illustration de l’algorithme de recherche à 3 étapes.	11
2.4	Illustration de l’algorithme adaptatif de recherche en croix.	11
2.5	Les trois cas possibles lors de l’extraction du flot. (a) aucune information disponible : pas de flot. (b) problème d’ouverture : on ne peut calculer que le flot normal au contour. (c) suffisamment d’informations : le flot calculé est correct.	13
2.6	Compensation du mouvement sur deux images d’un flux mpeg	16
2.7	Compensation du mouvement : (a) Image de la vidéo originale à $t = 0$; (b) Image de la vidéo originale à $t = 20$; (c) Image de la vidéo après suppression du mouvement de caméra à $t = 0$; (d) Image de la vidéo après suppression du mouvement de caméra à $t = 20$	17
2.8	Analyse d’écoulement à partir de coefficients POD	19
2.9	Détection de singularités dans un champ de déplacements	19
2.10	Illustration de la modélisation d’une image par SVD	20
2.11	Schéma d’un banc de filtres d’analyse et de synthèse	25
3.1	Mouvements affines élémentaires	38
3.2	Exemple de détection de singularités dans un champ de déplacements	39
3.3	Exemple de projections d’un champ complexe	40
3.4	Analyse d’écoulement à partir de coefficients POD	41
3.6	Exemple de mouvement humain	43
4.1	Subdiviser pour mieux régner : un cas d’école ; (a) Champ de déplacements composé de 4 champs affines juxtaposés ; (b) Approximation globale du champ (a) avec une base de degré 1.	48
4.2	EQM en fonction du niveau de subdivision et du degré de la base	49
4.3	Représentation de la carte d’erreurs quadratiques d’un champ pour différentes modélisations globales et par subdivision par grilles régulières	50
4.4	Représentation d’une image binaire par un quadtree	51
4.5	Représentation de la carte d’erreurs quadratiques associée à la modélisation du champ test à l’aide d’un quadtree avec 3 bases de degrés différents	52
4.6	Exemple de Kd-tree et son arbre associé	53
4.7	Représentation de la carte d’erreurs quadratiques associée à la modélisation du champ test à l’aide d’un kd-tree médian avec 3 bases de degrés différents	54
4.8	Illustration des six combinaisons possibles du kd-tree irrégulier	55

4.9	Représentation de l'erreur quadratique locale associée à la modélisation du champ test à l'aide d'un kd-tree irrégulier avec 3 bases de degrés différents . . .	56
4.10	Résultats visuels de la qualité d'approximation du quadtree et du kd-tree	57
4.11	Exemple de zones non séparables par une droite horizontale ou verticale	59
4.13	Représentation de la carte d'erreurs quadratiques associée à la modélisation du champ test à l'aide des boîtes englobantes	61
4.14	Champ synthétique composé de 5 mouvements affines séparés	62
4.15	Étape de la rotation discrète d'un champ synthétique affine (rotation)	63
4.16	Différents mouvements composant le champ synthétique	65
4.17	Illustration du diagramme de Voronoï et son dual la triangulation de Delaunay .	66
4.18	Partition obtenue par le diagramme de Voronoï à partir des points singuliers du champ 131 extrait de la séquence Cylindre-Carré	66
4.20	Carte d'erreurs quadratiques associée à l'approximation du champ test obtenue à partir du diagramme de Voronoï avec une distance $L1$ et avec une distance $L2$ pour des boîtes englobantes non-orientées	69
4.21	Carte d'erreurs quadratiques associée à l'approximation du champ test obtenue à partir du diagramme de Voronoï avec une distance $L2$ pour des boîtes englobantes orientées ou non-orientées	70
5.1	Illustration de l'approximation globale d'une rotation affine épars avec une base de polynômes dense de degré 1	74
5.2	Illustration de l'approximation globale d'une rotation affine épars avec une base de polynômes dense de degré 16	74
5.3	Champs tests épars pour les tests de subdivisions par quadtree	75
5.6	Champ $polynomial_{1/3}$	78
5.7	Modélisation du mouvement <i>Cylindre-carré épars 1</i> : comparaison entre une base dense et une base non-dense	79
5.8	Modélisation du mouvement <i>Cylindre-carré épars 2</i> : comparaison entre une base dense et une base non-dense	80
5.10	Résultats d'interpolation obtenus à partir d'une identification dans la base canonique ; (a) Interpolation d'une rotation affine épars (cf. figure FIG 5.9) (b) Interpolation du <i>champ polynomial_{1/3}</i> (cf. figure FIG 5.6).	84
5.11	Champ cylindre-carré $\mathcal{C}_{1/3}$	85
5.12	Comparaison entre l'approximation globale d'un champ non polynomial dense (en (a)) et l'approximation d'une version épars du même champ dont l'amplitude de deux vecteurs sur trois a été annulée horizontalement et verticalement ; (a) Champ Cylindre-Carré dense ; (b) Approximation obtenue par projection de (a) sur une base polynomiale dense de degré 16 ; (c) Approximation obtenue par projection de <i>cylindre-carré</i> $\mathcal{C}_{1/3}$ sur une base polynomiale épars de degré 16, après identification et interpolation ; (d) Carte d'erreurs quadratiques entre (b) et (c).	86
5.13	Modélisation d'un mouvement complexe non-dense (exemple 1) avec un quadtree associé à une base épars	88
5.14	Modélisation d'un mouvement complexe non-dense (exemple 1) avec un quadtree associé à une base épars	89
5.15	Exemple d'interpolation sur un extrait du champ Cylindre-carré non dense . . .	90
5.16	Utilisation des bases de polynômes épars pour une approximation basée sur le diagramme de Voronoï	92

6.1	Reconstruction parfaite d'une image après projection dans une base de polynômes bi-variables A-complète	98
6.2	Approximation de l'image Lena (512 × 512 pixels) avec les bases A-complètes avec une grille régulière de 2 × 2 pixels (a) image Lena originale; (b) image reconstruite	99
6.3	Image "bikes" extraite de la base LIVE Image Quality Database [Sheikh 03], [Sheikh 06], [Wang 04]	99
6.4	Evolution du PSNR en fonction de la quantité d'énergie conservée pour les coefficients DCT ou polynomiaux de chaque bloc; Les courbes en trait continu rouge correspondent aux bases de polynômes A-complètes et les traits discontinus vert correspondent aux DCT; (a) La grille est de 8 × 8 pixels; (b) La grille est de 16 × 16 pixels.	101
6.6	Evolution du PSNR en fonction du gain de compression pour une base A-complète et une SVD toutes les deux appliquées sur une grille régulière; (a) Résultats obtenus avec une grille composée de blocs de 8 × 8 pixels; (b) Résultats obtenus avec une grille composée de blocs de 16 × 16 pixels	103
6.7	Construction du premier niveau de l'analyse d'une image en niveau de gris avec une base polynomiale A-complète; (a) petite image synthétique de 4 × 4 pixels; (b) Projection de chaque bloc sur la base de polynômes A-complète de 2 × 2 polynômes; (c) Regroupement des coefficients polynomiaux en imgettes.	104
6.8	(a) Exemple de regroupement des coefficients issus de la projection dans un même élément de la base pour des bases A-complètes de 4 éléments (bloc de 2 × 2 pixels). (b) Exemple de regroupement des coefficients polynomiaux en imgettes avec une base A-complète de 64 éléments (bloc de 8 × 8 pixels); <u>Pour l'affichage, les images (a) et (b) ont été égalisées.</u>	105
6.9	Représentation des coefficients conservés d'une échelle à la suivante	105
6.10	Décomposition dyadique de l'image "bikes" par base de polynômes A-complète de $8imes8$ polynômes	106
6.11	Reconstruction dyadique de l'image "bikes" par base de polynômes A-complète à partir du résultat obtenu en 6.10	107
6.12	Images tests issues de la base LIVE; (a) Image "bikes"; (b) Image "lighthouse2"; (c) Image "parrots".	108
6.13	Evolution du PSNR en fonction du gain de compression pour une décomposition multi-résolutions dyadique par base de polynômes A-complète. Les bases de niveau 1 sont de 4 × 4, 8 × 8, 16 × 16, 32 × 32 polynômes. Les bases de niveau 2 sont alors respectivement de 2 × 2, 4 × 4, 8 × 8, 16 × 16 polynômes.	109
6.14	Visualisation de la qualité de l'image parrots en fonction de la résolution de l'approximation par décomposition dyadique	110
6.15	Décomposition de l'image "bikes" avec une base de niveau 1 de 8 × 8 polynômes et une base de niveau 2 de 7 × 7 polynômes	111
6.16	Comparaison de l'évolution du PSNR en fonction du gain de compression entre une décomposition multi-résolutions dyadique et une décomposition multi-résolutions non-dyadique à partir de l'image "parrots"	112

6.17	Evolution du PSNR en fonction du gain de compression pour une décomposition multi-résolutions dyadique par base de polynômes A-complète. Les bases de niveau 1 sont de 4×4 , 8×8 , 16×16 , 32×32 polynômes. Les bases de niveau 2 sont alors respectivement de 2×2 , 4×4 , 8×8 , 16×16 polynômes. (a) Graphique obtenu pour l'image "bikes"; (b) Graphique obtenu pour l'image "lighthouse2"; (c) Graphique obtenu pour l'image "parrots".	113
6.18	Présentation des champs tests; (a) Champ 131 de la séquence <u>Cylindre-carré</u> (cf. ANNEXE A);(b) Champ <u>Marche</u>	114
6.19	Evolution de l'erreur quadratique moyenne en fonction du gain de compression pour la décomposition dyadique d'un champ de déplacements; (a) Résultats pour le champ 131 de la séquence <u>cylindre-carré</u> ; (b) Résultats pour le champ " <u>Marche</u> "	115
6.20	Evolution de l'erreur quadratique moyenne en fonction du gain de compression pour la décomposition dyadique d'un champ de déplacements; (a) Résultats pour le champ 131 de la séquence <u>cylindre-carré</u> ; (b) Résultats pour le champ " <u>Marche</u> "	116
7.1	Arbre binaire de représentation des invariants algébriques de la matrice A pour la classification des singularités proposé dans [Maurizot 95]	131
7.2	Synthèse de l'algorithme à fenêtre adaptative présenté dans [Kihl 08a].	138
7.3	Synthèse de l'algorithme multi-résolutions à fenêtre mobile.	139
7.4	Illustration des deux résultats $W1$ et $W2$ obtenus par l'estimateur de LEVENBERG-MARQUARDT	142
7.5	Illustration d'un cas où la méthode de LEVENBERG-MARQUARDT ne fournit une bonne approximation, ni pour $W1$, ni pour $W2$	143
7.6	Illustration de la résistance au bruit Gaussien de la méthode de projection . . .	144
7.7	Champ 100 de la séquence <u>Cylindre-carré</u> utilisé pour tester nos trois algorithmes de détection de singularités	147
7.8	Détection de singularités avec l'algorithme à fenêtre adaptative sur le champ 100 de la séquence <u>Cylindre-carré</u>	148
7.9	Détection de singularités avec l'algorithme à fenêtre adaptative sur le champ 100 de la séquence <u>Cylindre-carré</u>	148
7.10	Détection de singularités sur le champ 100 de la séquence <u>Cylindre-carré</u> avec l'algorithme multi-résolutions à fenêtre mobile pour 11 niveaux de décomposition	149
7.11	Évolution de la détection de singularités sur le champ 100 de la séquence <u>Cylindre-carré</u> avec l'algorithme multi-résolutions à fenêtre mobile sur 3 niveaux de décomposition	149
7.12	Détection de singularités sur le champ 100 de la séquence <u>Cylindre-carré</u> avec l'algorithme par quadtree pour 6 niveaux de décomposition	150
7.13	Synthèse des algorithmes de détection de singularités sur le champ 100 de la séquence <u>Cylindre-carré</u>	150
7.14	Résultats de l'algorithme d'estimation de la zone de linéarité	152
7.15	Champ de déplacements synthétique extrait de [Corpetti 03] créé à partir des paramètres présentés dans le tableau TAB 7.5.4	154
7.16	Résultats de l'algorithme de quadtree sur le champ de déplacements synthétique FIGURE.7.15	154
7.17	Résultats de l'algorithme de quadtree sur le champ de déplacements synthétique (cf. figure FIG 7.15 avec l'algorithme modifié	155

7.18	Zoom sur la structure détectée à la position (97, 172)	157
7.19	Dispositif expérimental et aperçu du champ de déplacements correspondant.	158
7.20	Comparaison des méthodes : centres détectés par l’algorithme à fenêtre adaptative (□) et tourbillons détectés par le critère λ_2 (○).	158
7.21	Comparaison des méthodes sur un champ localement affine : centres détectés par l’algorithme à fenêtre adaptative (□) et tourbillons détectés par le critère λ_2 (○); Le champ présenté ici est obtenu à partir du champ de la figure FIG 7.20; Pour chaque singularité repérée par un carré sur cette figure, le champ est reconstruit localement avec un modèle affine.	159
7.22	Exemple de structures détectées et aperçu des champs de déplacements correspondant dans la fenêtre mobile attachée à chaque structure.	160
7.23	Cumul des tourbillons détectés en tout point du champ sur l’ensemble de la séquence	161
8.1	walk : Mesure de similarité entre le champ (c) et le champ (d) = 0.96 (avec une base de degré 7	177
8.2	jack : Mesure de similarité entre le champ (c) et le champ (d) = 0.80 (avec une base de degré 7	178
8.5	Les 5 mouvements de référence utilisés pour nos tests	182
8.6	Reconnaissance de mouvement de 5 actions avec une base de polynômes de degré 3. Pour chaque utilisateur, nous présentons sur la ligne du haut la séquence attendue et sur la ligne du bas les résultats obtenus avec notre algorithme de détection. * frapper avec un direct du bras avant; □ frapper avec un direct du bras arrière; ○ esquiver un coup en bougeant à gauche; △ esquiver un coup en bougeant à droite; ▽ se baisser.	183
8.7	Cinq premiers mouvements de la base de WEIZMANN	185
8.8	Cinq derniers mouvements de la base de WEIZMANN	186
8.10	Illustration de notre algorithme global de classification de vidéo	188
8.11	Illustration de la création d’une empreinte avec l’algorithme hiérarchique	189
8.12	Illustration de l’impact de l’extraction de la zone d’intérêt sur la classification des actions bend et wave1	191
A.1	Deux champs de la séquence <u>Cylindre-Carré</u> ; (a) Champ 100 de la séquence <u>Cylindre-Carré</u> ; (b) Champ 131 de la séquence <u>Cylindre-Carré</u>	212
A.2	Séquence « cylindre-carré » : topologie de l’écoulement	212
A.3	Séquence « cylindre-carré » : représentation du dispositif expérimental	213
A.4	Séquence « cylindre-carré » : exemple d’image obtenue avec le dispositif d’acquisition	213

LISTE DES TABLEAUX

3.1	Fonction de pondération des différentes famille de polynômes	36
4.1	Evaluation de l'erreur d'approximation et du gain de compression pour les méthodes globales, de subdivision par grille régulière, par quadtree et par kd-tree pour différents degrés et différents niveaux de hiérarchiques	58
4.2	Evaluation de l'erreur de modélisation et du gain de compression pour les méthodes globale, de subdivision par grille régulière, par quadtree, par kd-tree, par boîtes englobantes et par diagramme de Voronoï	71
6.1	Fonction de pondération des différentes famille de polynômes	97
6.2	Tableau représentant les coefficients polynomiaux d'un bloc de 4×4 pixels choisi au hasard dans l'image "bikes"	101
6.3	Tableau représentant les coefficients DCT d'un bloc de 4×4 pixels choisi au hasard dans l'image "bikes"	101
6.4	Evaluation de l'erreur de modélisation et du gain de compression pour les méthodes globale, de subdivision par grille régulière, par quadtree par boîtes englobantes, par diagramme de Voronoï, par décomposition dyadique avec une base polynomiale A-complète et par décomposition non-dyadique avec une base polynomiale A-complète	117
7.1	Classification de différents portraits de phase affine d'un champ de vecteurs $2D$	129
7.2	Tableau de synthèse des différentes approches de détection de singularités . . .	135
7.3	Résultats quantitatifs de détection de singularités dans des champs de vecteurs affines normalisés par trois méthodes (cas1)	145
7.4	Résultats quantitatifs de détection de singularités dans des champs de vecteurs affines non normalisés par trois méthodes (cas2)	145
7.5	Résultats quantitatifs de détection de singularités dans dans champs de vecteurs affines bruités par trois méthodes	146
7.6	Paramètres pour obtenir le champ de déplacements (396×276 vecteurs) présenté sur la figure FIG 7.15	153
7.7	Comparaison des positions des singularités obtenues par la méthode proposée dans [Corpetti 03] et par notre méthode	156
7.8	Comparaison de l'estimation des rayons des structures associées aux singularités détectées par la méthode proposée dans [Corpetti 03] et par notre méthode .	156
7.9	Comparaison de l'estimation de la force des structures associées aux singularités détectées par la méthode proposée dans [Corpetti 03] et par notre méthode .	156

8.1	Tableau récapitulatif des calculs de similarité entre les champs ; <u>Daria_walk</u> cf. figure FIG 8.1(c) ; <u>Denis_walk</u> cf. figure FIG 8.1(d) ; <u>Daria_jack</u> cf. figure FIG 8.2(c) ; <u>Denis_jack</u> cf. figure FIG 8.2(d) ;	176
8.2	Indice de début et de fin des actions pour chaque classe d'action et pour chaque acteur ; ces indices sont sélectionnés manuellement.	185
8.3	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 1 et une seule référence par mouvement type (6 coefficients, 73.84%).	190
8.4	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 5 et une seule référence par mouvement type (42 coefficients, 90.12%).	192
8.5	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 12 et une seule référence par mouvement type (182 coefficients, 92.34%).	193
8.6	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 1 et deux références par mouvement type (6 coefficients, 79.8%).	194
8.7	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 8 et deux références par mouvement type (90 coefficients, 95.76%).	194
8.8	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 1 et une seule référence par mouvement type (30 coefficients, 89.25%).	196
8.9	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 3 et une seule référence par mouvement type (100 coefficients, 91, 96%).	196
8.10	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 1 et deux références par mouvement type (30 coefficients, 94.08%).	198
8.11	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 3 et deux références par mouvement type (100 coefficients, 96.22%).	198
8.12	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme global, une base de degré 5 et la stratégie "Leave One Actor Out" (42 coefficients, 98.89%).	199
8.13	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme global, une base de degré 11 et la stratégie "Leave One Actor Out" (156 coefficients, 100.00%).	200
8.14	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 1 et la stratégie "Leave One Actor Out" (30 coefficients, 97.78%).	200
8.15	Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 3 et la stratégie "Leave One Actor Out" (100 coefficients, 100.00%).	201
8.16	Résultats de classification de la base de WEIZMANN de la littérature	202

Première partie

Introduction et contexte

CHAPITRE 1

INTRODUCTION

1.1 Contexte

Les travaux de cette thèse s'inscrivent dans le contexte du traitement du signal et des images. L'augmentation de la puissance de calcul des ordinateurs ainsi que des capacités de stockage ont permis d'étendre ces méthodes à un grand nombre de domaines tels que le traitement de vidéos. Il est maintenant possible de réaliser des acquisitions vidéos à une cadence de quelques milliers d'images par seconde. Ces avancées scientifiques et technologiques ont considérablement augmenté les possibilités expérimentales dans des domaines tels que celui de l'analyse des mouvements fluides. Dans le même temps, les quantités de données à manipuler ont considérablement augmenté, introduisant ainsi de nouvelles problématiques dans le domaine de l'analyse automatique des données numériques. Des recherches sont alors menées afin de répondre à ces problèmes, en réduisant la quantité d'informations utiles ou en détectant les structures principales des écoulements. Dans le domaine du grand public, la vidéo numérique a suivi le chemin de la photographie numérique. Le coût des caméras vidéo numériques est devenu très faible. Il est maintenant possible de réaliser des films avec un téléphone portable. Les réseaux sociaux et autres sites de visionnage de vidéos en direct fleurissent sur internet. De nouvelles thématiques de recherche voient alors le jour, notamment pour l'indexation automatique de vidéos par le contenu. Parallèlement, le domaine de la vidéo surveillance a explosé, en raison du faible coût des caméras, de la qualité et de la miniaturisation des dispositifs... La ville de Londres, par exemple, est actuellement équipée de plus de 10000 caméras de surveillance. Cependant, de si grands réseaux de caméras, ne peuvent pas tous être surveillés en permanence par des opérateurs humains. Il y a donc là aussi de grands enjeux pour la recherche en reconnaissance d'actions humaines. On peut encore citer l'analyse du geste sportif, l'aide à la personne pour des applications telles que la détection de chute dans les douches ou les escaliers, etc. Il y a au moins un point commun à toutes les thématiques que nous venons de citer, puisqu'elles font toutes intervenir la notion de mouvement.

1.2 Motivation

C'est justement sur ce point commun, le mouvement, que nous avons travaillé pendant cette thèse. Le mouvement, qu'il s'agit de modéliser, peut être représenté par un champ de vecteurs. Ce dernier peut être dense ou épars selon la nature du mouvement ou la méthode d'estimation. L'objet de ce travail concerne l'étude et le développement d'un outil générique de modélisation du mouvement, pouvant s'adapter aux différents domaines évoqués ci-dessus. Ces applications faisant intervenir différents domaines de la science expérimentale, il nous a semblé intéressant d'avoir recours à un modèle reposant sur des fondements mathématiques solides. Nous avons ainsi à disposition un formalisme pouvant être appliqué à différents contextes. Nous avons étudié les bases de polynômes orthogonaux pour la modélisation du mouvement.

Ce travail a été effectué au département SIC (Signal Image et Communication) du laboratoire Xlim, à l'Université de Poitiers. Le premier objectif est d'étendre les travaux de MARTIN DRUON [Druon 09] concernant l'utilisation des bases de polynômes orthogonaux pour l'analyse de mouvements issue de séquences vidéo expérimentales de la mécanique des fluides. Au cours de ses travaux, MARTIN DRUON a comparé cette méthode de modélisation globale du mouvement à une autre méthode couramment utilisée dans ce domaine ayant recours à une décomposition en valeurs singulières de la séquence. Encouragés par la qualité des résultats, nous avons souhaité poursuivre les recherches concernant l'utilisation des bases de polynômes pour la modélisation du mouvement suivant trois axes principaux. Le premier concerne l'étude des modélisations hiérarchiques permettant d'aboutir à différents niveaux d'analyses d'un même mouvement, le second, la modélisation des champs épars et le troisième, l'utilisation de cette méthode de modélisation à d'autres mouvements que les mouvements fluides.

1.3 Contributions

Modélisation hiérarchique d'un champ de déplacements

Dans un premier temps, nous nous sommes intéressés à la modélisation hiérarchique d'un champ de déplacements. Nous avons donc étudié différentes méthodes classiques de subdivisions du domaine spatial. En utilisant des méthodes arborescentes telles que le quadtree, nous avons constaté qu'il est possible d'obtenir, à partir d'une subdivision irrégulière, une modélisation d'un champ de déplacements de qualité comparable à celle obtenue avec la plus petite grille régulière correspondant à plus petite subdivision utilisée dans le quadtree. Ces méthodes permettent de garantir une qualité de reconstruction d'un champ de déplacements complexe avec des bases de polynômes de degrés faibles tout en obtenant des gains de compression relativement élevés. De plus, il est aussi possible de modéliser les champs épars en subdivisant le champ en fonction de sa densité.

Modélisation des champs épars

La modélisation hiérarchique d'un champ de déplacements permet d'obtenir une approximation des champs épars de meilleure qualité que les modélisations globales même avec un degré élevé. Cependant, en reconsidérant la méthode de génération des bases de polynômes orthogonaux, nous avons proposé une méthode de génération des bases de polynômes s'adaptant à la densité du champ. Ainsi, à condition de connaître la densité des champs pour générer la

base qui convient, nous pouvons modéliser les champs épars avec la même qualité que celle des champs denses. Pour pouvoir comparer deux champs épars, il est nécessaire de réaliser un changement de base des coefficients d'une base épars vers une base dense. Nous proposons une solution pour réaliser cette transformation. Il est alors possible de comparer deux champs épars en comparant leur coefficients dans une base dense.

Base de polynômes pour l'approximation à erreur nulle

Il est possible d'obtenir une approximation à erreur nulle à partir de projections sur des bases de polynômes orthogonaux, comme cela peut être effectué avec la transformée de Fourier ou avec les transformées en ondelettes. En nous appuyant sur le polynôme d'interpolation de LAGRANGE, nous présentons les conditions pour rendre cela possible et la procédure pour générer une telle base. Nous pouvons alors réaliser une modélisation à erreur nulle de tout champ de déplacement à conditions d'avoir une base composée d'autant de polynômes qu'il y a de vecteurs dans le champ. Cela étant, pour des raisons calculatoire, cette méthode n'est applicable dans la pratique, que sur de petites images ou par subdivision avec une grille régulière sur des blocs de petites dimensions. À partir de là, nous présentons une méthode permettant d'obtenir une transformée multi-résolutions d'un champ de déplacements utilisant les bases de polynômes. Cette méthode consiste à regrouper les coefficients d'un même polynôme sur une même image. Il est alors possible de reconstruire une image avec un nombre limité de polynômes afin d'obtenir une image défini sur une dimension spatiale réduite. Les bases de polynômes pour l'approximation à erreur nulle offre de nombreux avantages, elle permettent de calculer des dérivées sur un domaine donné, offre un gain de compression comparable à celui de la DCT et nous permettent d'obtenir un schéma multi-résolutions. La décomposition multi-résolutions à partir de bases de polynômes est utilisée pour l'un des algorithmes de détection de singularités que nous proposons.

Détection et classification de points singuliers

Dans un certain nombre de domaines d'applications tels que la mécanique expérimentale des fluides, la météorologie, etc, il est commun d'analyser les champs de déplacements à partir de certains points du champ présentant des propriétés intéressantes. Ces points sont appelés points singuliers. Nous utilisons notre méthode de modélisation du mouvement par projection sur une base de polynômes afin de détecter ces points singuliers et de les classifier. Nous présentons trois schémas multi-résolutions s'appuyant sur les modélisations présentées dans nos premières contributions.

Modélisation des mouvements humains

Nous avons appliqué notre modèle polynomial à deux applications nécessitant la modélisation du mouvement humain. La première concerne la détection et reconnaissance d'actions pour un application de type contrôle de jeu vidéo par webcam. Le seconde concerne la classification de la base de vidéo de WEIZMANN [], composée de dix mouvements humains différents. Le modèle polynomial permet d'obtenir des résultats proches de la littérature avec un nombre de références très faible. En appliquant une procédure qui consiste à utiliser toutes les vidéos de la base comme référence, mise à part celle que l'on cherche à classifier (Leave one out), nous avons obtenu 100 % de classification correcte sur cette base.

1.4 Organisation du document

Ce document est organisé en trois parties.

Première partie

La première partie donne des précisions sur le contexte de nos travaux. Nous présentons les principales méthodes d'estimation du mouvement apparent dans une séquence d'images. Ensuite, nous décrivons quelques exemples d'applications qui utilisent le mouvement. Enfin, nous insistons sur la nécessité d'utiliser un modèle, puis nous justifions le choix des bases de polynômes orthogonaux.

Seconde partie

La seconde partie concerne l'étude des bases de polynômes orthogonaux pour la modélisation du mouvement. Dans le chapitre 3, nous fixons le cadre mathématique de l'approximation de fonctions. Nous présentons une méthode de génération des bases de polynômes orthogonaux. Ensuite, nous présentons la modélisation globale d'un champ dense avec les bases de polynômes. Nous rappelons aussi quelques résultats de la thèse de MARTIN DRUON [Druon 09]. Après avoir mis en évidence quelques limites de la modélisation globale, nous traitons au chapitre 4, la modélisation hiérarchique d'un champ de déplacements. Nous utilisons les bases de polynômes avec un certain nombre d'algorithmes classiques de subdivision tels que le quadtree ou le kd-tree. Après avoir constaté que ces différentes méthodes classiques permettent d'obtenir une modélisation hiérarchique de qualité, nous abordons, au chapitre 5, la modélisation des champs épars. Bien que les méthodes hiérarchiques permettent d'améliorer les résultats d'approximation par rapport à une modélisation globale, les résultats restent insuffisants. Nous proposons alors de revenir sur la génération des bases de polynômes orthogonaux en générant une base en fonction du support spatial du champ. On peut alors à nouveau utiliser tous les algorithmes hiérarchiques étudiés aux chapitres précédents. Finalement, après avoir étudié différentes méthodes d'approximation des champs de déplacements, nous proposons, au chapitre 6, de générer des bases de polynômes d'approximation à erreur nulle. À partir de ces bases, nous proposons un algorithme de décomposition multi-résolutions des champs de déplacements.

Troisième partie

La troisième partie concerne deux applications développées à partir des travaux théoriques précédents. Ces deux applications portent sur des contextes complètement différents, ce qui témoigne de la généralité de notre modèle.

Au chapitre 7, nous traitons de la détection des points singuliers dans un champ de déplacements. Après avoir donné le cadre mathématique permettant de mieux comprendre ce que sont ces points, nous en rappelons les propriétés. Ces dernières nous permettent de comprendre pourquoi ces points sont essentiels à l'analyse d'un champ de déplacements. Nous présentons ensuite un état de l'art des méthodes de détection de singularités. Enfin, nous présentons trois algorithmes reposant sur notre modélisation du mouvement par un modèle affine obtenu par projection sur une base de polynômes orthogonaux. Nous comparons celui-ci à d'autres approches de la littérature.

Au chapitre 8, nous nous plaçons dans le contexte de l'analyse de mouvements humains. Dans un premier temps, nous présentons un état de l'art des méthodes de reconnaissance du mouvement humain qui utilisent l'information du mouvement. Puis, nous présentons un algorithme de détection et reconnaissance d'actions humaines dans un contexte de commande de jeu vidéo par webcam. Nous présentons nos résultats pour une application dans laquelle un joueur effectue une série de mouvements devant une caméra. Cette série de mouvements est composée de cinq mouvements types. L'objectif de l'application est de reconnaître tous les mouvements effectués par le joueur. Enfin, nous présentons une dernière application. Cette fois, nous cherchons à classifier la base de vidéos de Weizmann [Blank 05].

CHAPITRE 2

CONTEXTE GÉNÉRAL

2.1 Des séquences d'images aux champs de déplacements

L'ensemble des travaux présentés dans ce manuscrit s'applique à tout champ de déplacements, mais ici, afin de présenter les méthodes d'estimation du mouvement, on se restreint à ceux issus de séquences vidéos numériques.

Durant le processus d'acquisition, le mouvement réel en trois dimensions est projeté sur un plan en deux dimensions (cf. FIG 2.1). Ce déplacement 3D projeté sur un plan 2D est alors appelé mouvement apparent. Dans la suite de ce document, le terme mouvement correspond toujours au mouvement apparent.

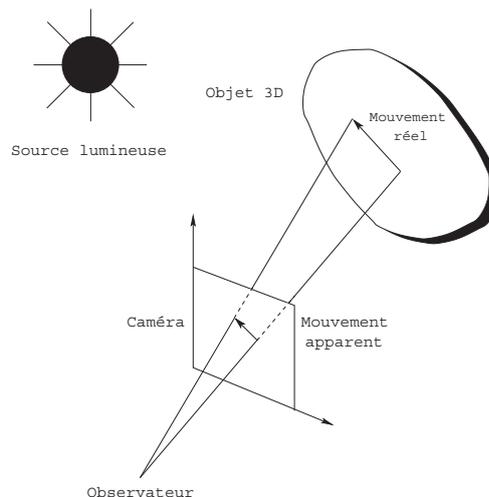


FIGURE 2.1 – Correspondance entre le mouvement réel et le mouvement apparent.

Ce mouvement peut être représenté par un champ de déplacements bidimensionnel \mathcal{C} , défini comme une application d'un ouvert Ω de \mathbb{R}^2 , représentant le domaine spatial sur lequel le champ est défini, dans \mathbb{R}^2 :

$$\begin{aligned} \mathcal{C} : \Omega \subset \mathbb{R}^2 &\rightarrow \mathbb{R}^2 \\ (x_1, x_2) &\mapsto (u, v) \end{aligned} \quad (2.1)$$

Ce champ peut être décomposé en deux applications \mathcal{U} et \mathcal{V} , définies par l'équation (2.2), correspondant respectivement aux déplacements horizontaux et verticaux aux points de coordonnées $(x_1, x_2) \in \Omega$.

$$\begin{aligned} \mathcal{U} : \Omega \subset \mathbb{R}^2 &\rightarrow \mathbb{R} & \text{et} & \quad \mathcal{V} : \Omega \subset \mathbb{R}^2 &\rightarrow \mathbb{R} \\ (x_1, x_2) &\mapsto \mathcal{U}(x_1, x_2) & & & (x_1, x_2) \mapsto \mathcal{V}(x_1, x_2) \end{aligned} \quad (2.2)$$

Ces champs sont calculés à partir d'une séquence d'images $\mathcal{S}(x_1, x_2, t)$ composée des images $I(x_1, x_2)$ à tous les instants t , où t représente la variable temporelle.

Ci après, nous présentons les grandes lignes des principales approches utilisées pour estimer le mouvement dans une séquence d'images. Le lecteur désireux de trouver des présentations plus exhaustives sur le sujet pourra se reporter aux références suivantes [Barron 94], [Weickert 06], [Heitz 10], [Fleet 05], [Ben-Ari 09], [Black 96a], [Nir 08]. Ces approches sont communément regroupées en deux familles, les méthodes d'estimation par mise en correspondance de blocs et les méthodes dites d'estimation par flot optique.

2.1.1 Méthode de mise en correspondance de blocs

Le principe des méthodes fondées sur la mise en correspondance de blocs est simple. Il consiste à identifier un motif dans une image I et le retrouver dans l'image suivante $I(t + \Delta t)$ où Δt est le pas d'échantillonnage temporel. La différence entre la position de ce bloc dans l'image $I(t)$ et sa position dans l'image $I(t + \Delta t)$ représente le mouvement de ce bloc au cours de l'intervalle de temps Δt . Ce motif peut être un pixel ou un groupe de pixels. En général, ce groupe de pixels est une fenêtre de forme rectangulaire qui est appelée bloc. Ces méthodes portent le nom de "block matching" en anglais.

Il est très rare de pouvoir retrouver un bloc identique dans l'image suivante en raison de la déformation des objets, de leur occultation (partielle ou totale) par un autre ou du changement de l'illumination globale de la scène. On va alors chercher dans l'image $I(t + \Delta t)$, le bloc qui ressemble le plus au bloc de l'image $I(t)$. Afin de mesurer le niveau de ressemblance d'un bloc de l'image $I(t)$ dans l'image $I(t + \Delta t)$, plusieurs mesures peuvent être utilisées.

Le plus souvent, on utilise la mesure de la corrélation normalisée. Soit F_1 est un bloc de l'image $I(t)$ de dimension M lignes et N colonnes. Les dimensions M et N doivent être impaires. La corrélation normalisée entre ce bloc et un bloc F_2 centré sur le pixel (i, j) de l'image $I(t + \Delta t)$ est calculée par l'équation (2.3)

$$\mathcal{C}(i, j) = \frac{\sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} F_1(m + \frac{M}{2}, n + \frac{N}{2}) \cdot F_2(i + m, j + n)}{\sqrt{\sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} (F_1(m + \frac{M}{2}, n + \frac{N}{2}) - \bar{F}_1)^2 \cdot \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} (F_2(i + m, j + n) - \bar{F}_2)^2}} \quad (2.3)$$

où \bar{F}_1 et \bar{F}_2 désignent les moyennes des blocs respectifs F_1 et F_2 . En calculant cette mesure pour toutes les positions i, j de l'image, il est possible de créer une carte de corrélation telle que celle présentée en figure FIG 2.2. Sur cette carte de corrélation, le pic le plus important représente le centre du bloc ressemblant le plus au bloc F_1 , sur l'image $I(t + \Delta t)$. Le vecteur permettant de

passer de la position centrale du bloc F_1 dans l'image $I(t)$ à la position de ce pic représente le mouvement associé au bloc F_1 entre l'instant t et l'instant $t + \Delta t$.

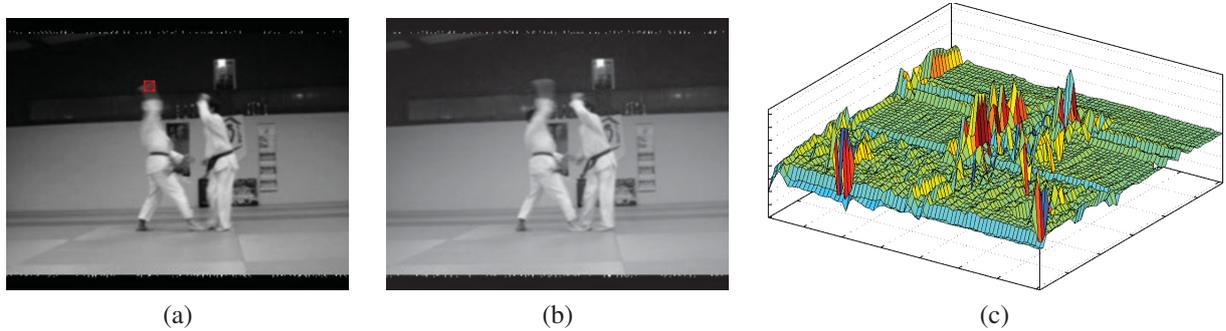


FIGURE 2.2 – Représentation d'une carte de corrélation ; (a) image à l'instant t , le rectangle rouge localise le bloc à rechercher ; (b) image à l'instant $t + \Delta t$ dans laquelle le bloc est recherché ; (c) Carte de corrélation obtenue par corrélation du bloc de l'image (a) avec l'image (b)

Cependant la complexité algorithmique de cette mesure est exprimée en $\Theta((M \times N)^2)$ pour le calcul sur un bloc et en $\Theta((M \times N)^2 \times C \times R)$ pour le calcul de la carte complète sur une image de dimension C lignes et R colonnes. Cela peut engendrer des temps de calculs longs selon les dimensions des fenêtres utilisées et sur des images de grandes dimensions. À cause de cela, la zone de recherche du maximum de corrélation est souvent limitée à une zone autour de la position centrale de la fenêtre F_1 dans I . Cette zone de recherche est dépendante de la dynamique du mouvement étudié ainsi que de la résolution temporelle (Δt).

Afin d'accélérer les calculs, il est possible d'utiliser la transformée de Fourier rapide. En effet, la corrélation de F_1 et F_2 peut se calculer par la formule suivante :

$$\mathcal{C} = FFT^{-1}(FFT(F_1) \otimes FFT(F_2)) \quad (2.4)$$

, où \mathcal{C} est la carte de convolution, FFT , pour Fast Fourier Transform, est la transformée de Fourier rapide d'un signal, FFT^{-1} sa transformée de Fourier rapide inverse et \otimes représente le produit terme à terme des valeurs (complexes) des deux matrices. De cette manière, la complexité de la méthode est de $\Theta(NM \log_2(NM))$.

D'autres mesures sont couramment utilisées telle que la différence absolue moyenne (MAD pour "Mean Absolute Difference") présentée en équation (2.5) et l'erreur quadratique moyenne (EQM) présentée par l'équation (2.6).

$$MAD = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |F_1(i, j) - F_2(i, j)| \quad (2.5)$$

$$EQM = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (F_1(i, j) - F_2(i, j))^2 \quad (2.6)$$

Les méthodes d'estimation du mouvement par mise en correspondance de blocs sont principalement utilisées dans deux domaines d'applications. Le premier concerne l'estimation du mouvement permettant la compensation de mouvement des vidéos. Le second est l'estimation du mouvement par vélocimétrie d'image de particules (PIV pour "Particle Image Velocimetry").

Dans chaque domaine, certains ajustements ont été appliqués à la méthode afin de répondre à la problématique particulière du domaine.

2.1.1.1 La Compensation du mouvement pour la compression de vidéos

La compensation en mouvement est une étape de la compression vidéo que l'on retrouve dans toutes les spécifications de la norme `mpeg` ainsi que dans la norme `H.264` qui sont les formats de compression les plus utilisés. Cette technique consiste à estimer le mouvement qui permet de passer d'une image t à l'image $t + \Delta t$. Une fois ce mouvement estimé, il est codé de manière efficace et stocké dans le flux de la vidéo à la place de l'image $t + \Delta t$. La qualité du champ estimé est évaluée indirectement en mesurant la qualité des images prédites à partir du mouvement et de l'image précédente. Très souvent, cette qualité est mesurée par le PSNR entre l'image originale et l'image prédite.

$$PSNR = 10 \times \log_{10} \frac{d}{EQM} \quad (2.7)$$

d représente la dynamique du signal, c'est-à-dire la valeur maximale que peut prendre un pixel. Le résidu entre l'image prédite à partir de l'image t et l'image $t + \Delta t$ originale est lui aussi stocké afin de conserver une bonne qualité d'image. Dans le cas de la compensation du mouvement, la principale contrainte est liée au temps. En effet, il est souhaitable que cette étape ne soit pas trop longue afin que le codeur puisse compresser des vidéos en temps réel. Les mesures utilisées pour comparer deux blocs dans les applications de compensation du mouvement sont la MAD (cf. (2.5)) ou l'EQM (cf. (2.6)). De même, afin d'accélérer les calculs, une zone de recherche dans l'image $t + \Delta t$ est définie autour de la position centrale du bloc F_1 de l'image t . Cette zone est souvent carrée. Un paramètre de recherche p (exprimé en pixels) définit la zone de recherche de part et d'autre du centre du bloc F_1 . De plus, de nombreux algorithmes ont été proposés afin d'obtenir une estimation du mouvement de qualité tout en limitant le nombre de blocs à comparer.

BARJATYA [Barjatya 07] proposent une comparaison de plusieurs algorithmes de mise en correspondance de blocs (MCB) dont ceux que nous présentons ci-dessous.

Le premier algorithme proposé se révélant plus efficace en temps de calcul que la méthode exhaustive est l'algorithme de recherche en trois étapes ("Three Step Search (TSS)"). Il date du milieu des années 80 [Koga 85]. Nous illustrons cet algorithme en figure FIG 2.3. Pour un paramètre de recherche p donné, on fait évoluer à chaque étape un paramètre de recherche par étape S . Si par exemple l'initialisation est effectuée avec $p = 7$, alors l'algorithme définit $S = \lceil \frac{p}{2} \rceil$ pour la première étape, avec $\lceil \frac{p}{2} \rceil$ la partie entière de $\frac{p}{2}$. On cherche alors dans les huit directions autour du centre, à la distance S , le bloc qui minimise l'erreur (MSE ou MAD). Puis, on place le nouveau centre à cette position. On effectue une nouvelle recherche cette fois-ci avec $S = S/2$. Enfin, dans une troisième étape, on cherche le minimum pour $S = 1$. Cet algorithme permet de réduire d'un facteur 9 les temps de calcul par rapport à l'algorithme de recherche exhaustive.

Cet algorithme a été amélioré ("New Three Step Search (NTSS)") [Li 94] afin de permettre la détection des mouvements de faible amplitude. Plusieurs autres algorithmes ont été proposés [Po 96],[Lu 97],[Zhu 97] pour être plus efficaces en temps de calcul.

Dans [Nie 02], le mouvement global entre deux images est considéré comme cohérent. Cela signifie que statistiquement, un bloc est animé du même mouvement que ceux qui l'entourent.

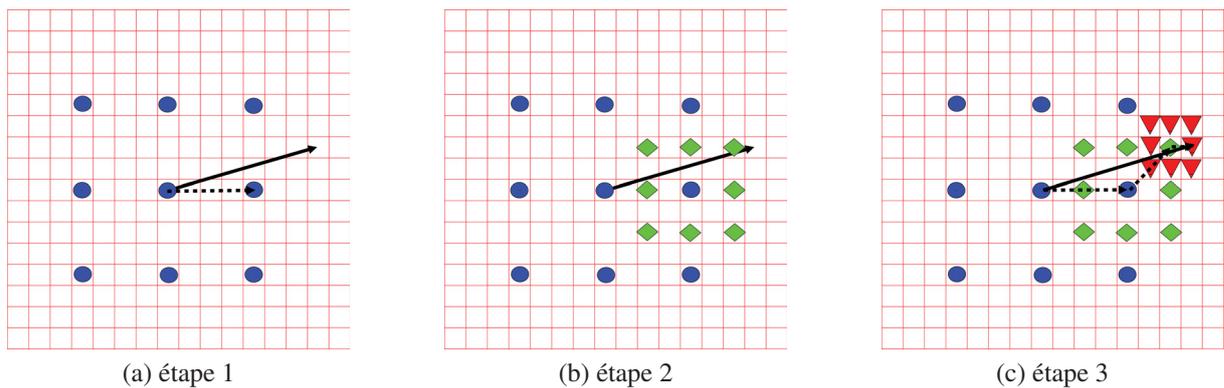


FIGURE 2.3 – Illustration de l'algorithme de recherche à 3 étapes.



FIGURE 2.4 – Illustration de l'algorithme adaptatif de recherche en croix.

Cela permet de déterminer une zone de recherche en fonction du vecteur directement à gauche du bloc courant. Ce vecteur est appelé vecteur prédictif (cf. figure FIG 2.4). La zone de mouvement est représentée par le maximum des directions du vecteur prédictif. Ensuite, autour de la position donnant le MAD minimum, on effectue une recherche avec un motif en diamant de petite taille, $S = 1$, comme pour l'algorithme de recherche en diamant [Zhu 97]. L'utilisation de cet algorithme dans le contexte de la compensation du mouvement pour la compression de vidéo permet d'obtenir, avec un coût calculatoire très faible, des images prédites ayant un taux de PSNR équivalent à celles obtenues avec l'algorithme de recherche exhaustive.

2.1.1.2 La vélocimétrie par imagerie de particules

La vélocimétrie par imagerie de particules est une méthode utilisée pour l'acquisition de champs de déplacements en analyse des mouvements fluides.

Le principe est d'ajouter à un écoulement de fines particules qui vont être éclairées avec un laser. Ces particules vont renvoyer de la lumière vers un système d'acquisition d'images ayant une fréquence d'acquisition suffisamment élevée pour capter le mouvement du fluide entre deux images. Ensuite, par des méthodes de mise en correspondance de blocs, on extrait le déplacement. Actuellement, la majorité des techniques PIV utilise des méthodes de corrélation,

nous nous limitons donc à cet aspect. Un état de l'art très complet des méthodes PIV est présenté dans [David 05].

La mesure de vraisemblance utilisée est souvent la mesure de corrélation normalisée présentée en équation (2.2). Dans certains cas, les fenêtres utilisées pour la mesure de corrélation sont déformées afin de retrouver plus précisément le mouvement.

Une autre approche a été proposée en 2002 dans [Susset 02]. L'objectif est d'accélérer les calculs de corrélation. Cette méthode repose sur l'utilisation d'une structure de données particulière pour le stockage. Cette structure de données est organisée comme suit : on ordonne les niveaux de gris du plus faible au plus élevé, à chaque case de ce tableau est associée une liste des positions en i et j où cette valeur est prise. Ensuite une mesure de corrélation directe est utilisée directement sur cette structure. Cela permet d'obtenir un gain en temps de calcul.

La mesure du mouvement par PIV ayant pour objectif l'analyse de phénomènes physiques réels et souvent complexes, le mouvement se doit d'être estimé avec précision. Les mesures de corrélation étant effectuées sur des données discrètes, la localisation d'un pic dans la carte de corrélation a donc une précision de ± 0.5 pixels. Pour résoudre ce problème, une solution est d'envelopper la zone entourant le pic de corrélation par une fonction continue [David 05] et de détecter le maximum de cette fonction (barycentre, Gaussienne ou parabolique).

2.1.2 Méthodes fondées sur l'équation du flot optique

Les méthodes fondées sur l'équation du flot optique s'appuient sur l'hypothèse que la luminosité est conservée entre une image et celle qui la suit. Il est alors possible de poser l'**eq.2.8** appelée "équation du flot optique".

$$\begin{aligned} \frac{d\mathcal{S}(x_1, x_2, t)}{dt} &= \frac{d\mathcal{S}(\vec{x}, t)}{dt} = 0 \\ \Leftrightarrow \partial_{x_1}\mathcal{S}(\vec{x}, t) \cdot \frac{dx_1}{dt} + \partial_{x_2}\mathcal{S}(\vec{x}, t) \cdot \frac{dx_2}{dt} + \partial_t\mathcal{S}(\vec{x}, t) \cdot \frac{dt}{dt} &= 0 \end{aligned} \quad (2.8)$$

Or $\frac{dx_1}{dt}$ et $\frac{dx_2}{dt}$ correspondent respectivement au déplacement dans le temps selon la première et la deuxième composante. On peut alors écrire :

$$\partial_{x_1}\mathcal{S}(\vec{x}, t) \cdot \mathcal{U}(\vec{x}) + \partial_{x_2}\mathcal{S}(\vec{x}, t) \cdot \mathcal{V}(\vec{x}) + \partial_t\mathcal{S}(\vec{x}, t) = 0 \quad (2.9)$$

On peut aussi l'écrire sous forme vectorielle :

$$\nabla_{\Omega}\mathcal{S}(\vec{x}, t) \cdot \vec{\mathcal{C}}(\vec{x}) + \partial_t\mathcal{S}(\vec{x}, t) = 0 \quad \text{avec} \quad \vec{\mathcal{C}}(\vec{x}) = (\mathcal{U}(\vec{x}) \ \mathcal{V}(\vec{x})) \quad (2.10)$$

où ∇_{Ω} représente l'opérateur gradient spatial.

On peut montrer que trouver $\vec{\mathcal{C}}$ qui annule (2.10) revient à trouver $\vec{\mathcal{C}}$ qui minimise la fonctionnelle :

$$\int_{\Omega} (\nabla_{\Omega}\mathcal{S}(\vec{x}, t) \cdot \vec{\mathcal{C}}(\vec{x}) + \partial_t\mathcal{S}(\vec{x}, t))^2 d\Omega = 0 \quad (2.11)$$

Malheureusement, cette équation ne suffit pas à résoudre le problème posé, puisque l'on cherche à déterminer deux inconnues $\mathcal{U}(\vec{x})$ et $\mathcal{V}(\vec{x})$ à partir d'une seule équation. À un point correspond alors une droite représentative de l'ensemble des solutions dans l'espace $(\mathcal{U}(\vec{x}), \mathcal{V}(\vec{x}))$.

Un autre problème est celui connu sous le nom de problème d'ouverture. Un exemple en est donné figure FIG 2.5 où l'on illustre les trois cas qui peuvent se présenter dans un voisinage donné. Dans le premier cas (a), il n'y a pas d'information. On ne peut donc pas extraire de champ de déplacements. Dans le second cas (b), nous sommes confrontés au problème d'ouverture et on ne peut donc calculer que le flot normal au contour. Le troisième cas (c) où nous avons suffisamment d'informations pour calculer le flot.

Lorsque le problème d'ouverture est présent, seule la composante normale au contour peut être extraite :

$$\vec{c}(\vec{x}) = -\frac{\partial_t \mathcal{S}(\vec{x}, t)}{\|\nabla_{\Omega} \mathcal{S}(\vec{x}, t)\|} \cdot \frac{\nabla_{\Omega} \mathcal{S}(\vec{x}, t)}{\|\nabla_{\Omega} \mathcal{S}(\vec{x}, t)\|} \quad (2.12)$$

La plupart des algorithmes d'extraction de flot optique essaient de résoudre ce problème en introduisant de nouvelles contraintes.

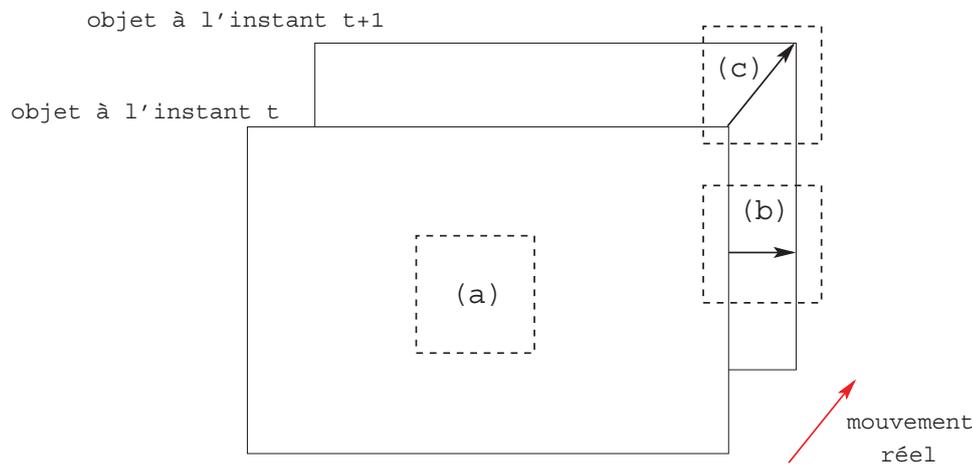


FIGURE 2.5 – Les trois cas possibles lors de l'extraction du flot. (a) aucune information disponible : pas de flot. (b) problème d'ouverture : on ne peut calculer que le flot normal au contour. (c) suffisamment d'informations : le flot calculé est correct.

HORN et SCHUNCK [Horn 81] ont été les premiers à proposer une solution permettant de résoudre le problème du nombre d'inconnues issues de l'équation du flot optique. Pour cela, ils ont introduit une contrainte concernant la cohérence spatiale du champ de déplacements qui permet de choisir le déplacement le plus faible. Le but est alors de minimiser la fonctionnelle :

$$\int_{\Omega} (\nabla_{\Omega} \mathcal{S} \cdot \vec{c} + \mathcal{S}_t)^2 + \lambda^2 (\|\nabla \mathcal{U}\|^2 + \|\nabla \mathcal{V}\|^2) \cdot d\Omega \quad (2.13)$$

avec λ un coefficient qui permet de faire un compromis entre les deux contraintes.

L'algorithme proposé est itératif et simple à mettre en œuvre. De plus, HORN et SCHUNCK ont démontré qu'il converge uniformément vers la solution. À tout moment, on peut donc arrêter l'algorithme et affirmer que la dernière solution obtenue est au moins équivalente (sinon meilleure) aux solutions déjà calculées. Depuis, on trouve dans la littérature de nombreux autres travaux issues de ce travail fondateur [Black 96a], [Weickert 06], [Ben-Ari 09].

LUCAS et KANADE [Lucas 81a] ont proposé une technique consistant à augmenter le nombre d'équations pour un pixel en considérant son voisinage V . Cela revient à considérer que

les pixels d'un même voisinage ont le même vecteur déplacement. Si le problème d'ouverture est toujours présent, LUCAS et KANADE proposent de recommencer le processus en prenant un voisinage plus important. L'algorithme offre l'avantage d'être rapide [Galvin 98]. C'est pourquoi certains travaux exploitent cette méthode en temps réel grâce à une programmation parallèle sur processeurs graphiques [Le Besnerais 05].

D'autres auteurs se sont intéressés au choix de l'estimateur utilisé. BLACK et ANANDAN [Black 96a] ont démontré que lorsque plusieurs mouvements sont présents, l'estimation au sens des moindres carrés n'est pas satisfaisante. Ils ont alors proposé l'utilisation d'estimateurs robustes en utilisant des normes différentes de la norme quadratique. Ils proposent d'utiliser pour le calcul du flot optique des fonctions robustes (ρ -fonctions), telles que les fonctions de GEMAN et MCCLURE ou les fonctions de LORENTZ. L'utilisation des fonctions robustes permet d'utiliser les mêmes méthodes d'estimation qu'avec la méthode des moindres carrés et d'être moins sensible aux valeurs aberrantes du champ. Les estimateurs robustes ont été utilisés dans [Mémmin 02]. Les auteurs réalisent conjointement l'estimation du mouvement et la segmentation du mouvement par une méthode hiérarchique.

Pour rajouter des contraintes à l'équation du flot optique, certains auteurs imposent des contraintes sur le champ de déplacements recherché. Ils peuvent alors utiliser une approche plus globale en utilisant un modèle paramétrique du champ de déplacements [Odobez 95], [Bruno 00], [Bruno 01], [Nir 08]. Le modèle utilisé est souvent un modèle affine de degré 1.

Une autre méthode d'extraction du mouvement est fondée sur l'étude des tenseurs de structure [Bigun 91] et [Jahne 93], [Augereau 05], [Druon 06]. Un tenseur de structure correspond simplement au produit du vecteur-gradient par son vecteur transposé. Aussi, si l'on se place dans un espace à trois dimensions alors le tenseur de structure correspondant est une matrice 3x3 réelle symétrique.

Soit $\nabla\mathcal{S}(\vec{x}, t) = (\partial_{x_1}\mathcal{S}(\vec{x}, t), \partial_{x_2}\mathcal{S}(\vec{x}, t), \partial_{x_3}\mathcal{S}(\vec{x}, t))^T$, alors le tenseur de structure T est obtenu par :

$$T(\mathcal{S}(\vec{x}, t)) = \nabla\mathcal{S}(\vec{x}, t) \cdot \nabla\mathcal{S}(\vec{x}, t)^T \quad (2.14)$$

Pour simplifier l'écriture, dans l'équation (2.15) \mathcal{S} signifie $\mathcal{S}(\vec{x}, t)$.

$$T(\mathcal{S}(\vec{x}, t)) = \begin{pmatrix} (\partial_{x_1}\mathcal{S})^2 & \partial_{x_1}\mathcal{S}\partial_{x_2}\mathcal{S} & \partial_{x_1}\mathcal{S}\partial_{x_3}\mathcal{S} \\ \partial_{x_1}\mathcal{S}\partial_{x_2}\mathcal{S} & (\partial_{x_2}\mathcal{S})^2 & \partial_{x_2}\mathcal{S}\partial_{x_3}\mathcal{S} \\ \partial_{x_1}\mathcal{S}\partial_{x_3}\mathcal{S} & \partial_{x_2}\mathcal{S}\partial_{x_3}\mathcal{S} & (\partial_{x_3}\mathcal{S})^2 \end{pmatrix} \quad (2.15)$$

Au final, on a donc, pour chaque image, un tenseur par pixel. Comme cette méthode est basée sur des éléments différentiels, elle est très sensible au bruit. Afin de limiter les effets du bruit, on réalise un moyennage des tenseurs de structure sur un voisinage. À partir de ce tenseur, on calcule ses valeurs propres $\lambda_1, \lambda_2, \lambda_3$ et ses vecteurs propres associés $\vec{v}_1 = (x_1, y_1, z_1)^T$, $\vec{v}_2 = (x_2, y_2, z_2)^T$, $\vec{v}_3 = (x_3, y_3, z_3)^T$. L'étude de ces valeurs propres triées par ordre décroissant ($\lambda_1 > \lambda_2 > \lambda_3$) permet de déterminer le mouvement de ce pixel. Il est possible d'identifier les situations suivantes : soit il n'y a pas de mouvement apparent, soit on se trouve devant le problème d'ouverture, soit il est possible d'identifier un mouvement cohérent, soit il n'y a pas de mouvement cohérent. En ne calculant les vecteurs du champ que lorsque le mouvement est cohérent, cette méthode fournit des champs épars. Le mouvement est cohérent sur les contours les plus importants de l'image.

Suivant la technique d'estimation du mouvement choisie, il est possible d'obtenir des champs de déplacements de densités différentes à partir des mêmes images originales. Les méthodes par mise en correspondance de blocs fournissent des champs denses, mais très sou-

vent le champ de déplacements est sous échantillonné par rapport aux images. Les méthodes de flot optique permettent d'obtenir des champs denses de mêmes dimensions que les images, à l'exception des méthodes reposant sur le tenseur de structure pour lesquelles, en général, on ne calcule le champ que lorsque le mouvement est cohérent. On obtient alors des champs épars. Dans tous les cas, un seuillage peut aussi être appliqué au champ de déplacements afin d'annuler tous les vecteurs de trop faible amplitude. Il est alors possible d'envisager différentes méthodes de modélisation en fonction de la densité du champ à modéliser.

2.2 Utilisation des champs de déplacements

Après avoir présenté quelques algorithmes de référence permettant d'obtenir un champ de déplacements à partir d'une séquence d'images, nous présentons ici des applications qui utilisent ces champs de déplacements.

2.2.1 La compensation du mouvement

La compensation du mouvement est une étape importante des formats de compression mpeg, H.263 ou H.264. L'objectif est d'utiliser la redondance temporelle des vidéos pour ne pas coder intégralement toutes les images. En effet, entre une image à l'instant t et la suivante à l'instant $t + 1$, les éléments présents dans l'image sont souvent les mêmes. La différence est souvent due au déplacement d'un objet de l'image. Le principe est donc d'estimer le mouvement entre deux images successives. En appliquant ce mouvement à la première image, on obtient une image très proche de la seconde. Il suffit alors de coder la première image et le mouvement entre les images suivantes ou précédentes. Les images obtenues par compensation peuvent être calculées à partir de l'image précédente (image P) ou à partir de l'image précédente et de l'image suivante (image B). Les images B sont les plus efficaces du point de vue de la redondance temporelle. En effet, un objet de l'image B peut ne pas encore se trouver dans l'image précédente, mais être présent dans l'image suivante. Cependant, ces images obligent à réorganiser le flux vidéo afin que le décodage soit possible. Finalement, de manière à ce que la qualité des images prédites ne soit pas trop dégradée, le résidu entre une image prédite et l'image originale (que l'on cherche à prédire) est aussi codé.

Les codeurs vidéo n'imposent aucune méthode d'estimation du mouvement. Cependant, ce sont presque toujours des méthodes de mise en correspondance de blocs qui sont utilisées. La taille des blocs varie de 8×8 à 16×16 pixels. Il est possible de représenter le mouvement de compensation sous la forme d'un champ de vecteurs épars. La figure FIG 2.6 présente deux images en (a) et (b) ainsi que le champ de déplacements en (c) permettant d'obtenir l'image (b) par compensation du mouvement de l'image (a). Le champ est superposé à l'image prédite.

Le mouvement estimé pour la compensation du mouvement a vocation à permettre une compression de l'information en réduisant la redondance temporelle des vidéos. Pour les applications à la compression de vidéos, le champ de déplacements n'est pas modélisé. Il est transmis directement dans le flux vidéo. Cependant, cette information de mouvement peut ensuite être utilisée par d'autres applications, par exemple, pour la segmentation ou la classification de vidéo. Dans ce cas, il est nécessaire d'utiliser un outil de modélisation afin d'analyser l'information contenue dans le champ de déplacements.

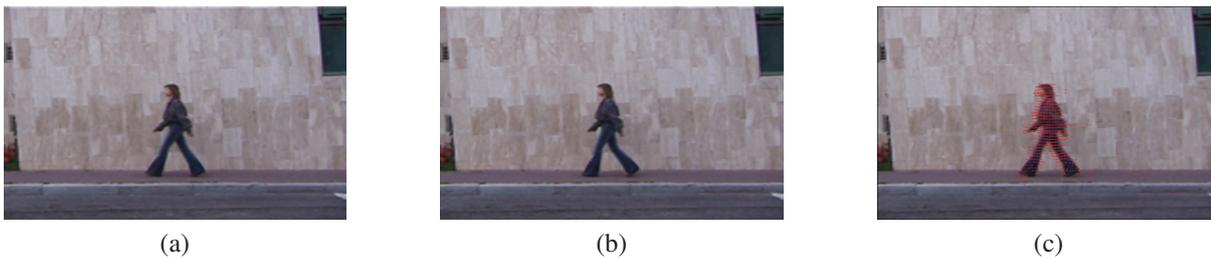


FIGURE 2.6 – Compensation du mouvement sur deux images d’un flux mpeg issu de la base de Weizmann [Blank 05] : (a) Image à $t = 42$; (b) Image à $t = 43$ estimée à partir de (a) et du mouvement ; (c) Mouvement entre (a) et (b) superposé avec (a) ;

2.2.2 L’étude des mouvements de caméra

Lorsqu’une scène est filmée avec une caméra vidéo, il est possible que la caméra ne soit pas fixe pour des raisons techniques ou par souci de composition artistique. Ces mouvements de la caméra ainsi que les zooms, font apparaître dans la séquence vidéo ce que l’on qualifie de “mouvement de caméra”. Il s’agit de tous les mouvements qui ne sont pas produits par les sujets ou les objets de la scène filmée, c’est-à-dire, les “travelling” (horizontaux ou verticaux), le zoom (positif ou négatif) ou la rotation.

Après extraction des champs de déplacements entre les images de la séquence vidéo, il est possible de reconnaître les mouvements de caméra. Ce domaine a fait l’objet d’un grand nombre de publications [Odobez 95], [Bouthemy 99], [Durik 01], [Ngo 02], [Park 04], [Smeaton 08]. Les mouvements de caméra sont souvent modélisés par des modèles paramétriques. Les principaux modèles paramétriques utilisés sont les modèles Constant, Affine, Bilinéaire, Pseudo-perceptif et Biquadratique.

Les mouvements de caméra peuvent être utilisés dans une chaîne d’indexation vidéo. L’information portée par ce mouvement peut servir notamment lors de la phase de séparation des plans d’une vidéo. Par exemple, dans [Durik 01] le champ de déplacements issu du flux mpeg est utilisé afin de caractériser le mouvement de caméra. Cette information est ensuite utilisée pour segmenter la vidéo.

Le mouvement de caméra est parfois appelé mouvement dominant puisque ce mouvement peut couvrir le mouvement des sujets de la scène. C’est pourquoi il est parfois utile d’annuler le mouvement de caméra afin de stabiliser une vidéo. Cela est particulièrement utile lorsque l’on cherche à détecter des objets en mouvement dans des séquences filmées avec une caméra mobile. En effet, dans ce cas, le mouvement de la caméra rend la détection des mouvements des objets plus difficile. Dans [Odobez 95] deux algorithmes multirésolutions d’estimation du mouvement de caméra (par un modèle paramétrique) sont proposés. Ces deux algorithmes sont intégrés dans le logiciel Motion2D [Inria]. Ce logiciel permet l’estimation et la compensation du mouvement dominant dans les vidéos. La figure FIG 2.7 représente deux images, en (a) et (b), issues d’une vidéo à deux instants différents. Dans cette vidéo, une personne effectue des mouvements sur place. Pendant l’acquisition, la caméra effectue un zoom. Le mouvement de caméra (zoom) est annulé avec le logiciel Motion2D. On obtient alors les images (c) et (d) correspondant respectivement aux images (a) et (b) après suppression du mouvement.



FIGURE 2.7 – Compensation du mouvement : (a) Image de la vidéo originale à $t = 0$; (b) Image de la vidéo originale à $t = 20$; (c) Image de la vidéo après suppression du mouvement de caméra à $t = 0$; (d) Image de la vidéo après suppression du mouvement de caméra à $t = 20$.

2.2.3 La segmentation d'images basée mouvement

La segmentation d'images au sens du mouvement consiste à partitionner une image en régions représentant chacune un mouvement différent. C'est une étape importante dans un grand nombre d'applications telles que l'analyse de scènes vidéo, la détection d'obstacles [Demonceaux 04] ou la vidéo-surveillance automatique. Il existe principalement trois familles de méthodes pour effectuer la segmentation au sens du mouvement. La première famille, dite séquentielle, consiste à estimer le mouvement dominant sur l'image globale [Irani 92] [Ayer 94]. Les pixels où le mouvement correspond au mouvement dominant forment une région. Ensuite, l'estimation du mouvement dominant est effectuée sur chaque région n'appartenant pas à cette première région. Le procédé se répète jusqu'à ce que tous les pixels soient associés à une région. La deuxième famille, dite par regroupement de régions, commence par segmenter l'image par une segmentation spatiale [Wang 94], [Altunbasak 98], [Moscheni 98] et [Gelgon 00]. Les critères sont définis à partir de l'intensité lumineuse, la couleur ou la texture des pixels ou des groupes de pixels. Ensuite, le mouvement de chaque région est estimé avec un modèle (souvent paramétrique 2D). Les régions ayant un mouvement similaire sont alors regroupées. La troisième famille réalise conjointement la segmentation au sens du mouvement et l'estimation du mouvement lui-même [Odobez 98], [Mémin 98], [Cremers 03].

2.2.4 La reconnaissance de mouvements

Une étude menée dans les années 70 par JOHANSSON [JOHANSSON 73] a montré qu'il est possible pour un humain de reconnaître une action à partir de l'information dynamique du mouvement. Cette expérience montre donc l'importance d'utiliser le mouvement d'une séquence d'image afin de reconnaître une action humaine.

La reconnaissance d'action humaine telle que "marcher" ou "courir" est d'une grande importance pour les systèmes de vidéo surveillance automatiques. À notre connaissance, les plus anciens travaux dans ce domaine ont été effectués par POLONA et NELSON [Polana 94]. Ils utilisent des champs obtenus par flot optique pour reconnaître des activités relatives à des mouvements répétitifs tels que la marche ou la course. Pour cela, ils modélisent le champ de déplacements par l'amplitude du mouvement dans chaque case d'une grille découpant le domaine spatial de manière grossière. Ensuite, un mouvement est reconnu par une méthode statistique utilisant la technique des *k* plus proches voisins. Plus récemment, en 2009, CHAUDHRY et AL. [Chaudhry 09] utilisent des histogrammes de l'orientation du flot optique pour classifier 10 types d'actions humaines comme "marcher", "courir" ...

La reconnaissance d'action humaine peut aussi intervenir pour des applications d'aide à la personne. Par exemple, dans [Snoek 09], le mouvement est utilisé pour détecter des événements inhabituels dans les escaliers. Le mouvement est calculé après une segmentation de l'image par une méthode de soustraction d'arrière plan. Sur cette image segmentée, le mouvement de la personne utilisant les escaliers est estimé par un modèle affine. L'analyse du mouvement est effectué en modélisant l'évolution des mouvements affines avec un modèle de Markov caché. Les mouvements inhabituels tels que les chutes sont alors détectés.

Nous présentons au chapitre 8, un état de l'art des méthodes d'analyse du mouvement dédiées à la reconnaissance d'actions humaines. Une attention particulière est portée aux méthodes qui utilisent les champs de déplacements. Nous y présentons aussi une nouvelle méthode de reconnaissance d'actions humaines.

2.2.5 L'analyse des champs de déplacements en analyse de mouvement fluide

En analyse expérimentale des mouvements fluides, de nombreux phénomènes physiques peuvent être analysés à partir des champs de déplacements représentant le mouvement instantané d'un fluide. L'importance de l'information portée par les champs de déplacements a poussé les chercheurs à développer des méthodes de mise en correspondance de blocs (PIV) dédiées aux écoulements fluides. L'analyse des champs de déplacements permet de comprendre ou de détecter certains phénomènes physiques. L'intérêt est, par exemple, de proposer des techniques de contrôles ou d'asservissement sur des écoulements.

Dans ce but, les champs peuvent être modélisés par une méthode de décomposition en valeurs singulières [Cordier 02] appelée POD pour Proper Orthogonal Decomposition. À partir des coefficients de ce modèle, il est possible d'identifier certaines propriétés d'un écoulement. Par exemple, en traçant l'évolution temporelle des deuxième et troisième coefficients de ce modèle, il est possible de savoir si un écoulement est périodique. Cette analyse est appelée portrait de phase. La figure FIG 3.4 représente deux exemples. Le premier graphique représente un écoulement périodique, tandis que le second graphique représente un écoulement de type cycle limite¹.

Il peut aussi être très utile de pouvoir détecter certaines particularités locales du champ telles que des tourbillons, des sources, des puits... Plusieurs méthodes permettent de détecter ces points. Par exemple, RAO et JAIN [Rao 92a] proposent d'utiliser un modèle affine afin de

1. En mathématiques, dans l'étude des systèmes dynamiques, on appelle cycle limite, ou cycle-limite sur un plan ou une variété bidimensionnelle, une trajectoire fermée dans l'espace des phases, telle qu'au moins une autre trajectoire spirale à l'intérieur lorsque le temps tend vers $\pm \infty$

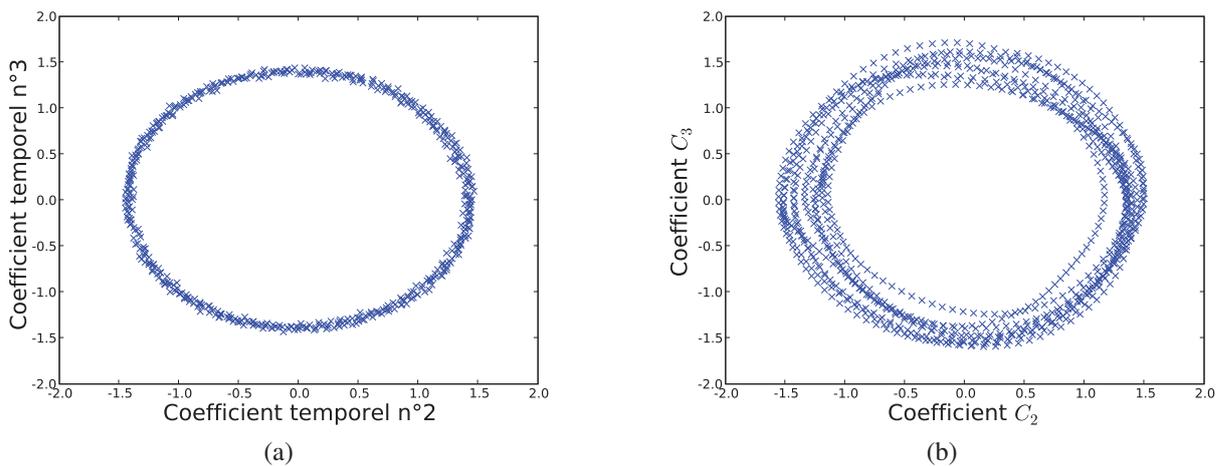


FIGURE 2.8 – Analyse d'écoulement à partir de coefficients POD ; (a) portrait de phase entre les deuxième et troisième coefficients temporels POD d'un écoulement périodique ; (b) portrait de phase entre les deuxième et troisième coefficients temporels POD d'un écoulement de type cycle limite. Figures extraites de [Druon 09]

détecter et de classifier les singularités d'un champ de déplacements. La figure 2.9 présente un exemple de champ dans lequel les singularités ont été identifiées. Nous présentons au chapitre 7 un état de l'art des méthodes de détection de singularités. Nous présentons aussi trois nouveaux algorithmes multi-résolutions pour la détection de singularités.

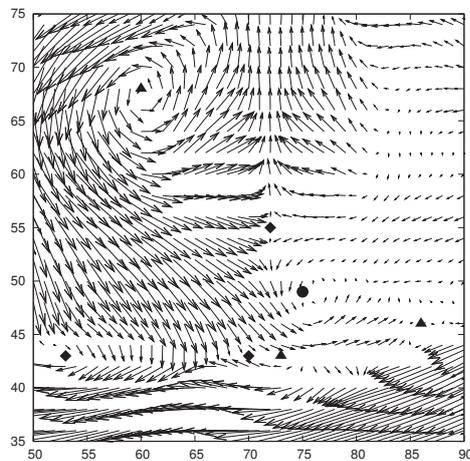


FIGURE 2.9 – Détection de singularités dans un champ de déplacements ; Méthode de détection présentée au chapitre 7 ; les ▲ représentent les tourbillons ; les ◆ représentent les points de selles ; les ● représentent les noeuds impropres et étoilés ; Ces différentes structures sont détaillées sur la figure FIG 7.1 au chapitre 7

2.3 Outils de modélisation en traitement d'image

L'information dynamique contenue dans les champs de déplacements est donc utilisée dans un grand nombre d'applications. Dans tous les cas, un modèle est utilisé pour permettre d'en

analyser le mouvement. Nous présentons maintenant les principaux outils de modélisation utilisés en traitement d'images.

2.3.1 Décomposition en valeurs singulières

La décomposition en valeurs singulières (SVD pour Singular Value Decomposition) est une méthode de factorisation de matrices rectangulaires. La SVD a été découverte indépendamment par EUGENIO BELTRAMI en 1873 et CAMILLE JORDAN en 1874. La première preuve de la SVD pour les matrices rectangulaires complexes a été publiée en 1936 dans [Eckart 36].

Soit A une matrice rectangulaire de dimension $m \times n$ et de rang r , alors il existe U , V et Σ telles que

$$A = U\Sigma V^T \quad (2.16)$$

où U est une matrice orthogonale d'ordre $m \times m$, V une matrice orthogonale d'ordre $n \times n$ et Σ une matrice pseudo-diagonale² de dimension $m \times n$.

La matrice U est obtenue en calculant les valeurs et vecteurs propres de la matrice AA^T . Ces vecteurs propres sont ordonnés dans l'ordre décroissant des valeurs propres. La matrice V est obtenue en calculant les vecteurs propres de la matrice $A^T A$. Ces vecteurs propres sont ordonnés en fonction du classement dans l'ordre décroissant de leur valeur propre associée. La diagonale de la matrice Σ est composée des valeurs propres de la matrice $A^T A$ ordonnées dans l'ordre décroissant. Les autres éléments de la matrice sont tous nuls.

Il est alors possible d'obtenir une modélisation à l'ordre p en ne conservant que les p premiers vecteurs propres de U et de V .



FIGURE 2.10 – Illustration de la modélisation d'une image par SVD ; (a) Image lena originale 512*512 pixels ; (b) SVD de lena avec les 5 premiers vecteurs propres conservés ; (c) SVD de lena avec les 50 premiers vecteurs propres conservés.

Nous donnons, en figure FIG 2.10, une illustration de la décomposition d'une image par SVD sur l'image Lena de dimensions 512×512 pixels. Nous présentons trois exemples avec respectivement 5 et 50 vecteurs propres conservés pour la reconstruction. Cependant bien que l'on puisse constater que la représentation de l'image est bonne, la représentation d'une image par SVD pour des applications de compression se révèle inefficace. La raison est qu'il faut

2. Le terme "pseudo-diagonale" signifie que tous les éléments hors de la diagonale sont nuls, mais que la matrice n'est pas carrée.

conserver aussi les vecteurs propres pour décompresser l'image. Comme ces vecteurs dépendent des données, il faut donc les stocker, ce qui implique un faible gain en compression.

La SVD est utilisée dans [Poelman 94] afin d'extraire conjointement les contours et le mouvement d'objets dans des séquences d'images.

Plus récemment, dans [Vasilescu 02], une SVD particulière appelée dans l'article "N-mode SVD" est utilisée pour extraire des signatures de mouvements humains. Ces signatures sont utilisées pour analyser et reconnaître des personnes et des actions. L'acquisition du mouvement est effectuée par 4 caméras et 18 marqueurs placés sur les personnes.

Dans le domaine de l'analyse d'écoulement, la SVD est très couramment utilisée dans toutes les techniques de Décomposition Orthogonale aux valeurs Propres ("Proper Orthogonal Decomposition" - POD). Un état de l'art sur l'utilisation de la POD dans le domaine de l'analyse d'écoulements fluides est présenté dans [Cordier 02].

Notons que la transformée de Karhunen-Loève et l'Analyse en Composantes Principales (ACP) sont très proches de la SVD.

2.3.2 Séries et transformée de Fourier

Les séries de Fourier sont utilisées pour analyser un signal périodique tandis que la transformée de Fourier permet une analyse similaire mais généralisée au cas de signaux non nécessairement périodiques.

Soit $f(x)$ un signal périodique, de période fondamentale T_0 ($T_0 = 1/\nu_0$, où ν_0 est la fréquence de la fondamentale), continu par morceau et monotone sur $[-T_0/2, T_0/2]$. Si pour tous les points de discontinuité, les limites de part et d'autre de la singularité existent, alors la série de Fourier modélisant ce signal est définie par :

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n(f) e^{-i2\pi \frac{n}{T_0} x} \quad (2.17)$$

où les coefficients c_n sont calculés par :

$$c_n(f) = \frac{1}{T_0} \int_{-T_0/2}^{+T_0/2} f(x) e^{-i2\pi \frac{n}{T_0} x} dx \quad (2.18)$$

La transformée de Fourier est définie pour un signal 1D par l'équation suivante :

$$TF\{f(x)\} = \mathcal{F}(\nu) = \int_{-\infty}^{+\infty} f(x) e^{-i\nu x} dx \quad (2.19)$$

où ν est la fréquence.

La transformée de Fourier est inversible par la formule :

$$TF^{-1}\{\mathcal{F}(\nu)\} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathcal{F}(\nu) e^{+i\nu x} d\nu \quad (2.20)$$

Ces définitions peuvent être étendues au cas nD.

Les algorithmes rapides de transformée de Fourier ont permis de rendre cette transformée utilisable dans un grand nombre d'applications de traitement du signal et des images.

Notons que la transformée en cosinus discret (DCT pour "Discret Cosine Transform") est très proche de la transformée de Fourier. La grande différence entre ces deux transformées est que les coefficients issus d'une DCT sont réels tandis que ceux fournis par transformée de Fourier sont des complexes. La DCT est la transformée utilisée dans le format de compression JPEG.

2.3.2.1 Applications en analyse du mouvement

Les séries de Fourier sont utilisées dans [Bruno 00] pour estimer le mouvement. Chaque composante du champ de déplacement est modélisée à l'harmonique M, N par une série de Fourier sous la forme :

$$\begin{aligned}\tilde{\mathcal{U}}^{M,N}(x_1, x_2) &= \sum_{k=-M/2}^{M/2} \sum_{l=-N/2}^{N/2} c_{k,l}(\mathcal{U}) e^{i2\pi(\frac{k}{M} + \frac{l}{N})} \\ \tilde{\mathcal{V}}^{M,N}(x_1, x_2) &= \sum_{k=-M/2}^{M/2} \sum_{l=-N/2}^{N/2} c_{k,l}(\mathcal{V}) e^{i2\pi(\frac{k}{M} + \frac{l}{N})}\end{aligned}\quad (2.21)$$

Les modèles $\tilde{\mathcal{U}}^{M,N}(x_1, x_2)$ et $\tilde{\mathcal{V}}^{M,N}(x_1, x_2)$ sont alors injectés dans l'équation du flot optique (cf. équation (2.9)) pour obtenir :

$$\partial_{x_1} \mathcal{S} \cdot \tilde{\mathcal{U}}^{M,N}(x_1, x_2) + \partial_{x_2} \mathcal{S} \cdot \tilde{\mathcal{V}}^{M,N}(x_1, x_2) + \partial_{x_3} \mathcal{S} = 0 \quad (2.22)$$

Une fois posée sous forme matricielle, on constate que l'équation est surdimensionnée. Ce système est alors résolu en cherchant à minimiser la fonctionnelle définie en équation (2.23) où $\rho(-, \sigma)$ est un M-estimateur robuste.

$$E = \sum_{x_1, x_2 \in \Omega} \rho(I(x_1 + \mathcal{U}(x_1, x_2), x_2 + \mathcal{V}(x_1, x_2), t + 1) - I(x_1, x_2, t), \sigma) \quad (2.23)$$

Les paramètres permettant de minimiser cette fonctionnelle sont obtenus par une estimation robuste incrémentale qui peut être résolue par une méthode des moindres carrés itérés.

Le modèle de mouvement obtenu a $[2 \times (M + 1) \times (N + 1)]$ paramètres. Le modèle est ensuite utilisé pour reconnaître des activités humaines à l'aide d'une classification ascendante hiérarchique. Cependant, l'auteur montre que cette méthode n'est pas adaptée à l'estimation des mouvements affines. Il propose alors une nouvelle méthode dans [Bruno 01] fondée sur l'utilisation des ondelettes. La modélisation par ondelettes est introduite dans la section suivante.

2.3.3 La transformée en ondelettes

Si la transformée de Fourier analyse et détecte dans un signal la redondance et la périodicité, la transformée en ondelettes cherche à détecter les discontinuités. La transformée en ondelettes est un complément à l'analyse par séries ou transformée de Fourier.

Des présentations plus complètes et précises sont fournies dans des ouvrages de référence tels que [Meyer 90], [Daubechies 92], [Truchetet 98], [Mallat 99] ou [Mallat 09].

La transformée en ondelette continue d'un signal réel en une dimension est donnée par l'équation (2.24)

$$\mathcal{W}(f(a, b)) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi_{a,b}(t) dt \quad (2.24)$$

où $\psi_{a,b}$ est une fonction d'ondelette, a est le facteur d'échelle et b est le paramètre de translation. Les fonctions d'ondelettes $\psi_{a,b}$ forment une famille de fonctions analysantes et sont construites à partir d'une ondelette mère décrite par :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right) \quad (2.25)$$

Afin de pouvoir être une ondelette, une fonction doit être oscillante et satisfaire la propriété suivante

$$C_\psi = \int_0^{+\infty} |TF(\psi)(w)|^2 \frac{dw}{|\omega|} < \infty \quad (2.26)$$

Dans la pratique, cette condition est simplifiée par :

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (2.27)$$

Si cette conditions est vérifiée, alors le signal $f(t)$ peut être parfaitement reconstruit par

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{a^2} \mathcal{W}(f(a, b)) \psi_{a,b}(t) da db \quad (2.28)$$

Le facteur d'échelle a et le paramètre de translation b étant des réels, la transformée en ondelettes continue est redondante. Pour limiter ou supprimer cette redondance, il est possible de réaliser une discrétisation de la transformation en ondelettes, appelée transformée en ondelettes discrètes pour laquelle le facteur d'échelle a et le paramètre de translation b sont des entiers.

Les ondelettes sont principalement utilisées dans le cadre de l'analyse multi-résolutions. De plus, ces analyses sont généralement effectuées dans le domaine discret. Nous nous plaçons donc dans le cas des ondelettes discrètes. Une analyse multi-résolutions est une suite de sous-espaces vectoriels $V_j, j \in \mathbb{Z}$ de $L^2(\mathbb{R})$ telle que les propriétés (2.29) à (2.32) soit vérifiées.

$$\forall j \in \mathbb{Z}, V_j \subset V_{j-1} \quad (2.29)$$

$$\lim_{j \rightarrow +\infty} V_j = \{0\} \quad (2.30)$$

$$\lim_{j \rightarrow -\infty} V_j = L^2(\mathbb{R}) \quad (2.31)$$

$$\exists \varphi(t-n), n \in \mathbb{Z} \text{ une base orthonormale de } V_0 \text{ telle que } \varphi(t) \in L^2(\mathbb{R}) \quad (2.32)$$

La fonction $\varphi(x)$ est appelée fonction d'échelle mère et elle permet de construire, par translation de n , une base de fonctions du sous espace V_j

$$\varphi_{j,n}(x) = 2^{-\frac{j}{2}} \varphi(2^{-j}x - n) \text{ avec } n \in \mathbb{Z} \quad (2.33)$$

À chaque résolution est associée un espace vectoriel V_j et ces sous-espaces sont emboîtés les uns dans les autres ($V_i \subset V_j$ où i et j sont des entiers positifs tel que $i > j$). La base des fonctions d'échelles à la résolution j est obtenue par dilatation de l'ondelette à la résolution $j-1$ puis

par translation. L'approximation à la résolution j de la fonction $f(x)$ est donnée par la suite numérique a_n^j suivante

$$a_n^j = \int_{-\infty}^{+\infty} f(x)\varphi_{j,n}(x)dx \quad (2.34)$$

Cette approximation devient de plus en plus grossière lorsque la résolution j augmente.

À ces espaces vectoriels d'approximation V_j s'ajoutent des espaces vectoriels de détail W_j tels que

$$V_{j-1} = V_j \oplus W_j \quad (2.35)$$

Les coefficients q_n^j de détails (cf. équation (2.37)) qui constituent ces espaces vectoriels W_j sont obtenus par projection sur une fonction d'ondelette discrète définie par l'équation (2.36)

$$\psi_{j,n}(x) = 2^{-\frac{j}{2}}\psi(2^{-j}x - n), n \in \mathbb{Z} \quad (2.36)$$

$$q_n^j = \int_{-\infty}^{+\infty} f(x)\psi_{j,n}(x)dx \quad (2.37)$$

Si f_a^j représente l'approximation "grossière" d'une fonction f^0 à la résolution j et f_d^j les détails de f^0 , alors $f^{j-1} = f_a^j + f_d^j$.

Il a été démontré qu'il est possible d'effectuer une décomposition en ondelette à partir d'un banc de filtres [Mallat 09]. Ces filtres sont définis en fonction des séries numériques h et g correspondant aux réponses impulsionnelles de filtres numériques obtenus à partir de $\varphi(x)$:

$$\varphi(x) = \sum_n h[n]\varphi_{-1,n}(x) \quad \psi(x) = \sum_n g[n]\varphi_{-1,n}(x) \quad (2.38)$$

Le filtre $g[n]$ peut être construit à partir de $h[n]$.

Le schéma présenté en figure FIG 2.11(a) représente l'analyse en ondelette d'un signal qui correspond à sa décomposition en ondelettes. En figure FIG 2.11(b), le schéma représente la reconstruction du signal à partir des coefficients issus de l'analyse.

Les coefficients d'échelle a_n^j sont obtenus à l'échelle j par filtrage des coefficients a^{j-1} par le filtre de réponse impulsionnelle \tilde{h} ($\tilde{h}[n]$ est la séquence numérique $h[n]$ retournée) suivi d'une décimation de rapport 2. Les coefficients d'ondelettes d_n^j sont obtenus à l'échelle j par filtrage des coefficients a^{j-1} par le filtre de réponse impulsionnelle \tilde{g} suivi d'une décimation de rapport 2.

On peut trouver un grand nombre de familles d'ondelettes dans la littérature. Chaque famille correspond à une construction particulière des filtres h et g . Les plus utilisées sont les ondelettes B-Spline et les ondelettes à support compact. L'objectif des bases d'ondelettes B-Spline est d'améliorer la localisation fréquentielle. Les fonctions B-Spline sont orthogonalisées afin de former une base orthogonale. Les filtres associés aux ondelettes B-Splines sont des filtres à réponse impulsionnelle infinie (RII). Dans la pratique, il faut tronquer ces filtres à un nombre limité de coefficients. La précision de l'analyse dépend alors du nombre de coefficients conservés. Cela implique que les filtres soit très grands pour avoir une approximation de bonne qualité. Les ondelettes à support compact proposées par INGRID DAUBECHIES dans [Daubechies 88] répondent à ce problème en proposant des filtres à réponse impulsionnelle finie (RIF). La contrepartie de ces filtres est la perte de linéarité en phase. Ces ondelettes sont utilisées dans le format de compression JPEG2000 [Taubman 00].

Les ondelettes B-Splines sont utilisées par BRUNO et PELLERIN [Bruno 01] pour estimer le mouvement global. Précédemment, dans [Bruno 00] les équations permettant d'approximer

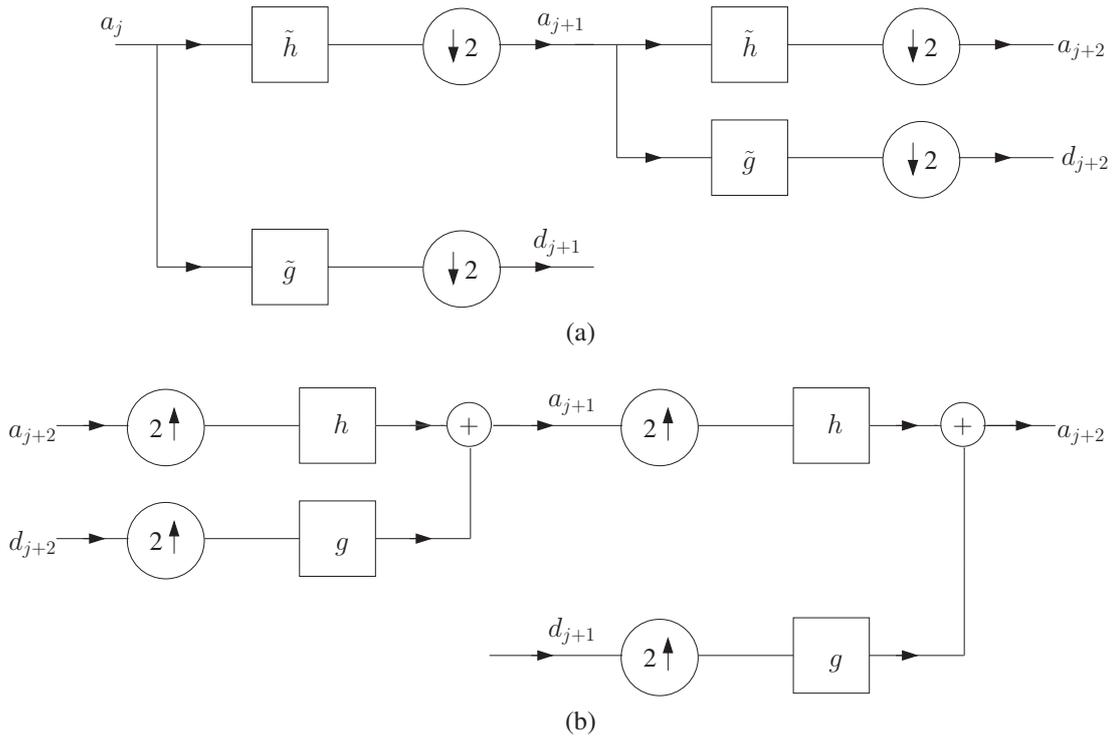


FIGURE 2.11 – Schéma d'un banc de filtre d'analyse et de synthèse ; (a) Schéma d'analyse en ondelettes ; (b) Schéma de synthèse en ondelettes.

chaque composante du mouvement étaient modélisées par des séries de Fourier. Ici, les composantes \mathcal{U} et \mathcal{V} sont modélisées par projection dans une base d'ondelettes B-Spline. Ces équations sont alors injectées dans l'équation du flot optique. Le système obtenu est creux et il est résolu par la méthode du gradient conjugué. Le schéma multi-résolutions consiste à estimer un modèle du mouvement global simple à la plus basse résolution, prenant en compte les plus grands déplacements. Cette estimation sert ensuite d'initialisation du système à la résolution suivante. Cela est répété jusqu'à la résolution la plus fine.

Une méthode très proche de celle-ci a été développée par WU et al. [Wu 00]. Ils utilisent les ondelettes de CAI-WANG [Cai 96] qui sont construites à partir de B-Spline pour représenter le champ de déplacements. Ce modèle multi-résolution du champ de déplacements est alors injecté dans l'équation du flot optique. Le système est résolu en cherchant à minimiser l'erreur quadratique entre l'image à l'instant $(t + 1)$ et l'image à l'instant t ayant subi le déplacement cherché. La minimisation de cette erreur est effectuée par l'algorithme itératif de Levenberg-Marquardt.

2.3.4 Les fonctions polynomiales

Les modèles polynomiaux trouvent un grand nombre d'applications liées à l'étude du mouvement dans les séquences d'images. Un modèle polynomial de degré N peut être écrit comme :

$$\mathcal{C} = (\mathcal{U}, \mathcal{V}) = \sum_{i,j}^{(i+j) \leq N} (a_{i,j}, b_{i,j}) x^i y^j \quad (2.39)$$

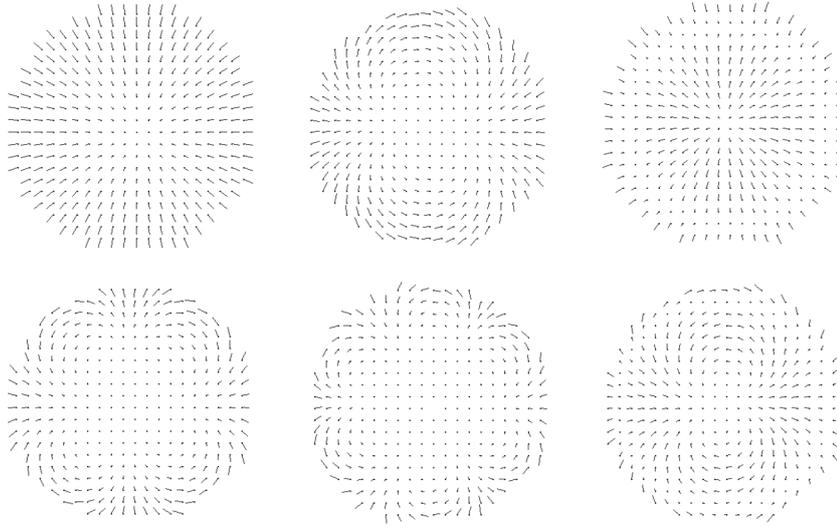


FIGURE 2.12 – Exemples de champs de déplacements extraits d’une base composée des polynômes de Zernicke. Image extraite de [Hoey 00].

Un tel modèle utilisé avec un degré $N = 1$ permet de représenter toutes les transformations affines du plan. Cela peut être très utile en analyse du mouvement puisque les mouvements de caméra tels que les zoom, les translations sont des transformations affines. De même, dans le contexte de l’étude des mouvements fluides, la détection de singularités (qui fait l’objet du chapitre 7) a recours au modèle affine. Ce choix est fortement influencé par le théorème de Grossman-Hartman qui précise que dans le voisinage proche d’une singularité, il est possible d’utiliser un modèle linéaire pour représenter le mouvement.

Par ailleurs, ce modèle a été utilisé dans l’estimation du mouvement global par flot optique [Black 96b], la segmentation du mouvement [Chang 97], [Odobez 98], [Cremers 05] et l’indexation de vidéo [Gelgon 98]. Ce modèle peut être utilisé à des degrés supérieurs à 1. Cependant ces modèles ne sont généralement pas très robustes. Ils sont donc très souvent utilisés en degré 1 ou 2.

Les fonctions polynomiales sont aussi utilisées par HOEY et LITTLE, mais sous un formalisme différent [Hoey 00], [Hoey 02]. Ici les fonctions polynomiales sont obtenues par projection dans une base de polynômes de Zernike. La figure FIG 2.12 extraite de [Hoey 00] présente quelques exemples de champs de déplacements extraits d’une base composée des polynômes de Zernike. Il s’agit de polynômes complexes définis sur le cercle unité. Ces polynômes forment une base orthogonale complète. Ils sont définis pour les coordonnées polaires $\rho = \sqrt{x_1^2 + x_2^2}$ ($\rho \in [0; 1]$) et $\theta = \arctan(\frac{x_2}{x_1})$ ($\theta \in [0; 2\pi[$) par

$$\mathcal{Z}_n^m(\rho, \theta) = \mathcal{R}_n^m(\rho)e^{im\theta} \quad (2.40)$$

où m et n sont deux entiers tels que $n \geq m \geq 0$ et $\mathcal{R}_n^m(\rho)$ est une fonction radiale définie par :

$$\mathcal{R}_n^m(\rho) = \sum_{l=0}^{\frac{n-m}{2}} \frac{(-1)^l (n-l)!}{l! (\frac{1}{2}(n+m) - l)! (\frac{1}{2}(n-m) - l)!} \rho^{n-2l} \quad (2.41)$$

si $n - m$ est pair et par $\mathcal{R}_n^m(\rho) = 0$ si $(n - m)$ est impair.

Il est possible d'exprimer une fonction \mathcal{F} par une combinaison linéaire de ces polynômes :

$$\mathcal{F}(\rho, \theta) = \sum_{m=0}^M \sum_{n=0}^M (A_n^m \cos(m\theta) + B_n^m \sin(m\theta)) \mathcal{R}_n^m(\rho) \quad (2.42)$$

avec M et N permettant de définir le nombre de polynômes utilisés et les coefficients A et B définis par

$$\begin{cases} A_n^m = \frac{\epsilon_m(n+1)}{\pi} \int_{x_1, x_2}^{\Omega} \mathcal{F}(x_1, x_2) \mathcal{R}_n^m(\rho) \cos(m\theta) \\ B_n^m = \frac{\epsilon_m(n+1)}{\pi} \int_{x_1, x_2}^{\Omega} \mathcal{F}(x_1, x_2) \mathcal{R}_n^m(\rho) \sin(m\theta) \end{cases} \quad \text{avec } \epsilon_m = \begin{cases} 1 & \text{si } m = 0 \\ 2 & \text{sinon} \end{cases} \quad (2.43)$$

En projetant chacune des composantes \mathcal{U} et \mathcal{V} d'un champ de déplacements sur cette base de polynôme, on obtient une approximation du champ de déplacement. Cette méthode est utilisée pour modéliser les mouvements du visage afin d'interpréter les différentes émotions d'une personne comme le dégoût, la peur, la joie, la tristesse ou la surprise. Les auteurs estiment qu'une base composée de 11 polynômes de Zernike permet de modéliser les cinq expressions du visage étudiées.

Dans ces travaux de thèse, DRUON a présenté une méthode de modélisation polynomiale par projection sur une base de polynômes bivariés orthogonaux [Druon 09]. Les polynômes ne sont pas définis sur le cercle unité comme c'est le cas dans [Hoey 00], mais bien sur le domaine complet du champ. De plus, la famille de polynômes est laissée libre par le choix d'une fonction de pondération. Les coefficients polynomiaux sont obtenus par projection sur la base de polynômes. Les travaux que nous présentons dans ce manuscrit sont fondés sur cette approche. Nous reviendrons plus en détails sur celle-ci au chapitre 3.

2.3.5 Conclusion

Ce chapitre nous a permis d'introduire le vocabulaire, le contexte et certains enjeux de nos travaux. Nous avons présenté les grandes lignes des méthodes d'estimation du mouvement. Bien que nos travaux de modélisation soient indépendants de la méthode d'estimation du mouvement, lorsque dans la suite du document, nous présenterons un champ de déplacements pour la première fois, nous donnerons des informations sur la méthode d'estimation utilisée.

Ensuite, nous avons présenté différentes applications faisant intervenir des méthodes d'analyse du mouvement. Dans la suite, nous présenterons dans la partie III deux applications. L'une se place dans le contexte de l'analyse de mouvement fluide et l'autre dans le contexte de l'analyse de mouvement humain.

Finalement, nous avons présenté quelques méthodes de modélisations. Notre objectif est de proposer un modèle permettant d'analyser le mouvement, qui dans notre cas, est représenté par un champ de déplacements. Le modèle polynomial proposé par DRUON [Druon 09] est particulièrement bien adapté à la modélisation du mouvement. Ce modèle offre l'avantage d'être indépendant des données à modéliser, ce qui permet d'avoir un même modèle quelque soit le type de mouvements étudiés. Les modèles fondés sur la SVD tels que la POD ne permettent pas cela. En effet, les fonctions polynomiales offrent la possibilité de représenter des mouvements simples

avec peu de coefficients. Les mouvements de caméra, par exemple, peuvent être représentés avec 6 coefficients. De plus, les mouvements plus complexes sont aussi modélisables à condition d'augmenter le degré polynomial. De plus, les polynômes utilisés sont orthogonaux. Cette orthogonalité est importante puisqu'elle nous permet de s'affranchir de l'estimation au sens des moindres carrés et d'obtenir les coefficients par projection sur la base.

Cependant, si dans le cas de mouvements simples, l'analyse des coefficients est aisée, elle devient délicate lorsque la complexité du champ augmente. De plus, nous avons vu que selon la méthode utilisée pour estimer le mouvement, les champs de déplacements peuvent être denses, épars ou sous-échantillonnés. La méthode proposée par DRUON, s'attache à l'analyse de champs de déplacements denses dans leur globalité et nécessite parfois d'utiliser des bases de degré élevé pour analyser les mouvements complexes. Nous proposons donc dans la partie suivante des solutions pour agir sur ces deux points.

Deuxième partie

Approximation des champs de déplacements dans des bases polynomiales

INTRODUCTION

L'objet de cette partie concerne l'étude et le développement d'un outil générique de modélisation du mouvement, pouvant s'adapter aux différents domaines de la science expérimentale ayant recours à l'analyse du mouvement. Nous cherchons à étendre les travaux de MARTIN DRUON [Druon 09] concernant l'utilisation des bases de polynômes orthogonaux. Au cours de ses travaux, MARTIN DRUON a comparé cette méthode de modélisation globale du mouvement à une autre méthode couramment utilisée dans ce domaine ayant recours à une décomposition en valeurs singulières de la séquence. Encouragés par la qualité des résultats, nous avons souhaité poursuivre les recherches concernant l'utilisation des bases de polynômes pour la modélisation du mouvement suivant trois axes principaux. Le premier concerne l'étude des modélisations hiérarchiques permettant d'aboutir à différents niveaux d'analyses d'un même mouvement, le second, la modélisation des champs épars et le troisième, l'utilisation de cette méthode de modélisation à d'autres mouvements que les mouvements fluides. Cette partie est organisée en quatre chapitres.

Dans le chapitre 3, nous fixons le cadre mathématique de l'approximation de fonctions et nous présentons une méthode de génération des bases de polynômes orthogonaux. Ensuite, nous présentons la modélisation globale d'un champ dense avec les bases de polynômes. Nous rappelons aussi quelques résultats de la thèse de MARTIN DRUON [Druon 09].

Après avoir mis en évidence quelques limites de la modélisation globale, nous traitons au chapitre 4, la modélisation hiérarchique d'un champ de déplacements. Nous utilisons les bases de polynômes avec un certain nombre d'algorithmes classiques de subdivision tels que le quadtree ou le kd-tree. Ces différentes méthodes de subdivisions permettent d'obtenir une modélisation hiérarchique de qualité, sans utiliser des bases de degré élevé.

Au chapitre 5, nous abordons la modélisation des champs épars. Bien que les méthodes hiérarchiques permettent d'améliorer les résultats d'approximation d'un champ épars par rapport à une modélisation globale, les résultats restent insuffisants. Nous proposons alors de revenir sur la génération des bases de polynômes orthogonaux en générant une base en fonction du support spatial du champ. On peut alors à nouveau utiliser tous les algorithmes hiérarchiques étudiés aux chapitres précédents.

Finalement, après avoir étudié différentes méthodes d'approximation des champs de déplacements, nous proposons, au chapitre 6, de générer des bases de polynômes d'approximation à erreur nulle. À partir de ces bases, nous proposons un algorithme de décomposition multi-résolutions des champs de déplacements.

CHAPITRE 3

APPROXIMATION GLOBALE DES CHAMPS DENSES

3.1 Introduction et définitions

Nous cherchons à obtenir une représentation analytique de chacune des composantes d'un champ de déplacements. Pour cela, nous proposons d'approximer les fonctions \mathcal{U} et \mathcal{V} comme étant des combinaisons linéaires de fonctions polynomiales.

Dans un premier temps, nous présentons le cadre théorique général de l'approximation de fonctions dans un espace préhilbertien. Puis, dans un deuxième temps, nous appliquons ce cadre au cas particulier de l'approximation de fonctions par combinaisons linéaires de fonctions polynomiales à deux variables. Finalement, nous donnons quelques exemples d'approximation de champs de déplacements dans le cas particulier des polynômes de Legendre.

Pour les développements qui suivent nous supposons que \mathcal{U} et \mathcal{V} appartiennent à l'espace vectoriel préhilbertien de fonctions \mathcal{E} . Étant donné que ces fonctions possèdent les mêmes caractéristiques et afin de simplifier les développements mathématiques, nous proposons d'étudier plus particulièrement l'approximation de la fonction \mathcal{U} . Nous cherchons à trouver la fonction $\tilde{\mathcal{U}}$ qui approche le mieux \mathcal{U} . Par conséquent, nous devons dans un premier temps préciser comment mesurer la distance entre deux fonctions de l'espace \mathcal{E} .

Dans le cadre des espaces préhilbertiens, le produit scalaire de deux fonctions f_1 et f_2 noté $\langle f_1 | f_2 \rangle$ permet de définir une norme $\|\cdot\|_{\mathcal{E}}$ sur cet espace. Le produit scalaire est défini par

$$\langle f_1 | f_2 \rangle = \int_{\Omega} f_1(x_1, x_2) f_2(x_1, x_2) w(x_1, x_2) d\Omega \quad (3.1)$$

où $w(x_1, x_2)$ est une fonction de pondération et Ω le domaine de définition des variables x_1 et x_2 .

La norme $\|\cdot\|_{\mathcal{E}}$ est alors définie par la relation $\|f\|_{\mathcal{E}} = \sqrt{\langle f | f \rangle}$.

Ainsi l'écart entre deux fonctions f_1 et f_2 de \mathcal{E} est défini par la distance :

$$d(f_1, f_2) = \|f_1 - f_2\|_{\mathcal{E}} \quad (3.2)$$

Soit F_n un sous espace vectoriel complet de \mathcal{E} et de dimension $n+1$. Nous cherchons donc à construire $\tilde{\mathcal{U}}$ appartenant à F_n de dimension n , engendré par une base de fonctions $\{\varphi_0, \dots, \varphi_n\}$ qui approxime au mieux \mathcal{U} . Ces fonctions pourront être suivant les cas, soit des polynômes, soit des fonctions trigonométriques, soit des fonctions polynomiales par morceaux (splines), soit des ondelettes... Dans notre cas, nous utilisons une base orthogonale de fonctions polynomiales.

Soit $\{\varphi_0, \varphi_1, \dots, \varphi_n\}$ notre base dans F_n , alors nous pouvons exprimer $\tilde{\mathcal{U}}$ par une combinaison linéaire des éléments de cette base :

$$\tilde{\mathcal{U}} = \sum_{i=0}^n \tilde{\alpha}_i \varphi_i \quad (3.3)$$

Il est possible de démontrer que si \mathcal{U} est une fonction continue sur Ω et φ_i un ensemble de fonctions linéairement indépendantes, la fonction $\tilde{\mathcal{U}}$ telle que définie en équation (3.3) est une fonction approchée de \mathcal{U} unique au sens de la norme L_2 .

À partir des fonctions approximantes et de la fonction de distance, nous cherchons la meilleure approximation de \mathcal{U} . Cela revient à déterminer les paramètres $\tilde{\alpha}_i$ tels que $\|\mathcal{U} - \tilde{\mathcal{U}}\|_{\mathcal{E}}$ soit minimal. Il faut donc résoudre le système linéaire d'équations suivant [Rice 69] :

$$\forall j \in \{0, \dots, n\} \quad \sum_{i=0}^n \langle \varphi_i | \varphi_j \rangle \tilde{\alpha}_i = \langle \mathcal{U} | \varphi_j \rangle \quad (3.4)$$

qui peut s'écrire sous la forme matricielle :

$$\Phi \tilde{A} = B \quad (3.5)$$

où

$$\Phi = \begin{pmatrix} \langle \varphi_0 | \varphi_0 \rangle & \cdots & \langle \varphi_0 | \varphi_n \rangle \\ \vdots & & \vdots \\ \langle \varphi_n | \varphi_0 \rangle & \cdots & \langle \varphi_n | \varphi_n \rangle \end{pmatrix}, \quad \tilde{A} = \begin{pmatrix} \tilde{\alpha}_0 \\ \vdots \\ \tilde{\alpha}_n \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} \langle \mathcal{U} | \varphi_0 \rangle \\ \vdots \\ \langle \mathcal{U} | \varphi_n \rangle \end{pmatrix}$$

Φ est inversible car $\tilde{\mathcal{U}}$ est unique donc \tilde{A} est unique et vaut :

$$\tilde{A} = \Phi^{-1} B \quad (3.6)$$

La résolution du système est alors un problème d'inversion de matrice. Cependant ce problème peut être coûteux en temps de calcul et numériquement instable. En effet, l'inversion d'une matrice de dimension $n \times n$ par la méthode d'élimination de Gauss-Jordan nécessite $\frac{2}{3}n^3$ opérations.

L'erreur d'approximation de \mathcal{U} par $\tilde{\mathcal{U}} = \sum_{i=0}^n \tilde{\alpha}_i \varphi_i$ est égale à :

$$\|\mathcal{U} - \tilde{\mathcal{U}}\|_{\mathcal{E}} = \sqrt{\left(\|\mathcal{U}\|_{\mathcal{E}}^2 - \sum_{i=0}^n \tilde{\alpha}_i \langle \mathcal{U} | \varphi_i \rangle \right)} \quad (3.7)$$

Cependant sous certaines conditions, il est possible de résoudre ce problème sans avoir à inverser cette matrice. En effet, si la base $\{\varphi_0, \dots, \varphi_n\}$ est orthogonale, la matrice Φ est diagonale et le système (3.5) est par conséquent facilement inversible. Les coefficients $\tilde{\alpha}_i$ sont alors donnés par :

$$\tilde{\alpha}_i = \frac{\langle \mathcal{U} | \varphi_i \rangle}{\langle \varphi_i | \varphi_i \rangle}, \forall i \in \{0, \dots, n\} \quad (3.8)$$

et correspondent aux projections orthogonales de \mathcal{U} sur les éléments de la base. Ceci met en exergue l'intérêt de travailler avec une base de fonctions orthogonales. Dans ce cas, l'erreur d'approximation de \mathcal{U} par $\tilde{\mathcal{U}}$ est égale à :

$$\|\mathcal{U} - \tilde{\mathcal{U}}\|_{\mathcal{E}} = \sqrt{\left(\|\mathcal{U}\|_{\mathcal{E}}^2 - \sum_{i=0}^n \tilde{\alpha}_i^2 \langle \varphi_i | \varphi_i \rangle \right)} \quad (3.9)$$

Nous présentons dans la section suivante une technique permettant de générer une base de polynômes orthogonaux.

3.2 Construction de la base polynomiale

Nous définissons d'abord la famille de fonctions polynomiales à deux variables réelles suivante :

$$P_{K,L}(x_1, x_2) = \sum_{k=0}^K \sum_{l=0}^L a_{k,l} (x_1)^k (x_2)^l \quad (3.10)$$

où $K \in \mathbb{N}^+$ et $L \in \mathbb{N}^+$ sont respectivement les degrés maximaux des variables x_1 et x_2 et les $\{a_{k,l}\}_{k \in \{0..K\}, l \in \{0..L\}} \in \mathbb{R}^{(K+1) \times (L+1)}$ sont les coefficients du polynôme. Le degré global du polynôme est alors $K + L$.

Soit $\mathcal{B} = \{P_{k,l}\}_{k \in \{0..K\}, l \in \{0..L\}}$ une base orthogonale de polynômes de F_n classés par ordre lexicographique croissant des exposants des degrés de x_1 et x_2 . Nous appelons degré D de cette base le degré le plus élevé des polynômes qui la constituent, c'est-à-dire, l'ensemble des polynômes $P_{k,l} / k+l \leq D$. Ainsi, une base de degré D est composée de n polynômes avec $n = (D+1)(D+2)/2$:

$$\mathcal{B} = \{P_{0,0}, P_{0,1}, \dots, P_{0,L}, P_{1,0}, \dots, P_{1,L-1}, \dots, P_{K-1,0}, P_{K-1,1}, P_{K,0}\} \quad (3.11)$$

Nous choisissons ici de caractériser l'écart entre deux fonctions de \mathcal{E} par la norme quadratique classiquement utilisée dans le cadre de l'étude des fonctions polynomiales. Pour cela, nous faisons intervenir le produit scalaire de deux fonctions f_1 et f_2 de \mathcal{E} défini par l'équation (3.1). Ce dernier fait intervenir $w(x_1, x_2)$ une fonction de poids positive permettant de pondérer la qualité de l'approximation aux points de coordonnées (x_1, x_2) . Notons aussi que celle-ci conditionne la famille de polynômes utilisée. Plusieurs fonctions de pondération sont présentées dans le tableau TAB 3.1. Comme nous travaillons dans le domaine discret, nous nous intéressons à la manière de discrétiser le domaine. Le domaine discret est ramené à une grille d'échantillonnage du domaine continu, par exemple $[-1; 1]^2$. Nous présentons deux méthodes de collocation, la première, la plus classique, est la collocation uniforme en équation (3.12) et la seconde est la collocation de Gauss-Tchébichev en équation (3.13). La collocation de Gauss-Tchébichev cor-

Famille	Ω	$w(x_1, x_2)$
Legendre	$[-1; 1]^2$	1
Tchebychev 1	$[-1; 1]^2$	$\frac{1}{\sqrt{(1-x_1)^2(1-x_2)^2}}$
Tchebychev 2	$[-1; 1]^2$	$\sqrt{(1-x_1)^2(1-x_2)^2}$
Laguerre	$[0; \infty]^2$	$e^{-(x_1+x_2)}$
Hermite	$[-\infty; \infty]^2$	$e^{-\frac{x_1^2+x_2^2}{2}}$

TABLEAU 3.1 – Quelques fonctions de pondération w permettant d’obtenir différentes familles de polynômes

respond aux zéros des polynômes de Tchebychev et est optimale au sens de la quadrature de Gauss.

$$x_i(k) = -1 + k \frac{2}{N-1} \quad (3.12)$$

$$x_i(k) = -\cos\left(\frac{2k+1}{2N}\pi\right) \quad (3.13)$$

De même, le produit scalaire doit être calculé par une approximation numérique d’intégrale. Plusieurs solutions existent : trapèzes, NEWTON, intégration de GAUSS, transposition élémentaire. Nous avons choisi la transposition élémentaire. Cela nous permet d’obtenir des résultats satisfaisants tout en conservant une grande efficacité au niveau des coûts calculatoires.

En se basant sur un produit scalaire, on peut générer une base en appliquant la procédure de récurrence à trois termes (aussi appelée procédure de STIELJES dans le cas discret lorsque l’intégrale est calculée par transposition élémentaire) :

$$\left\{ \begin{array}{l} P_{-1,j}(x_1, x_2) = 0 \\ P_{i,-1}(x_1, x_2) = 0 \\ P_{0,0}(x_1, x_2) = 1 \\ P_{i+1,j}(x_1, x_2) = (x_1 - \lambda_{i+1,j})P_{i,j}(x) - \mu_{i+1,1}P_{i-1,j}(x) \\ P_{i,j+1}(x_1, x_2) = (x_2 - \lambda_{i,j+1})P_{i,j}(x) - \mu_{i,j+1}P_{i,j-1}(x) \end{array} \right. \quad (3.14)$$

avec les coefficients λ et μ donnés par

$$\begin{aligned} \lambda_{i+1,j} &= \frac{\langle x_1 P_{i,j} | P_{i,j} \rangle}{\langle P_{i,j} | P_{i,j} \rangle} & \lambda_{i,j+1} &= \frac{\langle x_2 P_{i,j} | P_{i,j} \rangle}{\langle P_{i,j} | P_{i,j} \rangle} \\ \mu_{i+1,j} &= \frac{\langle P_{i,j} | P_{i,j} \rangle}{\langle P_{i-1,j} | P_{i-1,j} \rangle} & \mu_{i,j+1} &= \frac{\langle P_{i,j} | P_{i,j} \rangle}{\langle P_{i,j-1} | P_{i,j-1} \rangle} \end{aligned} \quad (3.15)$$

Notons qu’il est aussi possible de générer une base de polynômes orthogonaux par orthogonalisation d’une base de polynômes quelconque avec la procédure de GRAMM-SCHMIDT.

Des informations concernant la complexité algorithmique et l’occupation mémoire des bases de polynômes orthogonaux sont présentées ANNEXE B. La section suivante montre comment il est possible d’utiliser une base de polynômes pour réaliser l’approximation d’un champ dense par projection sur la base.

3.3 Approximation globale des champs denses

L'approximation $\tilde{\mathcal{C}} = (\tilde{\mathcal{U}}, \tilde{\mathcal{V}})$ d'un champ de vecteurs bidimensionnels $\mathcal{C} = (\mathcal{U}, \mathcal{V})$ s'exprime à partir de la projection de chacune des composantes sur les polynômes de la base. Nous présentons en équation 3.16 comment une composante, ici \mathcal{U} , est projetée sur un polynôme de la base.

$$\tilde{u}_{i,j} = \frac{\langle \mathcal{U} | P_{i,j} \rangle}{\langle P_{i,j} | P_{i,j} \rangle} \quad (3.16)$$

À partir de là, en projetant chaque composante sur l'ensemble des polynômes de la base, on obtient les approximations $\tilde{\mathcal{U}}$ et $\tilde{\mathcal{V}}$ des composantes \mathcal{U} et \mathcal{V} d'un champ de déplacements. Il est alors possible de reconstruire le champ à partir de cette approximation.

$$\begin{cases} \tilde{\mathcal{U}} = \sum_{i=0}^D \sum_{j=0}^{D-i} \tilde{u}_{i,j} P_{i,j} & \text{avec } \tilde{u}_{i,j} = \frac{\langle \mathcal{U} | P_{i,j} \rangle}{\langle P_{i,j} | P_{i,j} \rangle} \\ \tilde{\mathcal{V}} = \sum_{i=0}^D \sum_{j=0}^{D-i} \tilde{v}_{i,j} P_{i,j} & \text{avec } \tilde{v}_{i,j} = \frac{\langle \mathcal{V} | P_{i,j} \rangle}{\langle P_{i,j} | P_{i,j} \rangle} \end{cases} \quad (3.17)$$

La reconstruction du champ se fait simplement en réévaluant, à partir des coefficients $\tilde{u}_{i,j}$ et $\tilde{v}_{i,j}$ et de la base \mathcal{B} , l'équation (3.17) pour tous les sites (x_1, x_2) du domaine Ω du champ. Ainsi, un champ de résolutions $N \times M$ est entièrement caractérisé par $(D+1)(D+2)$ coefficients et leurs polynômes associés. Ceci met en évidence la capacité de cette approche à compresser les données initiales.

3.3.1 Utilisation des bases de degré 1

Les coefficients issus de la projection dans une base de degré 1 permettent de représenter l'ensemble des mouvements affines sans erreur. On peut ainsi modéliser l'ensemble des mouvements élémentaires présentés en figure FIG 3.1 par les combinaisons de polynômes suivantes.

- Mouvement de translation horizontale (cf. FIG 3.1a)

$$\begin{cases} P_{\mathcal{U}} = tr_{x_1} P_{0,0} + 0 P_{0,1} + 0 P_{1,0} \\ P_{\mathcal{V}} = 0 P_{0,0} + 0 P_{0,1} + 0 P_{1,0} \end{cases} \quad (3.18)$$

- Mouvement de translation verticale (cf. FIG 3.1b)

$$\begin{cases} P_{\mathcal{U}} = 0 P_{0,0} + 0 P_{0,1} + 0 P_{1,0} \\ P_{\mathcal{V}} = tr_{x_2} P_{0,0} + 0 P_{0,1} + 0 P_{1,0} \end{cases} \quad (3.19)$$

- Mouvement divergent (cf. FIG 3.1c)

$$\begin{cases} P_{\mathcal{U}} = 0 P_{0,0} + 0 P_{0,1} + div_{x_1} P_{1,0} \\ P_{\mathcal{V}} = 0 P_{0,0} + div_{x_2} P_{0,1} + 0 P_{1,0} \end{cases} \quad (3.20)$$

- Mouvement de rotation affine (cf. FIG 3.1d)

$$\begin{cases} P_{\mathcal{U}} = 0 P_{0,0} - rot_{x_1} P_{0,1} + 0 P_{1,0} \\ P_{\mathcal{V}} = 0 P_{0,0} + 0 P_{0,1} + rot_{x_2} P_{1,0} \end{cases} \quad (3.21)$$

– Mouvement hyperbolique de type 1 (cf. FIG 3.1e)

$$\begin{cases} P_U = 0 P_{0,0} + 0 P_{0,1} + hypx_1 P_{1,0} \\ P_V = 0 P_{0,0} - hypx_2 P_{0,1} + 0 P_{1,0} \end{cases} \quad (3.22)$$

– Mouvement hyperbolique de type 2 (cf. FIG 3.1f)

$$\begin{cases} P_U = 0 P_{0,0} + hyp2_{x_1} P_{0,1} + 0 P_{1,0} \\ P_V = 0 P_{0,0} + 0 P_{0,1} + hyp2_{x_2} P_{1,0} \end{cases} \quad (3.23)$$

où $tr_{x_1}, div_{x_1}, rot_{x_1}, hypx_1, hyp2_{x_1}$ (resp. $tr_{x_2}, div_{x_2}, rot_{x_2}, hypx_2, hyp2_{x_2}$) sont les paramètres horizontaux (resp. verticaux) du mouvement affine.

La modélisation affine (polynomiale de degré 1) est utilisée dans de nombreux domaines. Les mouvements de caméra sont souvent représentés ainsi. Dans sa thèse, DRUON [Druon 09], a étudié l'analyse des mouvements de caméra à partir des coefficients polynomiaux obtenus par projection sur une base de polynômes orthogonaux bi-variables. Il est donc très intéressant que notre modèle permette d'avoir une bonne représentation de ces mouvements.

Pour analyser des écoulements fluides, il est usuel de chercher à détecter des structures singulières du champ de déplacements. Ces structures sont détectables entre autres par les méthodes de portrait de phase [Rao 92a]. Les méthodes de portraits de phase utilisent une modélisation affine du champ de déplacements. De plus, le théorème de GROBMAN-HARTMAN [Palis 82a] établit que, dans le voisinage de ces singularités, une approximation par un modèle linéaire correspond à des solutions asymptotiquement exactes des équations de NAVIER-STOKES.

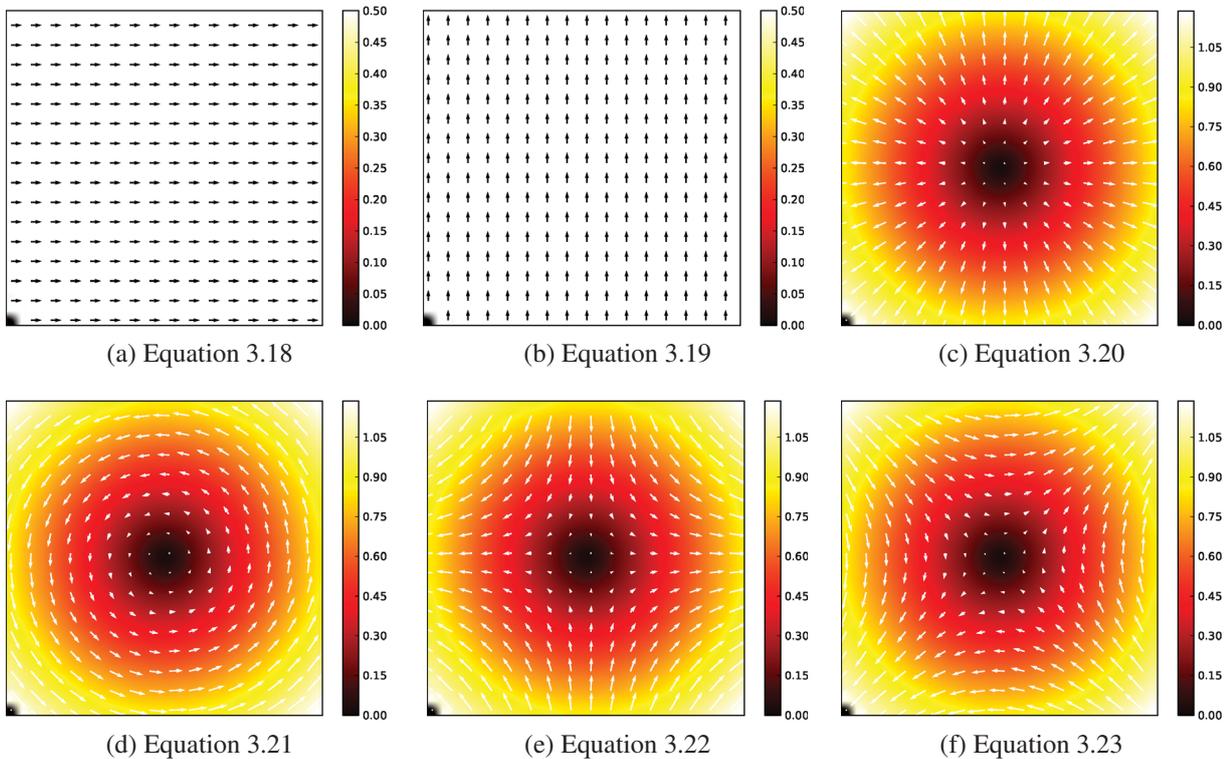


FIGURE 3.1 – Mouvements affines élémentaires

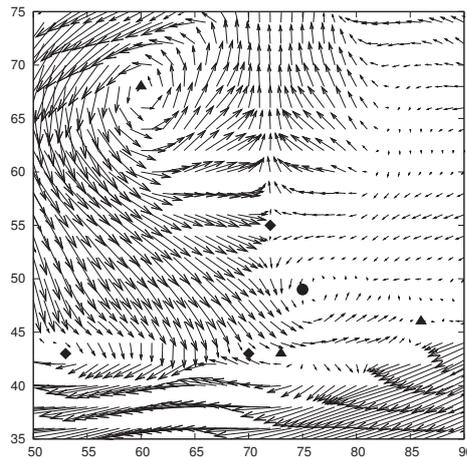


FIGURE 3.2 – Exemple de détection de singularités dans un champ de déplacements ; les ▲ représentent les tourbillons ; les ◆ représentent les points de selles ; les ● représentent les noeuds impropres et étoilés ; Ces différentes structures ainsi que les méthodes de détection sont détaillées au chapitre 7 ; Le champ de déplacements est le champ 100 de la séquence cylindre-Carré présenté dans ANNEXE A

La détection d'une singularité par portrait de phase consiste à détecter une singularité à partir d'une représentation affine du mouvement de la forme :

$$\begin{aligned}\tilde{\mathcal{U}}(x) &= a_{11}x_1 + a_{12}x_2 + b_{11} \\ \tilde{\mathcal{V}}(x) &= a_{21}x_1 + a_{22}x_2 + b_{21}\end{aligned}\quad (3.24)$$

qui peut être écrite sous forme matricielle par

$$\mathcal{C} = Ax + b \quad (3.25)$$

La détection de la position d'une singularité est obtenue par :

$$\tilde{x} = -A^{-1}b \quad (3.26)$$

Il est alors possible de classifier ces singularités en fonction des valeurs propres de la matrice A . La figure FIG 3.2 présente une illustration de singularités détectées dans un champ de déplacement. Nous reviendrons plus en détails sur ce genre d'application au chapitre 7.

Les bases de polynômes de degré 1 permettent une approximation sans erreur de l'ensemble des mouvements affines. En fait, notre modèle permet d'obtenir une approximation à erreur nulle de tous les mouvements polynomiaux, à condition d'utiliser une base de degré appropriée. De plus, il est possible de projeter n'importe quel champ non polynomial sur une base polynomiale. Cependant la qualité de l'approximation dépendra de la complexité du mouvement et du degré de la base.

3.3.2 Utilisation de bases de degré supérieur à 1

Ainsi, nous présentons quelques exemples de champs afin de mettre en évidence le liens entre la complexité du mouvement et le degré de la base.

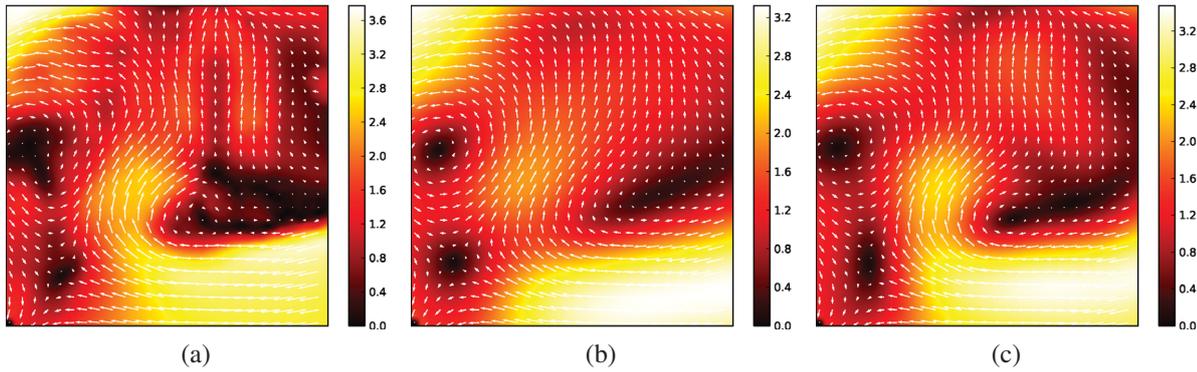


FIGURE 3.3 – Exemple de projections d’un champ complexe ; (a) Champ original (cf. ANNEXE A) ; (b) Champ projeté sur une base de degré 16 ; (c) Champ projeté sur une base de degré 24.

La figure FIG 3.3 montre un extrait d’un champ de déplacements d’un mouvement fluide obtenu par PIV. Ce champ est d’abord approximé par projection dans une base de Legendre de degré 16 puis 24. Cet exemple met en évidence la nécessité d’utiliser une base de degré élevé pour représenter un champ de déplacement complexe.

Nous avons présenté en section 2.2 un exemple d’une méthode usuelle d’analyse des écoulements en mécanique expérimentale des fluides. Il s’agit de l’analyse par portrait de phase des coefficients temporels d’une modélisation par décomposition orthogonale aux valeurs propres (POD). Dans sa thèse, DRUON [Druon 09] réalise une étude similaire à partir des coefficients polynomiaux. Nous en présentons deux résultats en figure FIG 3.4. On peut constater qu’il est possible d’interpréter l’évolution temporelle des coefficients polynomiaux de manière similaire aux coefficients POD. Sur les figures FIG 3.4a et FIG 3.4c, on peut constater qu’un seul anneau apparaît. Cela est significatif d’un écoulement périodique. Sur les figures FIG 3.4b et FIG 3.4d, on peut voir plusieurs anneaux très proches les uns des autres, on peut en déduire que certains phénomènes vont se reproduire mais pas de manière parfaitement périodique. Cela est significatif d’un écoulement de type cycle limite. La modélisation globale par base de polynômes peut donc se substituer à l’analyse POD dans ce contexte. DRUON propose une étude plus poussée de cette modélisation en la comparant à une modélisation par POD sur trois séquences d’écoulement fluide [Druon 09].

Lorsque l’on utilise une base de degré supérieur à 1, le choix de la qualité d’approximation (donc du degré de la base) va dépendre de l’application. Ce choix sera contraint par le temps de calcul nécessaire à la génération de la base, l’espace mémoire disponible pour stocker les coefficients polynomiaux ou encore la rapidité des traitements (Projection, reconstruction, analyse) des coefficients polynomiaux.

3.3.3 Gain de compression des bases de polynômes

Nous étudions maintenant la faculté de notre modèle à regrouper une grande quantité d’informations en peu de coefficients par la mesure du gain de compression exprimé par :

$$\tau = 1 - \frac{\text{Nombre de coefficients conservés}}{\text{Nombre total de pixels dans l'image}} \quad (3.27)$$

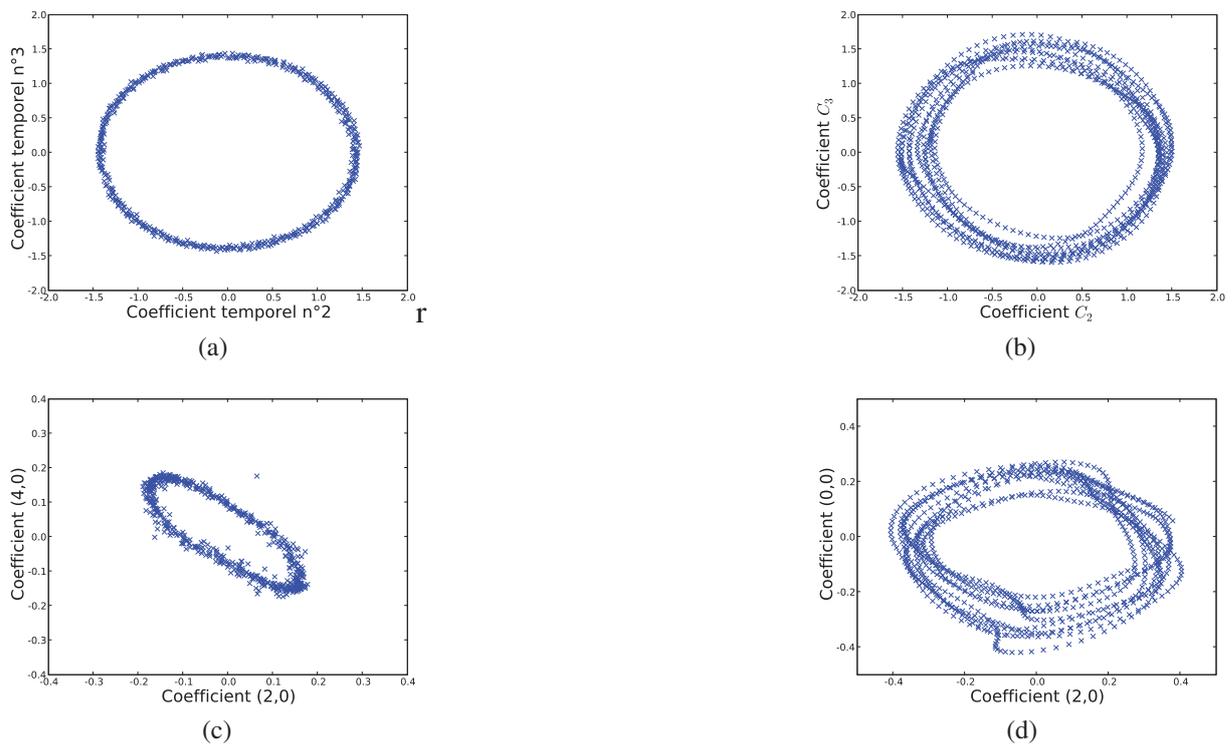
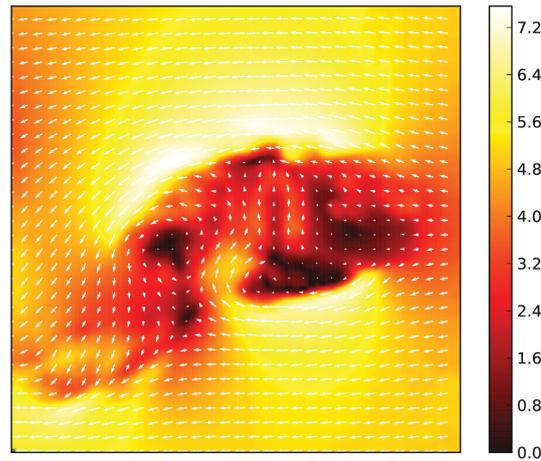


FIGURE 3.4 – Analyse d’écoulement à partir de coefficients POD ; (a) portrait de phase entre les deuxième et troisième coefficients temporels POD d’un écoulement périodique ; (b) portrait de phase entre les deuxième et troisième coefficients temporels POD d’un écoulement de type cycle limite ; (c) portrait de phase entre le coefficient polynomial $\tilde{v}_{2,0}$ et le coefficient polynomial $\tilde{v}_{4,0}$ d’un écoulement périodique ; (d) portrait de phase entre le coefficient polynomial $\tilde{v}_{2,0}$ et le coefficient polynomial $\tilde{v}_{0,0}$ d’un écoulement de type cycle limite ; Figures extraites de [Druon 09]

Afin d’estimer le gain de compression de notre modèle, nous le comparons à une modélisation par décomposition en valeurs singulières (SVD). Il est à noter que les vecteurs propres de la SVD doivent être conservés pour la reconstruction tandis que les bases polynomiales peuvent être générées indépendamment des données. La SVD est effectuée sur chaque composante du champ. Pour cet exemple, nous utilisons le champ 131 de la séquence cylindre-Carré (cf. ANNEXE A) présenté sur la figure FIG 3.5. Nous avons calculé la SVD de ce champ. Nous avons reconstruit les 8 premières approximations issues de la SVD. Dans chaque cas, nous avons cherché le degré de la base de polynômes permettant d’obtenir une erreur d’approximation équivalente.

L’erreur de modélisation est calculée par l’erreur quadratique moyenne (cf. équation (2.6)) entre les composantes \mathcal{U} et \mathcal{V} du champ original et leur approximation respective $\tilde{\mathcal{U}}$ et $\tilde{\mathcal{V}}$. Cela nous permet d’obtenir une estimation du gain de compression de chacun des modèles pour une erreur d’approximation équivalente.

Les résultats sont présentés dans le tableau TAB 3.2. On peut constater que pour des approximations correspondantes aux premiers vecteurs propres de la SVD, la projection dans une base de polynômes a un gain de compression plus important. De plus, une approximation de degré 24 produit une très bonne approximation du champ, puisque l’ensemble des structures présentes dans le champ sont représentées.

FIGURE 3.5 – Champ 131 de la séquence Cylindre-Carré (cf. ANNEXE A)

Méthode	EQM	nombre de coefficients	Gain de compression
SVD (1 vecteur propre)	1.357	249	0.984
MBP (degre 4)	1.272	15	0.999
SVD (2 vecteurs propres)	0.885	497	0.968
MBP (degre 7)	0.852	36	0.998
SVD (3 vecteurs propres)	0.439	745	0.952
MBP (degre 15)	0.426	136	0.991
SVD (4 vecteurs propres)	0.329	993	0.935
MBP (degre 19)	0.329	210	0.986
SVD (5 vecteurs propres)	0.264	1241	0.919
MBP (degre 24)	0.265	325	0.979
SVD (6 vecteurs propres)	0.218	1489	0.903
MBP (degre 27)	0.220	406	0.974
SVD (7 vecteurs propres)	0.186	1737	0.887
MBP (degre 29)	0.192	465	0.970
SVD (8 vecteurs propres)	0.144	1985	0.871
MBP (degre 35)	0.148	666	0.957

TABLEAU 3.2 – Illustration du gain de compression des bases de polynômes orthogonaux bivariés comparées à des approximations par SVD pour plusieurs mesures d'erreurs quadratiques globales équivalentes à partir du champ Mouvement fluide.

Nous présentons une étude similaire à partir d'un autre champ de déplacements. Ici le mouvement est un mouvement humain. Le sujet est filmé pendant un exercice de marche sur une plateforme de mesure de pression. Les images ont une résolution de 512×512 pixels. Le champ de vecteurs est calculé par une méthode de flot optique avec l'algorithme de HORN et SHUNK. La figure FIG 3.6 présente ce champ ainsi que les deux images à partir desquelles il est calculé. Le tableau TAB 3.3 présente les résultats du gain de compression obtenus sur ce champ et la figure FIG 3.7 présente les champs correspondant aux différentes approximations présentées dans ce tableau.

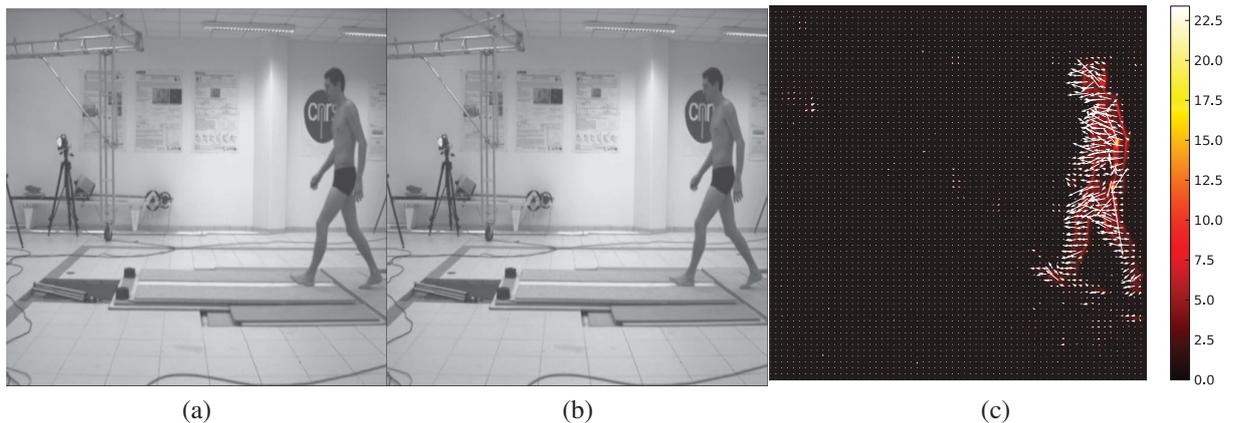


FIGURE 3.6 – Exemple de mouvement humain ; (a) et (b) deux images successives d'un homme qui marche sur une plateforme de mesure de pression ; (c) champ de vecteur représentant le mouvement entre ces deux images (champ Marche).

Méthode	EQM	nombre de coefficients	Gain de compression
SVD (1 vecteur propre)	0.219108	1025	0.933
MBP (degré 24)	0.218003	325	0.979
SVD (2 vecteurs propres)	0.205627	2049	0.867
MBP (degré 30)	0.200956	496	0.968
SVD (3 vecteurs propres)	0.181046	3073	0.800
MBP (degré 35)	0.186685	666	0.957

TABLEAU 3.3 – Illustration du gain de compression des bases de polynômes orthogonaux bivariés comparées à des approximations par SVD pour plusieurs mesures d'erreurs quadratiques globales équivalentes sur le champ Marche.

On peut remarquer que la silhouette humaine est clairement reconnaissable à partir d'une modélisation de degré 24 tandis que par SVD, elle commence seulement à être identifiable à partir de la modélisation qui conserve les trois premiers vecteurs propres. Sur ces deux exemples on peut en conclure que notre méthode de modélisation globale permet d'obtenir une bonne approximation d'un champ de déplacements. De plus, le gain de compression est meilleur que celui d'une modélisation par SVD de qualité globale équivalente.

3.4 Conclusion sur la modélisation globale

Dans ce chapitre, nous avons approximé des champs de déplacements à partir de projections sur une base de polynômes. Dans un premier temps, nous avons utilisé des bases de degré 1. À partir de ces bases, il est possible d'obtenir une modélisation de tous les mouvements affines. Les mouvements affines sont très intéressants car ils permettent une analyse très simple. Il est aisé de donner un sens physique aux coefficients polynomiaux dans une base de degré 1.

Cependant, pour modéliser des mouvements plus complexes, il est essentiel d'utiliser des bases de degrés plus élevés. Nous avons présenté des résultats d'approximation avec des bases de degrés 16 et 24. Il est alors intéressant de constater que notre outil de modélisation permet

une approximation des champs de déplacements avec un gain de compression supérieur à la SVD à qualité équivalente (approximation à partir des premiers vecteurs propres de la SVD).

Le gain de compression est un atout très important de la modélisation par base de polynômes. Cependant, il ne doit pas nous faire oublier que le choix de fonctions polynomiales est justifié, entre autre, par la facilité à donner un sens physique aux coefficients polynomiaux en degré 1. Cet avantage est moins évident lorsque l'on utilise des degrés élevés.

Nous proposons au chapitre suivant d'approximer plus localement le champ de déplacements afin de bénéficier des avantages d'une modélisation en degré faible pour analyser les champs complexes.

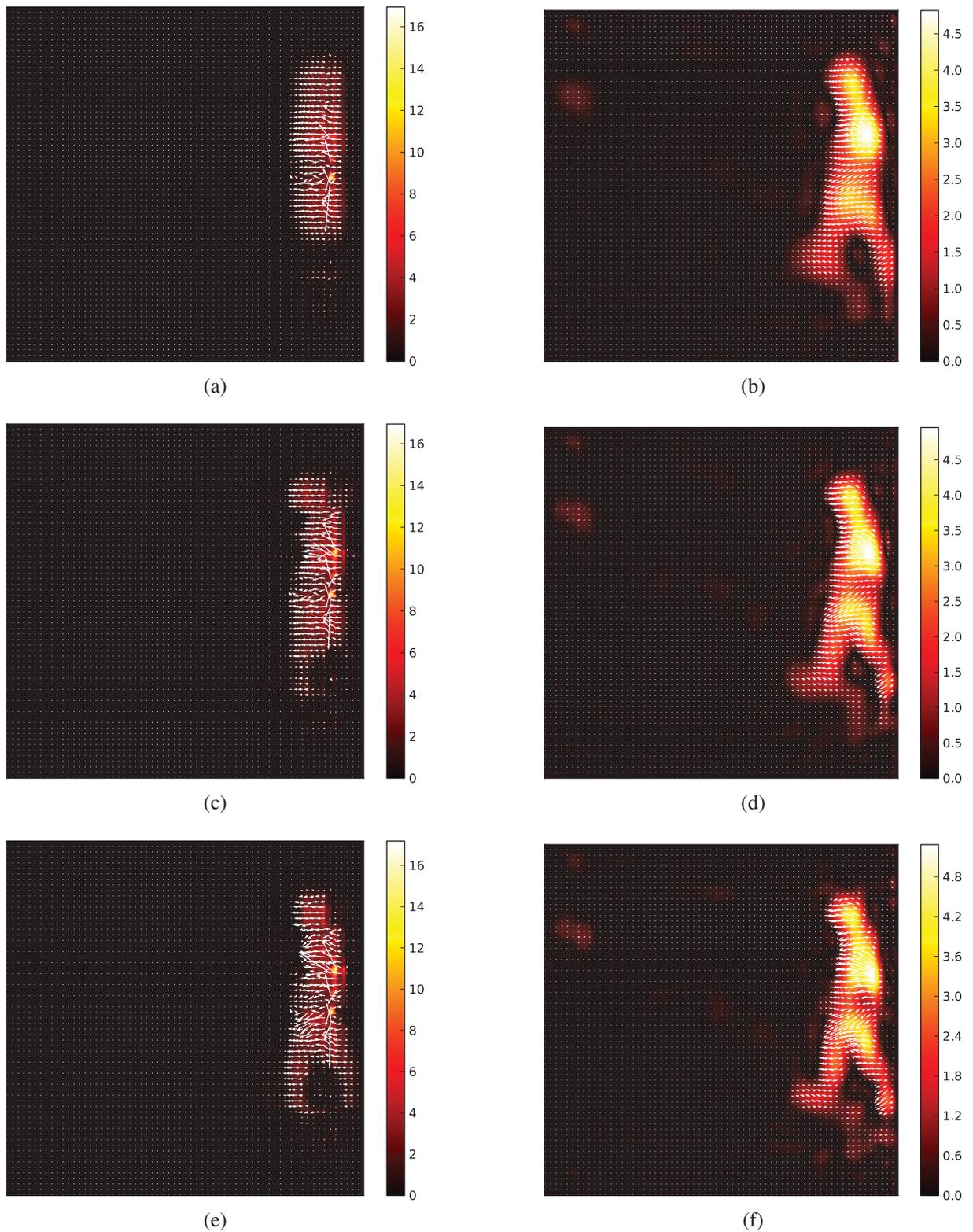


FIGURE 3.7 – Résultats d'approximation par SVD et Projection dans une base de polynômes bi-variables ; (a) Approximation avec le premier vecteur propre de la SVD ; (b) Approximation par projection sur une base de polynômes de degré 24 ; (c) Approximation avec les deux premiers vecteurs propres de la SVD ; (d) Approximation par projection sur une base de polynômes de degré 30 ; (e) Approximation avec les trois premiers vecteurs propres de la SVD ; (f) Approximation par projection sur une base de polynômes de degré 35 ;

CHAPITRE 4

APPROXIMATION HIÉRARCHISÉE DES CHAMPS DENSES

4.1 Subdiviser le champ de déplacements

L'idée de ce chapitre est de démontrer qu'une approximation de qualité d'un champ complexe est possible à obtenir même avec une base de degré faible. Partons du constat qu'un champ aussi complexe soit-il, est toujours modélisable localement à partir de translations. Il est alors possible de modéliser une partie d'un champ de déplacements avec une base de degré faible. Pour illustrer cette idée, nous proposons l'étude du cas d'école présenté sur la figure FIG 4.1(a). Il s'agit d'un champ de déplacements composé de 4 champs de déplacements affines juxtaposés. Chacune des 4 parties de ce champ peut être modélisée avec une base de degré 1. L'approximation globale de ce champ, présentée sur la figure FIG 4.1(b), fournit un résultat très imprécis. Cependant, en utilisant une subdivision judicieuse de ce champ, il est possible de l'approximer sans erreur.

L'objectif de ce chapitre est d'approximer un champ de déplacements avec précision tout en permettant d'analyser les coefficients polynomiaux plus simplement. En effet, nous avons vu au chapitre précédent que les coefficients d'une base de degré 1 peuvent être analysés en terme de translation, rotation, homothétie... Nous étudions ici différentes méthodes de subdivisions pour l'approximation de champs denses : subdivision par grille régulière, quadtree, Kd-tree, boîtes englobantes, Diagramme de Voronoï. Il est important de noter que quelle que soit la méthode de subdivision utilisée, le degré D de la base de polynômes ne doit jamais être supérieur au nombre de vecteurs constituant la plus petite dimension de la fenêtre d'approximation. Si la fenêtre d'approximation est de dimension $M \times N$ vecteurs et que N est la plus petite dimension, alors $D \geq N$.

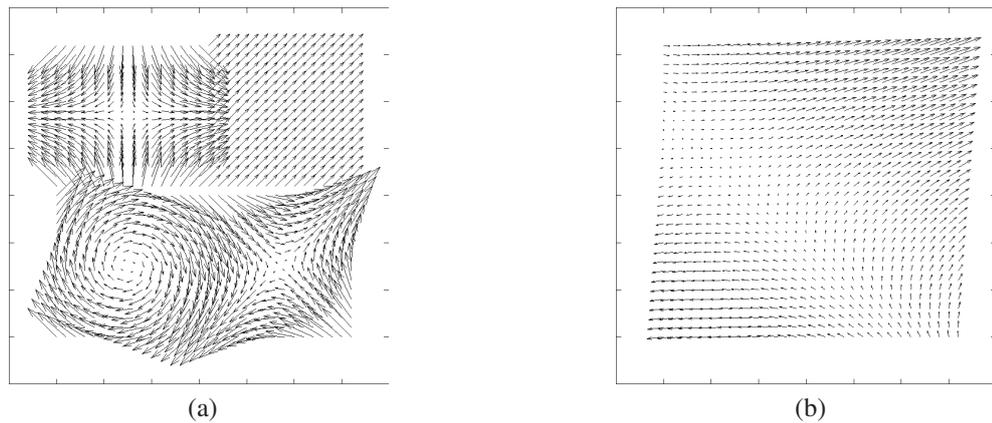


FIGURE 4.1 – Subdiviser pour mieux régner : un cas d'école ; (a) Champ de déplacements composé de 4 champs affines juxtaposés ; (b) Approximation globale du champ (a) avec une base de degré 1.

4.2 Subdivision en grille régulière

Nous commençons par l'étude de la subdivision par grille régulière. Cette méthode consiste à décomposer le domaine spatial du champ en blocs de mêmes dimensions juxtaposés les uns à côté des autres. Chaque subdivision est associée à un niveau de subdivision du domaine. Ce niveau de subdivision n représente un champ découpé en blocs de dimension $\frac{M}{2^n} \times \frac{N}{2^n}$ vecteurs si $M \times N$ est la dimension du champ initial. Lorsque les dimensions du champ ou du bloc ne sont pas paires, alors nous avons choisi, pour les subdivisions horizontales, d'arrondir à l'entier inférieur à droite et à l'entier supérieur à gauche et pour les subdivisions verticales, d'arrondir à l'entier inférieur en haut et à l'entier supérieur en bas.

Une illustration de cette méthode est donnée en figure FIG 4.3(a). Nous étudions l'erreur de modélisation par l'Erreur Quadratique Moyenne (EQM) défini en équation (2.6). La figure FIG 4.2 représente l'EQM associée à des modélisations par grille, pour des bases de degré 1, 5 et 9, en fonction du niveau de subdivision. Cette courbe nous permet de constater que l'écart de l'erreur d'approximation entre les bases de différents degrés diminue quand le niveau de subdivision augmente. Sur la figure FIG 4.3, nous illustrons l'intérêt d'une approximation par subdivision avec une grille régulière par rapport à une approximation globale. Le champ présenté en FIG 4.3(a) est modélisé par une approximation globale avec une base de degré 16 puis 24. Les cartes des erreurs quadratiques issues de ces deux modélisations sont présentées en figures FIG 4.3(b) et FIG 4.3(c). Les figures FIG 4.3(d) et FIG 4.3(e) représentent les cartes d'erreurs quadratiques obtenues par subdivision de niveau 2 (31×31 vecteurs) et 4 (8×8 vecteurs). Nous constatons que l'approximation est de meilleure qualité en utilisant une subdivision par grille régulière tout en utilisant une base de degré moins élevé. Le tableau TAB 4.1 regroupe les résultats obtenus par les différentes méthodes de subdivision étudiées dans ce chapitre.

L'inconvénient majeur de la subdivision par grille régulière est la difficulté à définir un niveau de subdivision adéquat pour tout type de champ. De plus, ce type de modélisation ne prend pas en compte l'aspect global qui est pourtant très utile pour l'analyse. Nous proposons donc plusieurs solutions permettant de conserver les avantages de la modélisation globale, c'est-à-dire sa simplicité d'analyse, et la qualité d'approximation avec une base de degré faible que permet la subdivision par grille régulière.

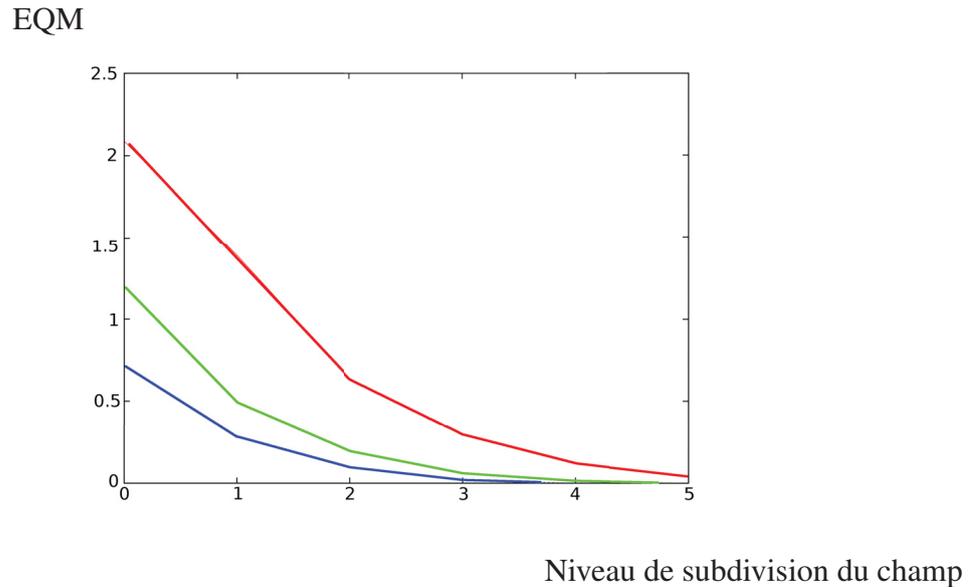


FIGURE 4.2 – EQM en fonction du niveau de subdivision et du degré de la base ; en rouge la base de degré 1 ; en vert la base de degré 5 ; en bleu la base de degré 9.

4.3 Modélisation par quadtree

La structure que nous utilisons est un arbre quaternaire, plus couramment appelé Quadtree. Le principe du quadtree est de découper récursivement le domaine en quatre sous domaines. Ce découpage est associé à un arbre, dont les fils d'un noeud sont souvent appelés Nord-ouest (NW), Nord-est (NE), Sud-ouest (SW), Sud-est (SE).

Il a été utilisé pour la première fois par Finkel R.A. et Bentley J.L. dans [R. A. Finkel 74], puis approfondi par H. Samet dans [Samet 07]. Depuis, cette structure a été utilisée dans tous les domaines de la vision par ordinateur et du traitement d'images. Cette structure peut être utilisée pour aider à résoudre localement des problèmes physiques nécessitant des modélisations complexes. C'est le cas dans [Popinet 03] où cette structure de données est utilisée sous sa forme 3D (appelée octree) afin de résoudre localement les équations d'Euler.

Nous présentons en figure FIG 4.4 un exemple très classique d'utilisation du quadtree pour la représentation d'une image binaire. Nous partons de l'image entière et on cherche à ce qu'un noeud représente uniquement une seule couleur (noir ou blanc). Si ce n'est pas le cas, on découpe récursivement l'image jusqu'à ce que cela soit le cas. Si un noeud ne représente pas une seule couleur alors on dira que le noeud est gris. S'il représente une seule couleur alors nous n'aurons plus besoin de découper l'image. On appelle alors ce noeud une feuille. Sur cette représentation, les feuilles (ou noeuds feuilles) de l'arbre sont : 1, 21, 22, 23, 24, 321, 322, 323, 324, 411, 412, 413, 414, 42, 43, 44. Le noeud I est appelé racine de l'arbre.

Dans notre cas, nous utilisons cette structure afin de modéliser un champ de déplacements en associant à chaque noeud les coefficients issus de la projection de la zone qu'ils représentent dans une base de degré donné. Le champ est subdivisé tant que l'erreur d'approximation est supérieure à un seuil fixé. Nous choisissons l'erreur quadratique moyenne comme mesure d'erreur.

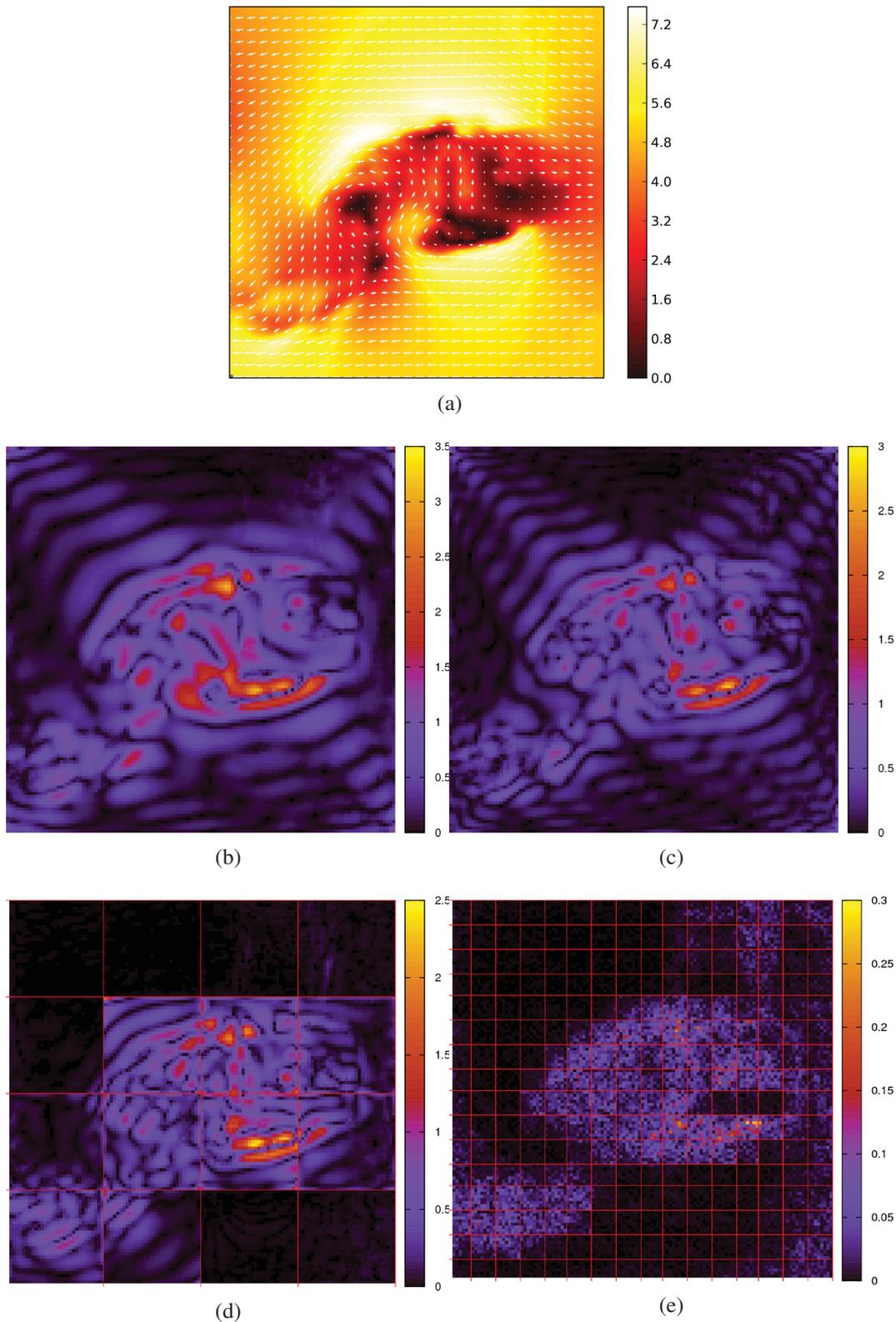


FIGURE 4.3 – Représentation de la carte d’erreurs quadratiques d’un champ pour différentes modélisations globales et par subdivision par grilles régulières ; (a) Champ 131 extrait de la séquence Cylindre-carré de 124×124 vecteurs (cf. ANNEXE A) ; (b) Modélisation globale du champ test avec une base de polynômes de degré 16 ; (c) Modélisation globale du champ test avec une base de polynômes de degré 24 ; (d) Modélisation du champ test par subdivision de niveau 2 (31×31 vecteurs) avec une base de polynômes de degré 5 ; (e) Modélisation du champ test par subdivision de niveau 4 (8×8 vecteurs) avec une base de polynômes de degré 5

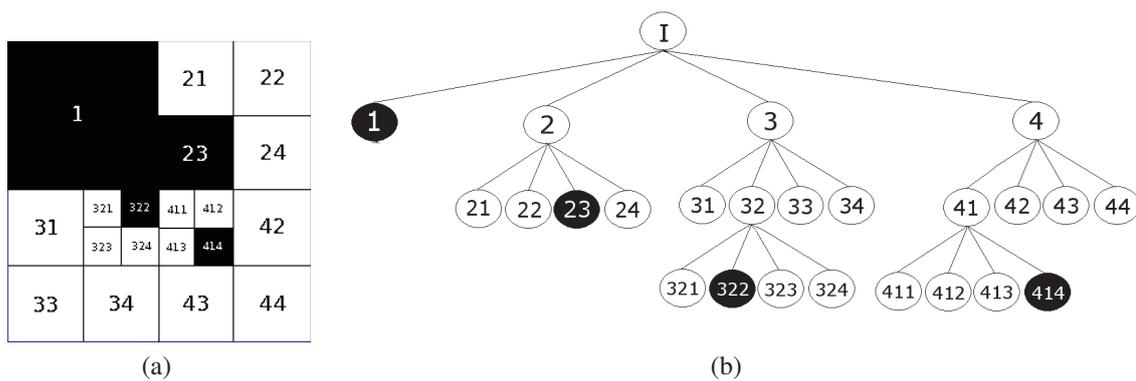


FIGURE 4.4 – Représentation d’une image binaire par un quadtree

L’algorithme est le suivant :

1. Calculer l’approximation du champ par projection sur la base ;
2. Reconstruire le champ à partir du champ approximé ;
3. Calculer la EQM entre le champ initial et le champ approximé ;
4. Comparer la EQM au seuil fixé
 - Si la EQM est inférieure au seuil fixé alors arrêter le processus ;
 - Sinon découper le champ en quatre parties et reproduire le processus sur chaque partie.

On peut ainsi garantir une certaine qualité de reconstruction au niveau de décomposition correspondant aux feuilles du quadtree tout en permettant d’avoir une représentation plus globale en remontant dans l’arbre.

Nous présentons en figure FIG 4.5, l’erreur quadratique locale associée à trois exemples de modélisation d’un champ avec un quadtree et des bases de degré 9, 5 et 1. Le quadtree permet donc d’obtenir une grille adaptative en fonction du degré de la base utilisée. Nous constatons que pour de nombreuses zones, la subdivision n’est pas au niveau de subdivision le plus élevé, ce qui signifie que ces zones sont modélisées de manière assez précise. Dans ce cas, le seuil est fixé à $EQM = 0.1$.

Cependant, la subdivision récursive en 4 parties peut aboutir à une sur-segmentation. Afin de limiter un peu cette sur-segmentation, nous étudions un autre algorithme de subdivision utilisant cette fois un arbre binaire.

4.4 Subdivision par kd-tree

Le Kd-tree est une structure de données inventée par J.L. BENTLEY [Bentley 75]. Cette structure permet de stocker, dans un arbre binaire, un ensemble de points dans un espace de dimension k . Contrairement au quadtree, le kd-tree permet de diviser l’espace de manière irrégulière. Dans le cas d’un domaine spatial en 2D, on découpe par une droite parallèle à l’un des axes du domaine (axe x ou axe y). En général, on choisit une droite médiane au domaine. Dans le cas 3D, le domaine est découpé par un plan et en dimension supérieure par un hyperplan. La figure FIG 4.6 nous permet d’introduire par un exemple cet algorithme. Il s’agit de séparer un ensemble de six points d’un domaine spatial en dimension 2. Ce domaine peut être séparé soit horizontalement soit verticalement.

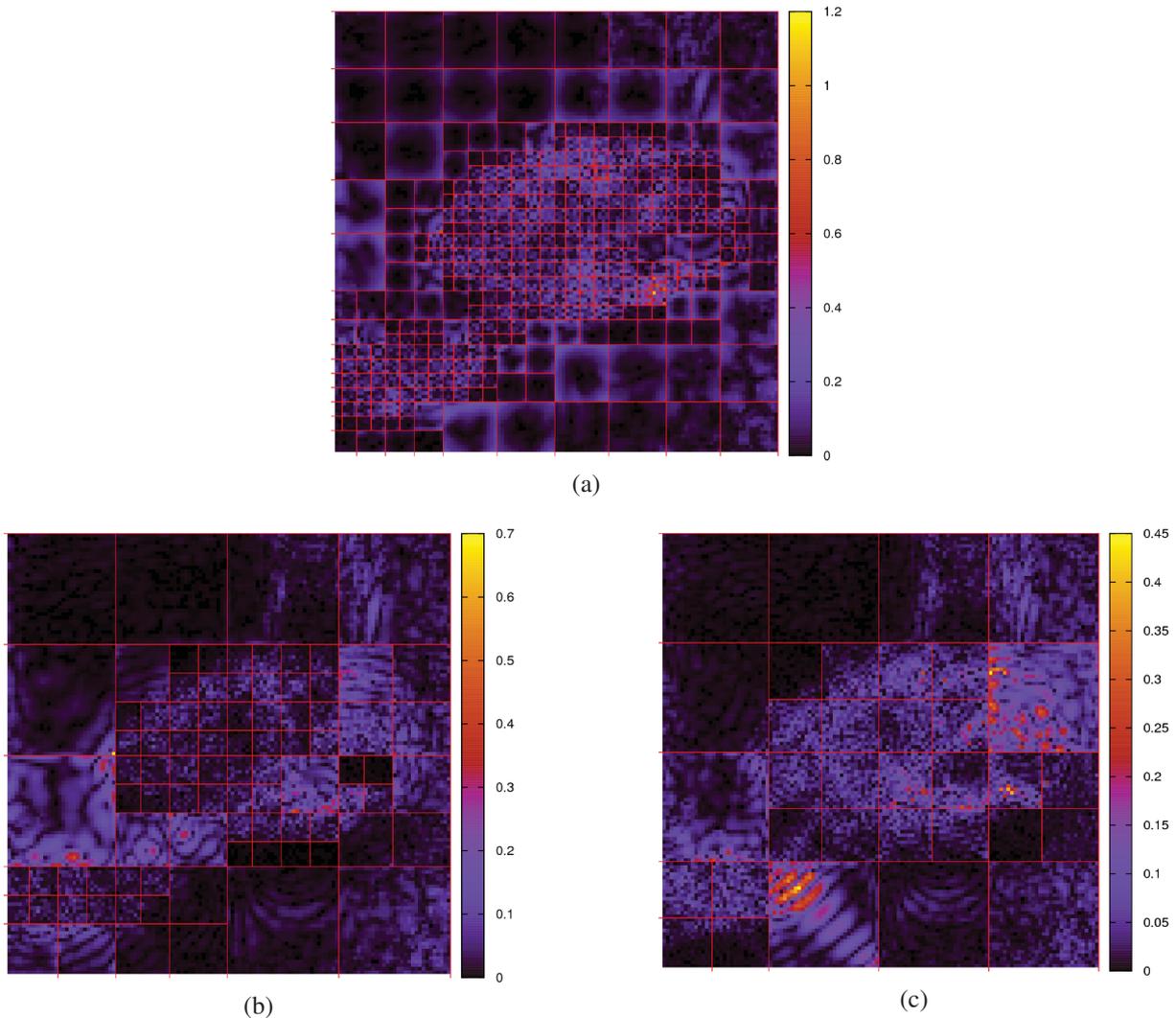


FIGURE 4.5 – Représentation de la carte d’erreurs quadratiques associée à la modélisation du champ test à l’aide d’un quadtree avec 3 bases de degrés différents ; (a) Quadtree avec une base de degré 1 ; (b) Quadtree avec une base de degré 5 ; (c) Quadtree avec une base de degré 9.

L’algorithme est le suivant :

1. *Trier les points du domaine par leur position suivant l’axe horizontal (resp. vertical) si le noeud est pair (resp. impair) ;*
2. *Choisir le point médian ;*
3. *Découper le domaine spatial en deux sous domaines selon l’axe vertical (resp. horizontal) si le noeud est pair (resp. impair) et stocker l’ensemble des points à gauche (resp. au dessus) dans le noeud fils de droite et l’ensemble des points à droite (resp. en dessous) dans le noeud fils de gauche ;*
4. *Recommencer le processus à partir de 1 sur chaque noeud fils tant que l’ensemble des points n’est pas nul. Si l’ensemble est nul, alors il n’y a plus de points dans cette région du domaine spatial initial ;*

On peut citer [Wald 06] pour un exemple d’utilisation du kd-tree dans le contexte du lancer de rayon ainsi qu’une étude sur la complexité de certains algorithmes.

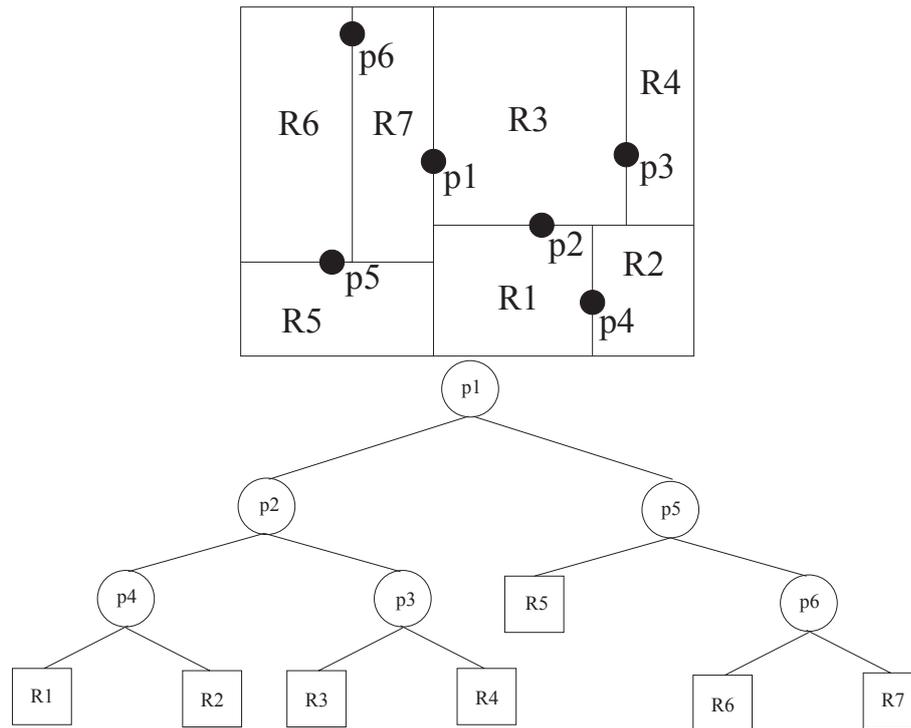


FIGURE 4.6 – Exemple de Kd-tree et son arbre associé ; les p_i représentent les points du domaine et les R_k les régions ne contenant plus de points

Dans notre cas, nous utilisons cette structure pour décomposer le champ. Dans un premier temps, nous proposons d'utiliser l'algorithme kd-tree médian suivant :

1. Calculer la projection du champ sur la base ;
2. Reconstruire le champ à partir des coefficients issus de la projection ;
3. Calculer l'erreur entre le champ original et le champ approximé ;
4. Comparer la mesure de l'erreur au seuil fixé :
 - Si la mesure d'erreur est inférieure au seuil alors arrêter le processus ;
 - Sinon découper le domaine en deux parties égales par une droite horizontale (resp. verticale) si la profondeur du noeud est impaire (resp. paire). Recommencer le processus sur les deux sous-domaines.

Nous présentons sur la figure FIG 4.7, la carte d'erreurs quadratiques associée à trois exemples de modélisation d'un champ avec un kd-tree et des bases de degré 9, 5 et 1.

L'avantage de cet algorithme est sa rapidité de calcul. Cependant, comme pour le quadtree, le découpage est effectué systématiquement par rapport à un axe médian au domaine dans l'une ou l'autre des directions. On peut donc facilement aboutir à des partitions de l'espace très proches.

C'est pourquoi nous apportons une modification à cet algorithme. À l'étape 4, lorsque l'erreur de modélisation est supérieure au seuil fixé, le champ est subdivisé. Jusqu'alors, la subdivision consistait à découper le champ par rapport au plan médian soit horizontalement, soit verticalement selon la profondeur de l'arbre. Nous proposons de chercher une meilleure subdivision. En effet, la recherche de la meilleure subdivision nécessite de tester toutes les possibilités, ce qui est évidemment trop coûteux en temps de calcul. Nous avons donc choisi de tester six combinaisons de subdivisions, trois horizontales et trois verticales, présentées en figure FIG

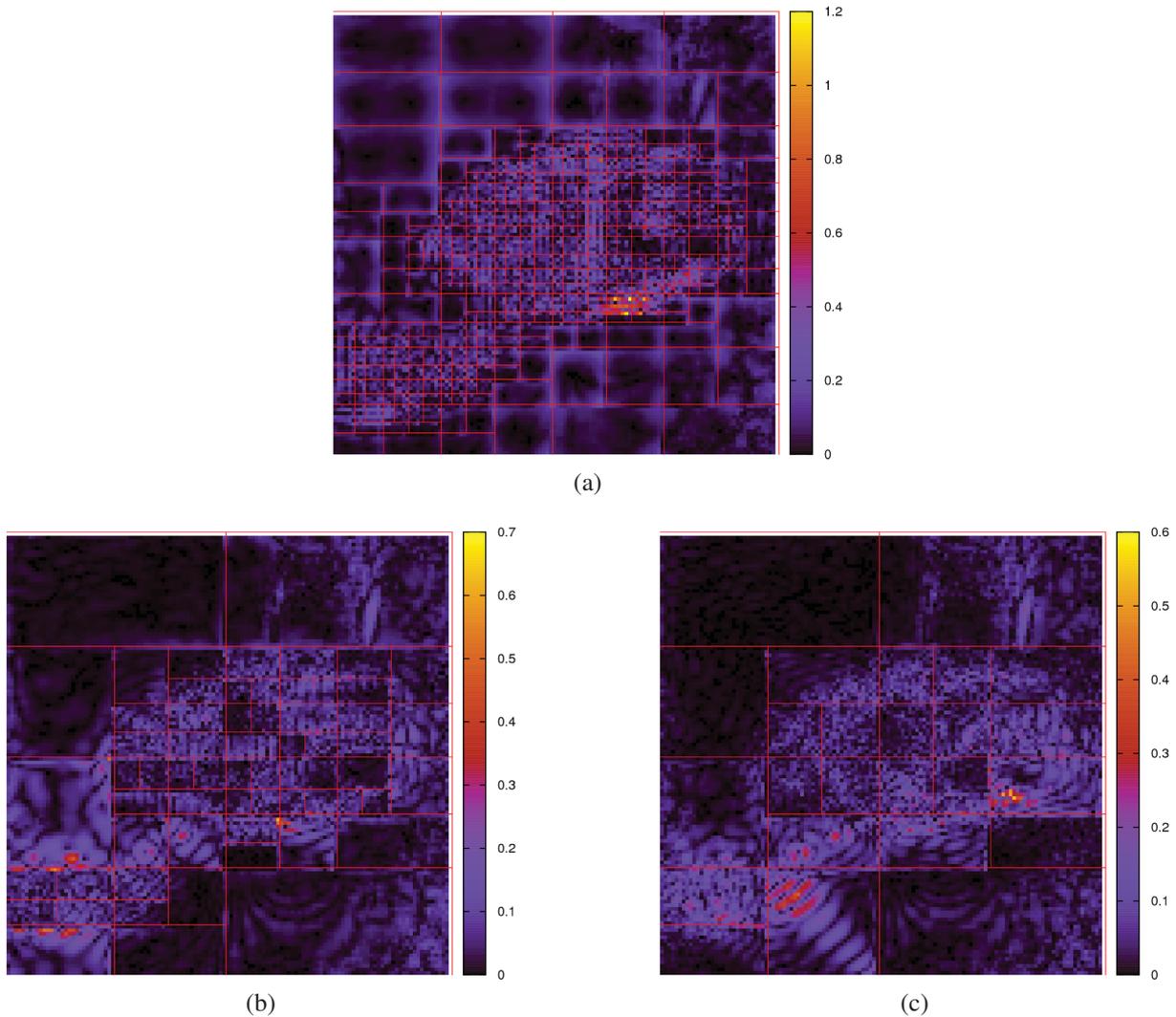


FIGURE 4.7 – Représentation de la carte d’erreurs quadratiques associée à la modélisation du champ test à l’aide d’un kd-tree médian avec 3 bases de degrés différents ; (a) kd-tree médian avec une base de degré 1 ; (b) kd-tree médian avec une base de degré 5 ; (c) kd-tree médian avec une base de degré 9

4.8. Dans chaque cas, les étapes 1, 2 et 3 sont appliquées à chacun des deux sous-domaines. La subdivision dont la somme des erreurs est minimale est retenue. L’algorithme est alors :

1. Calculer la projection du champ sur la base ;
2. Reconstruire le champ à partir des coefficients issus de la projection ;
3. Calculer l’erreur entre le champ original et le champ approximé ;
4. Comparer la mesure de l’erreur au seuil fixé :
 - Si la mesure d’erreur est inférieure au seuil alors arrêter le processus ;
 - Sinon, découper le champ avec chacune des six configurations sélectionnées et dans chaque cas
 - (a) Projecter le champ sur une base de polynômes ;
 - (b) Reconstruire le champ à partir des coefficients issus de la projection ;
 - (c) Chercher la subdivision qui minimise la somme des erreurs et conserver les coefficients associés ;

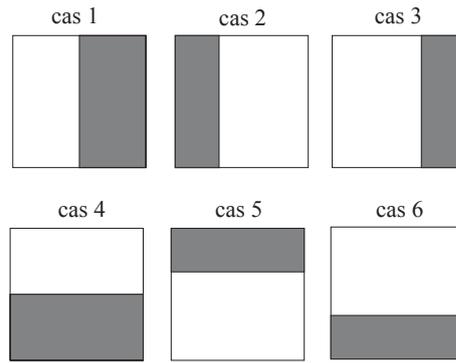


FIGURE 4.8 – Illustration des six combinaisons possibles du kd-tree irrégulier

(d) *Rejeter les autres subdivisions et recommencer le processus sur les deux sous-domaines.*

Dans ce cas, le choix de découper horizontalement ou verticalement n'est plus dicté par la profondeur de l'arbre. Il faudra donc penser à stocker cette direction ainsi que la position de la droite dans la structure de données. Nous présentons en figure FIG 4.9, la représentation de l'erreur quadratique locale associée à la modélisation du champ test à l'aide d'un kd-tree de ce type avec 3 bases de degré différent. On peut constater que la subdivision de l'espace est très intéressante puisque l'on obtient une grille irrégulière. Nous appelons cet algorithme kd-tree irrégulier pour y faire référence. Cependant, dans le tableau TAB 4.2, on peut constater que du point de vue de la précision de l'approximation, cet algorithme ne fournit pas de meilleurs résultats que le kd-tree médian.

Nous présentons des résultats visuels de la qualité de reconstruction du quadtree et du kd-tree sur un extrait du champ en figure FIG 4.10. La modélisation est bien effectuée sur le champ entier, mais nous n'en présentons qu'un extrait afin de le rendre plus lisible.

Le tableau TAB 4.1 regroupe les résultats qualitatifs des différentes techniques de subdivision étudiées dans cette section. Ces différentes méthodes de subdivisions sont comparées en fonction de l'erreur d'approximation globale obtenue en fonction du degré de la base utilisée. Nous ajoutons aussi la mesure de l'erreur angulaire (E_{Θ}) définie par

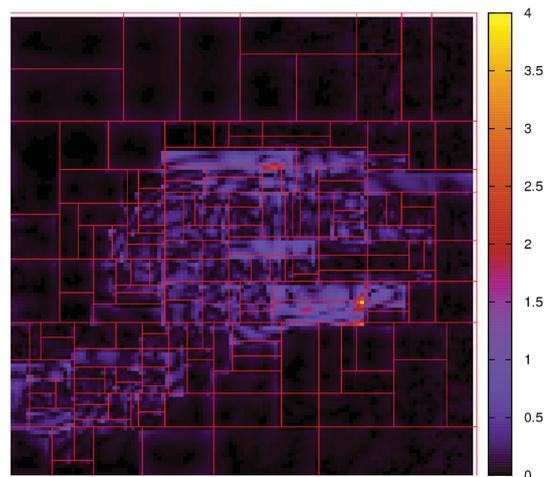
$$E_{\Theta} = \sum_{\Omega} \frac{1}{2} | \sin(\Theta(x_1, x_2) - \tilde{\Theta}(x_1, x_2)) | \quad (4.1)$$

où $\Theta(x_1, x_2)$ représente l'orientation entre $[-\frac{\pi}{2}; \frac{\pi}{2}]$ d'un vecteur $\mathcal{C}(x_1, x_2)$ du champ.

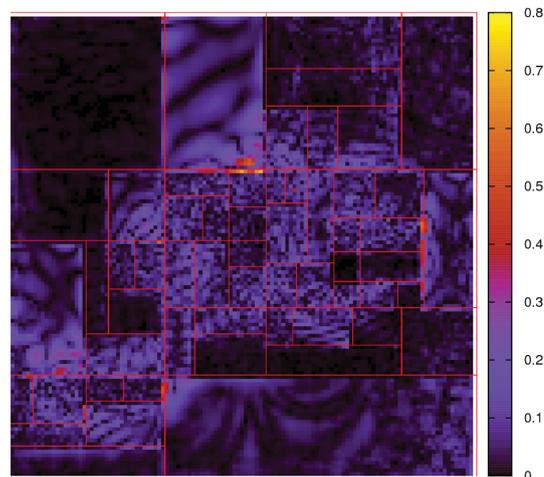
Dans le cas des grilles régulières, afin de pouvoir être facilement comparés aux résultats du quadtree, les découpages utilisés sont des subdivisions de l'espace par puissance de 2 que nous appelons niveaux de subdivision. Nous rappelons que l'erreur globale est calculée par l'EQM. Le gain de compression (exprimé en pourcentage) est donné par :

$$\tau = 1 - \left(\frac{(D+1)(D+2)}{2 \times M \times N} \right) \times 100 \quad (4.2)$$

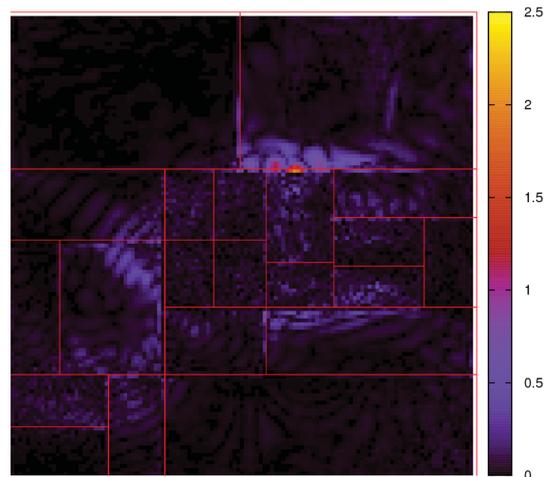
On peut alors constater que les différentes méthodes de subdivisions étudiées nous permettent d'obtenir une approximation de qualité tout en conservant un taux de compression impor-



(a)

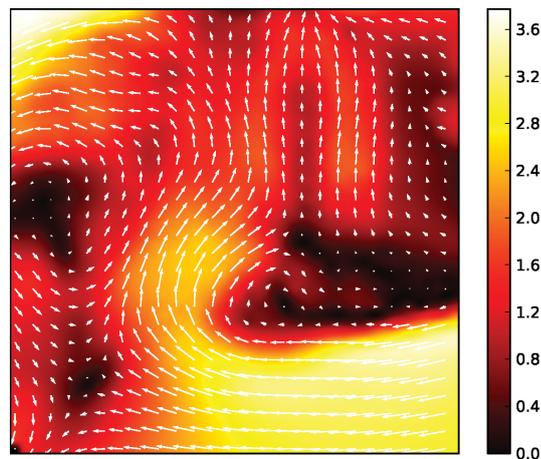


(b)

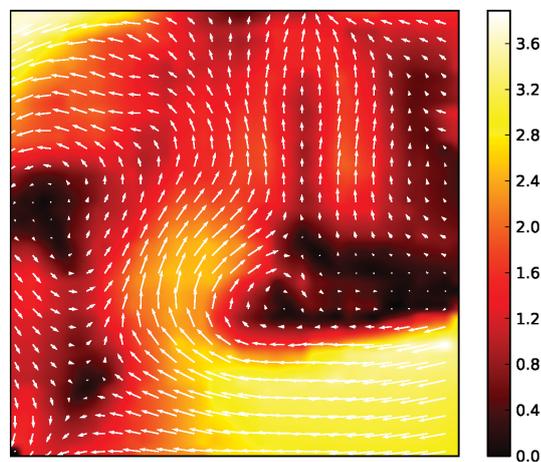


(c)

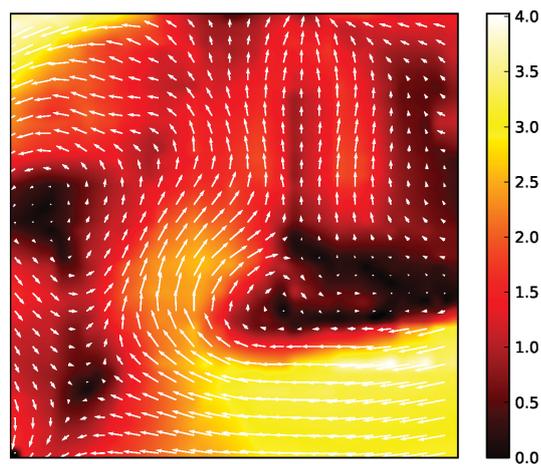
FIGURE 4.9 – Représentation de l'erreur quadratique locale associée à la modélisation du champ test à l'aide d'un kd-tree irrégulier avec 3 bases de degré différent ; (a) kd-tree irrégulier avec une base de degré 1 ; (b) kd-tree irrégulier avec une base de degré 5 ; (c) kd-tree irrégulier avec une base de degré 9



(a)



(b)



(c)

FIGURE 4.10 – Résultats visuels de la qualité d'approximation du quadtree et du kd-tree ; (a) Extrait du champ test ; (b) Extrait de l'approximation du champ avec un quadtree et une base de degré 1 ; (c) Extrait de l'approximation du champ avec un kd-tree median et une base de degré 1 ;

tant. Le quadtree et le kd-tree permettent d'obtenir une grille adaptative de manière automatique. Cela permet d'améliorer la compression de l'approximation par grille régulière.

En effet, pour une base de degré 1, le gain de compression passe de 80% pour une grille régulière à environ 92% avec un quadtree ou un kd-tree médian. Le seul paramètre à régler est le seuil de qualité permettant de décider si le champ doit être subdivisé.

Méthode	Degré	Niveau	Gain	EQM	E_{\ominus}
globale	5	0	99.863	1.167	0.210
globale	9	0	99.642	0.631	0.112
globale	16	0	99.005	0.396	0.080
globale	20	0	98.498	0.308	0.069
grille régulière	9	3	77.107	0.020	0.005
grille régulière	5	4	65.036	0.014	0.004
grille régulière	1	5	80.021	0.041	0.011
quadtree	9	3	87.838	0.034	0.007
quadtree	5	4	87.981	0.036	0.008
quadtree	1	5	92.254	0.072	0.015
kd-tree median	9	3	91.058	0.040	0.008
kd-tree median	5	4	91.669	0.045	0.009
kd-tree median	1	5	92.683	0.081	0.015
kd-tree irrégulier	9	3	92.488	0.053	0.009
kd-tree irrégulier	5	4	92.215	0.050	0.009
kd-tree irrégulier	1	5	96.449	0.135	0.032

TABLEAU 4.1 – Evaluation de l'erreur d'approximation et du gain de compression pour les méthodes globales, de subdivision par grille régulière, par quadtree et par kd-tree pour différents degrés et différents niveaux de hiérarchiques

Nous avons constaté que la subdivision par quadtree est trop systématique puisqu'elle divise en 4 le domaine même si seulement une petite partie du domaine ne répond pas au critère de subdivision. Cela permet d'aboutir rapidement à une solution répondant au problème de minimisation de l'erreur quadratique globale, au prix d'un sur-partitionnement de l'image. Ce sur-partitionnement peut être limité par l'utilisation d'un kd-tree, au prix cette fois-ci d'un sur-coût calculatoire dû à la recherche de la meilleure droite de partitionnement de l'espace en deux. Nous proposons une autre approche de la subdivision dans la section suivante.

4.5 Modélisation obtenue par subdivision à partir de points singuliers du champ de déplacements

Les points singuliers présentent un grand intérêt en analyse de mouvement fluide. En effet, le champ qui entoure ces points est très intéressant et peut représenter des tourbillons ou différentes autres structures très informatives pour l'analyse. Dans le domaine de l'analyse des mouvements humains, il existe des méthodes permettant de détecter certains points particuliers du corps humain [Panagiotakis 04]. Ces points peuvent être utilisés pour analyser une séquence représentant un mouvement humain.

Nous cherchons donc à subdiviser un champ de déplacements à partir d'une liste de points. Il existe de nombreuses techniques permettant de subdiviser un domaine à partir d'un nombre défini de points de ce domaine. On peut citer le kd-tree, présenté dans la section précédente ou le diagramme de Voronoï que nous présentons dans la suite de ce chapitre.

Cependant, dans notre cas, la zone qui entoure chaque point est souvent très importante du point de vue de l'analyse. Nous appelons ces zones zones d'intérêt. Il faudrait donc privilégier les solutions qui partitionnent le domaine en séparant ces zones d'intérêt sans les sur-partitionner. Le kd-tree qui a pourtant été inventé pour répondre au problème de subdivision d'un domaine en séparant un certain nombre de points, ne permettra pas de résoudre notre problème dans ce cas précis. L'exemple de partition présenté sur la figure FIG 4.11, montre que dans certains cas il n'est pas possible de séparer les différentes zones d'intérêt par un kd-tree sans en obtenir un sur-partitionnement. En effet, il n'existe aucune droite horizontale ou verticale qui permette de séparer les structures sans en intersecter une autre. C'est pourquoi, nous nous tournons vers d'autres solutions. Notre objectif est de simplifier l'analyse d'un champ de déplacements.

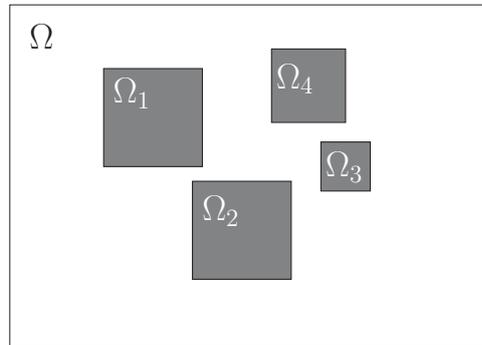


FIGURE 4.11 – Exemple de zones non séparables par une droite horizontale ou verticale ; Cela implique une impossibilité de partitionner l'espace en séparant les zones sans sur-partitionner les zones d'intérêt

Nous présentons une première technique très intuitive utilisant des boîtes englobantes. La boîte englobante d'une région est le domaine spatial de forme rectangulaire englobant toute cette région. À l'intérieur de cette boîte, la projection est effectuée de manière globale. Après avoir introduit cette méthode, nous l'étendons au cas des boîtes englobantes orientées.

Ensuite, ces boîtes englobantes sont utilisées pour approximer les régions issues d'une subdivision obtenue par un diagramme de Voronoï.

4.5.1 Les boîtes englobantes

Nous proposons un modèle simple basé sur l'utilisation de boîtes englobantes. Le champ global est modélisé par projection dans une base de polynômes. Les coefficients de projection sont stockés à la racine d'un arbre. Cette arbre aura une profondeur de 2, et autant de fils qu'il y a de points singuliers dans le champ. À chaque fils de la racine de l'arbre est associé un point singulier du domaine et sa région. Ces régions sont modélisées par une boîte englobante. Les boîtes englobantes pourront être de dimensions fixes (définies par un utilisateur) si aucune information n'est connue. Dans notre cas, les points singuliers sont associés à une région spatiale

à l'intérieur de laquelle il est possible d'utiliser un modèle linéaire. Les dimensions de ces régions définissent les boîtes englobantes associées aux points singuliers. De cette manière, il est possible de se ramener à une analyse globale du champ telle que cela a été fait au chapitre 3 pour analyser le mouvement principal du champ à partir des informations stockées à la racine de l'arbre. Pour des informations plus locales, chaque fils de la racine de l'arbre apporte des

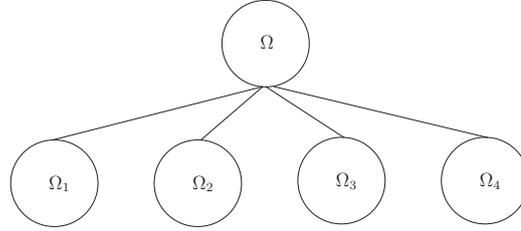


FIGURE 4.12 – Arbre associé à la modélisation de la figure FIG 4.11

précisions. La figure FIG 4.12 présente l'arbre associé à la modélisation de la partition proposée en figure FIG 4.11. Une illustration des résultats d'approximation de ce type de méthode est présentée en figure FIG 4.13. En figure FIG 4.13(a), nous présentons le champ à approximer sur lequel on a représenté en noir les boîtes englobantes. En FIG 4.13(b), une approximation du champ avec une base de degré 16 associée à la racine de l'arbre et une base de degré 1 pour chacun de ces fils, en FIG 4.13(c), une approximation du champ avec une base de degré 9 associée à la racine de l'arbre et une base de degré 5 pour chacun de ces fils.

Nous proposons d'étendre cette modélisation avec des boîtes englobantes orientées. Le champ synthétique en figure FIG 4.14a représente un exemple de structures que l'on veut modéliser par des boîtes englobantes indépendamment du reste du champ. Il s'agit d'un champ synthétique dédié à l'étude des boîtes orientées. Chaque structure qui compose le champ est un mouvement affine.

Nous commençons par modéliser chaque région avec une boîte englobante non orientée. Le résultat de cette modélisation est présenté en figure FIG 4.16(b), (e), (h), (k) et (n). On peut constater que les mouvements sont globalement bien reconstruits. Du point de vue de l'analyse, on peut reconnaître s'il s'agit d'une rotation ou d'une homothétie par exemple. Cependant, la norme des vecteurs n'est pas satisfaisante en raison de la répartition de l'énergie sur l'ensemble du domaine.

Nous proposons d'améliorer la qualité d'approximation. Les difficultés de modélisation sont dues à la présence de vecteurs nuls dans le champ et dans ce cas précis dans la boîte englobante. Nous cherchons à obtenir une boîte englobante minimisant le nombre de points nuls. Nous orientons donc chaque boîte englobante (rectangulaire) dans la direction principale de la forme. Pour trouver l'orientation de la forme, nous utilisons une méthode basée sur les moments centrés μ d'ordre p, q calculés par la formule :

$$\mu_{p,q} = \sum_{x_1, x_2 \in \Omega} I_b(x_1, x_2) (x_1 - \bar{x}_1)^p (x_2 - \bar{x}_2)^q \quad (4.3)$$

où $I_b(x_1, x_2) = 0$ si le mouvement est nul et 1 sinon. L'angle θ qui représente l'orientation de la forme se calcule par les formules suivantes :

$$\theta = \tan^{-1} \left(\frac{\lambda_{max} - \mu_{2,0}}{\mu_{1,1}} \right) \quad (4.4)$$

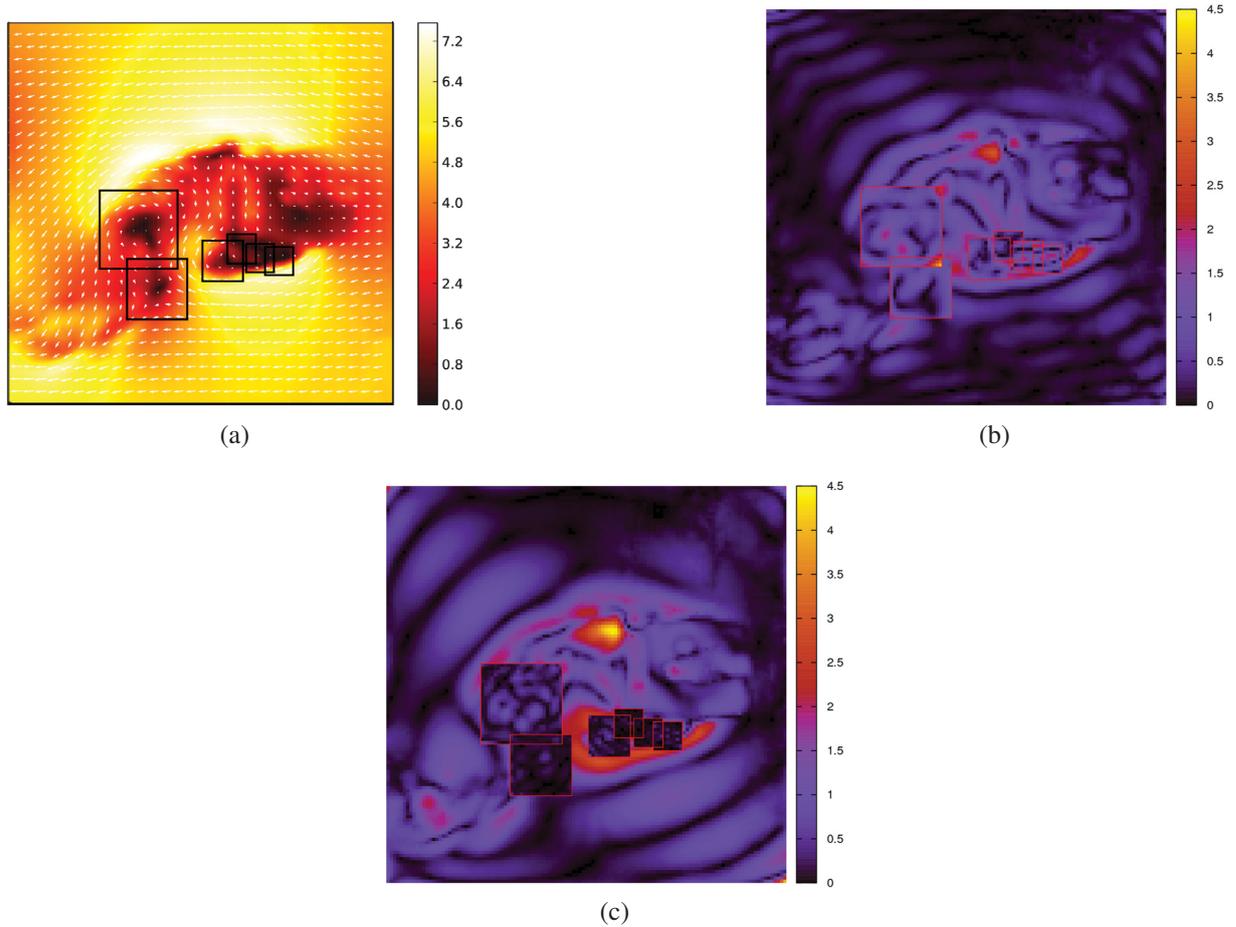


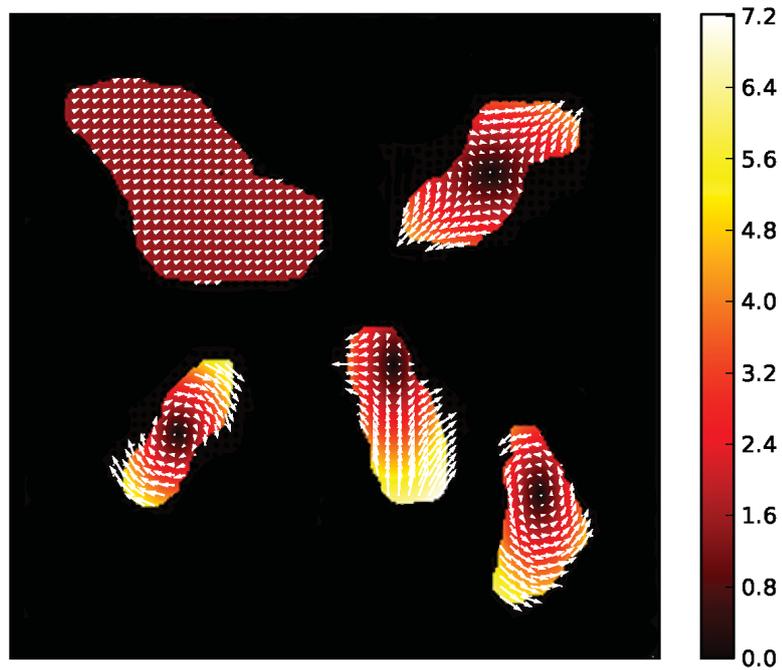
FIGURE 4.13 – Représentation de la carte d’erreurs quadratiques associée à la modélisation du champ test à l’aide des boîtes englobantes ; (a) Champ test à approximer sur lequel on a représenté en noir les boîtes englobantes ; (b) Modélisation globale avec une base de degré 16 et chaque boîte avec une base de degré 1 ; (c) Modélisation globale avec une base de degré 9 et chaque boîte avec une base de degré 5 ;

avec

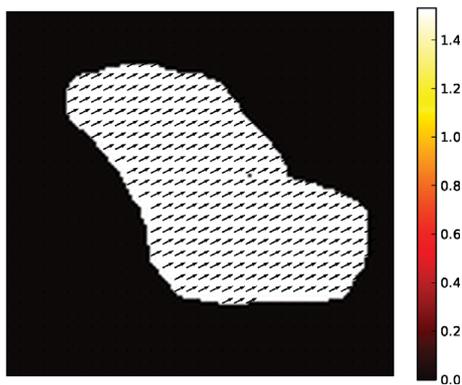
$$\lambda_{max} = \frac{1}{2} (\mu_{2,0} + \mu_{0,2}) + \frac{1}{2} \sqrt{\mu_{2,0}^2 + \mu_{0,2}^2 - 2\mu_{0,2}\mu_{2,0} + 4\mu_{1,1}^2} \quad (4.5)$$

Une fois cette phase de calcul de l’orientation de la forme effectuée, nous calculons la rotation de chaque composante connexe afin d’en déduire une nouvelle boîte englobante.

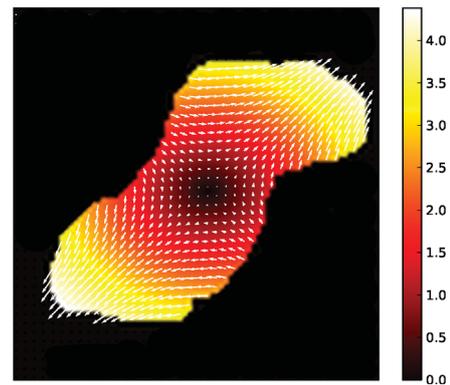
A ce stade, nous sommes confrontés au calcul d’une rotation discrète qui en utilisant une simple matrice de rotation n’est pas bijective. Nous avons choisi d’utiliser la méthode des angles pythagoriciens introduite par Eric Andres dans [Andres 91] et dont la bijectivité est prouvée dans [Jacob 95] pour calculer la rotation discrète. Le principe est de découper une rotation d’un angle Θ quelconque, en une succession de rotations d’angles pythagoriciens. Θ est un angle pythagoricien si $\Theta = atan(a/b)$ avec $a = 2k + 1$ et $b = 2k(k + 1)$, a, b, k des entiers, avec $k \neq 0$.



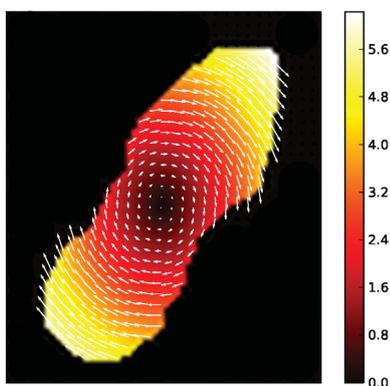
(a)



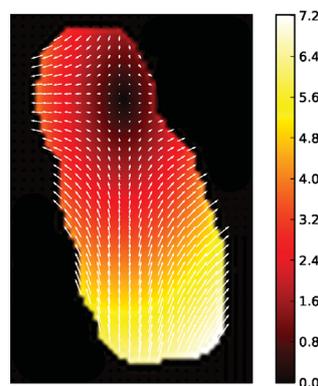
(b)



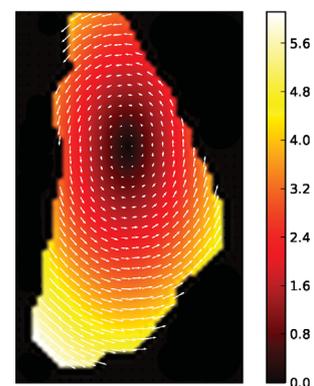
(c)



(d)



(e)



(f)

FIGURE 4.14 – ; (a) Champ synthétique composé de 5 mouvements affines séparés ; (b) à (f) Mouvement affine de degré 1 dans l'ordre de lecture classique.

Afin de trouver les différentes valeurs des angles pythagoriciens, nous utilisons la formule suivante pour obtenir k , duquel nous déduisons $a = 2k + 1$ et $b = 2k(k + 1)$, et donc l'angle.

$$k = \lfloor \frac{1 - \tan(\Theta) + \sqrt{\tan(\Theta)^2 + 1}}{2 \tan(\Theta)} + \frac{1}{2} \rfloor \quad (4.6)$$

avec $\lfloor \rfloor$ la partie entière.

Par cette méthode, nous pouvons obtenir uniquement les angles commençant par la série suivante : 38.86, 22.62, 16.26, 12.68, ... qui est une suite qui tend vers 0 lorsque k tend vers l'infini. Cette difficulté peut être surmontée en composant plusieurs rotations pythagoriciennes. Dans nos expériences, nous ne réalisons que trois rotations. Nous obtenons ainsi une rotation bijective.

Après avoir effectué la rotation du support de notre champ de déplacements, nous devons appliquer la même rotation aux vecteurs. En effet, on peut voir en figure FIG 4.15(a) et FIG 4.15(b) que lorsque le support du champ a été tourné, le mouvement après rotation du domaine ne représente plus le même mouvement que le champ d'origine (ici une rotation affine). Il faut alors appliquer à tous les vecteurs du champ la même rotation que celle qui a été appliquée au domaine spatial du champ. Ainsi, on obtient la figure FIG 4.15(c) qui représente bien une rotation affine.

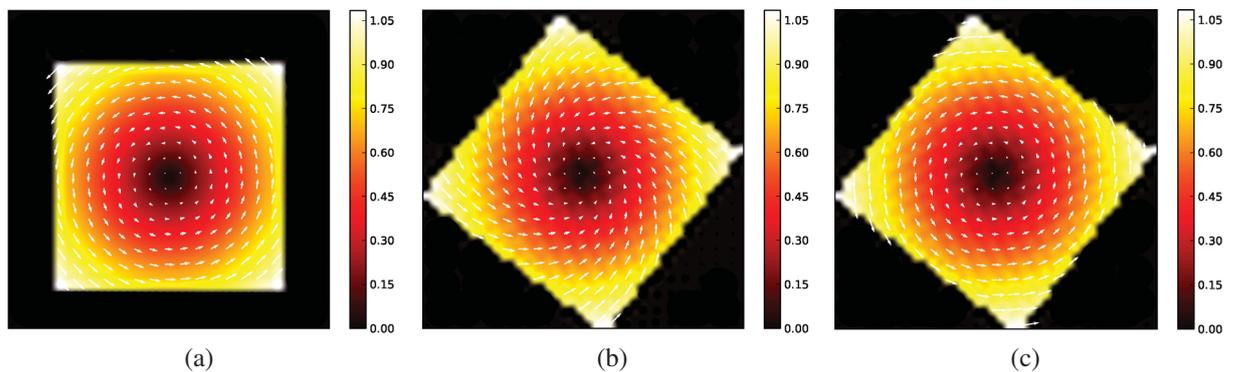


FIGURE 4.15 – Étape de la rotation d'un champ synthétique affine (rotation) ; (a) Champ original ; (b) Rotation discrète du support du champ (ne ressemble plus à une rotation) ; (c) Rotation discrète du support du champ ainsi que des vecteurs.

Cette méthode est à rapprocher de celle utilisée pour les bandelettes de première génération [Le Pennec 00], [Pennec 02]. Les bandelettes de première génération s'appuient sur une détection préalable des contours (singularités) d'une image. Ces contours sont classés en contours majoritairement horizontaux ou verticaux. L'algorithme de feu de prairie [Blum 73] est utilisé afin d'obtenir des zones presque rectangulaires autour de ces contours. Ces formes suivant le contour sont ensuite transformées afin de s'adapter au mieux à la transformée en ondelette 2D séparable. Dans notre cas, nous n'utilisons pas une base d'ondelettes et la seule transformation est une rotation discrète bijective, mais l'idée générale reste la même.

Nous appliquons cette rotation sur les composantes connexes du champ présenté sur la figure FIG 4.14 afin de supprimer un maximum de vecteurs nuls autour de chaque composante. La figure FIG 4.16 présente pour chaque composante connexe, le mouvement original à gauche,

l'approximation par boîte englobante au milieu et l'approximation par boîtes englobantes orientées à droite. Cette méthode permet d'améliorer sensiblement les résultats d'approximation, mais ne permet toujours pas d'obtenir une modélisation parfaite des mouvements affines non-denses.

De plus, à cause de la rotation des vecteurs, deux translations identiques ayant un domaine de définition dont l'orientation est différente, donnent un modèle différent. Cette solution rend donc plus complexe l'analyse des champs.

4.6 Le diagramme de Voronoï

Un diagramme de Voronoï est défini à partir d'un ensemble de n points ou sites p_1, p_2, \dots, p_n de l'espace euclidien. On associe à chaque site p_i la région $V(p_i)$. Cette région est constituée de tous les points p du domaine plus proche de p_i que de n'importe quel autre site par rapport à une distance d .

Si l'on utilise la distance L_1 (ou distance de Manhattan) d devient :

$$d(p, q) = \|p_x - q_x\| + \|p_y - q_y\| \quad (4.7)$$

Dans les cas Euclidien, la distance L_2 est définie par :

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (4.8)$$

En général, avec la distance L_n :

$$d(p, q) = \sqrt[n]{(p_x - q_x)^n + (p_y - q_y)^n} \quad (4.9)$$

On peut donc définir la région $V(s_i)$ par l'ensemble des points p vérifiant :

$$V(s_i) = \{p \mid d(p, s_i) \leq d(p, s_j), \text{ pour tout } j \neq i\} \quad (4.10)$$

On dit que $V(s_i)$ est la région de Voronoï de centre s_i . La région $V(s_i)$ est l'intersection d'un nombre fini de demi-espaces limités par les hyperplans médiateurs des segments $[s_i, s_j]$, $j = 1, \dots, n, j \neq i$. C'est donc un polytope convexe, éventuellement non borné. Les $V(s_i)$ constituent le diagramme de Voronoï. Dans un diagramme de Voronoï, on définit des arrêtes et des sommets.

Définition 1 Une arrête est définie par l'ensemble des points appartenant à la demi-droite séparant deux régions.

Définition 2 Un sommet est défini comme un point à la frontière entre trois (ou plus) régions.

Le diagramme de Voronoï est très utilisé notamment en raison de sa dualité avec le diagramme de Delaunay qui permet d'obtenir une triangulation du domaine. En effet, si deux sites s_i et s_j ont une arrête commune dans le diagramme de Voronoï, alors ces deux sites sont reliés

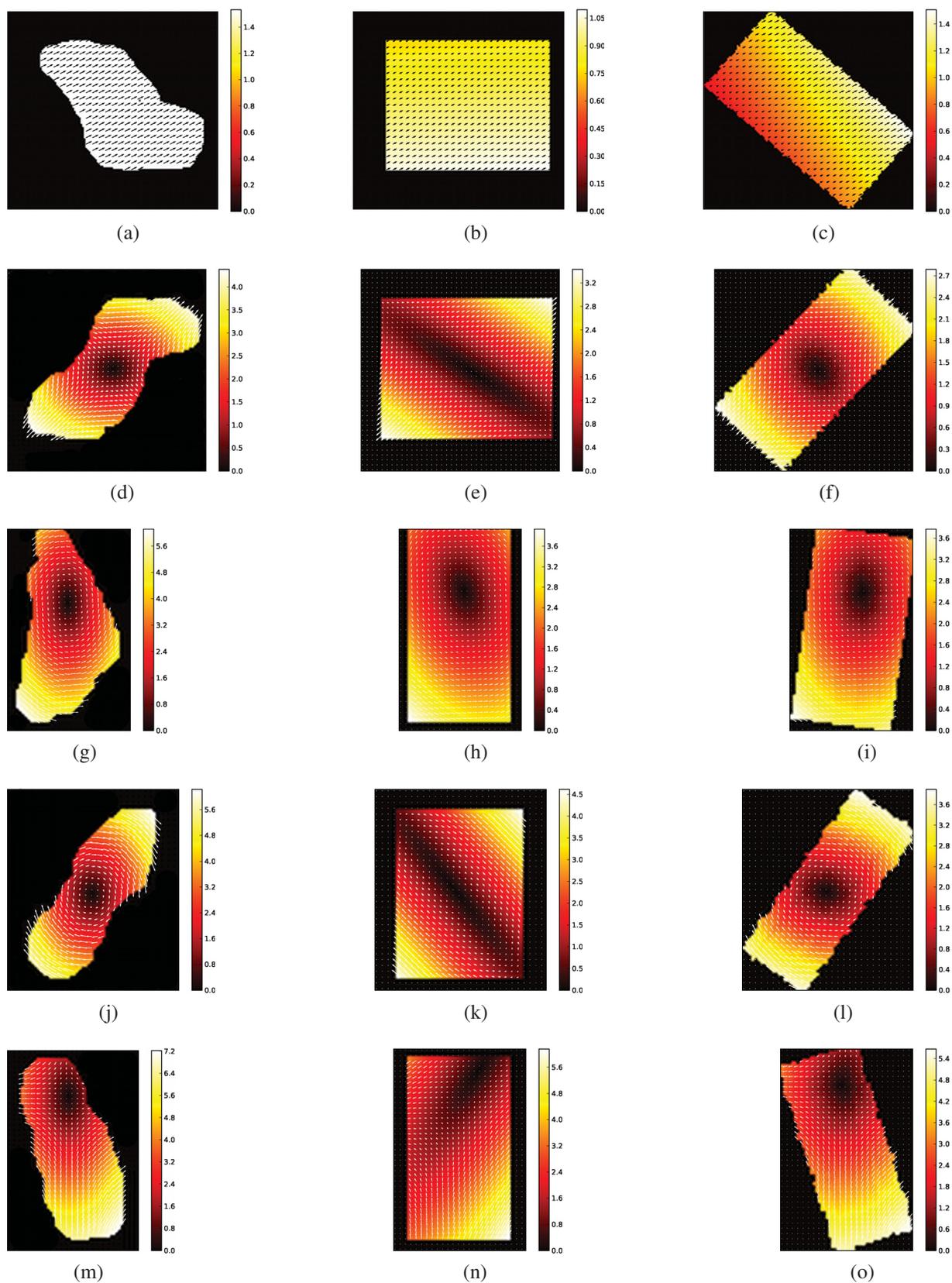


FIGURE 4.16 – Différents mouvements composant le champ synthétique présenté FIGURE.4.14. (a), (d), (g), (j) et (m) sont les champs originaux ; (b), (e), (h), (k) et (n) leur approximation respective avec une boîte englobante (degré 1) ; (c), (f), (i), (l) et (o) leur approximation respective avec une boîte englobante orientée(degré 1).

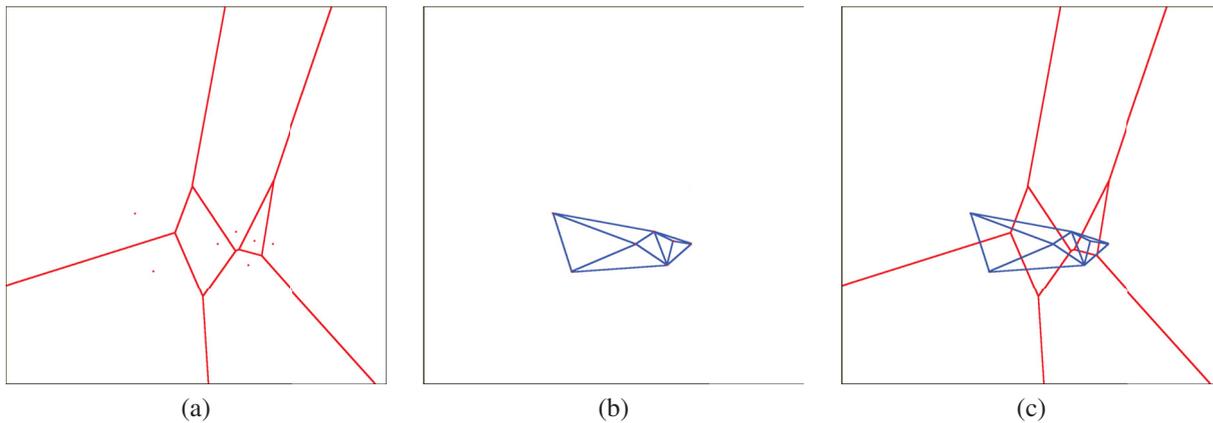


FIGURE 4.17 – Illustration du diagramme de Voronoï et son dual la triangulation de Delaunay ; (a) Diagramme de Voronoï (en rouge) ; (b) Triangulation de Delaunay (en bleu) ; (c) Triangulation de Delaunay (en bleu) superposée au diagramme de Voronoï (en rouge).

par une arête dans le diagramme de Delaunay (cf. figure FIG 4.17). On peut aussi extraire un graphe d'adjacence des régions à partir du diagramme de Voronoï.

Dans notre cas, les sites sont définis par les points singuliers d'un champ de déplacements. Pour le champ 131 de la séquence Cylindre-Carré, que nous avons utilisé pour nos tests, la partition obtenue est présentée en figure FIG 4.18.



FIGURE 4.18 – Partition obtenue par le diagramme de Voronoï à partir des points singuliers du champ 131 extrait de la séquence Cylindre-Carré ; (a) avec une distance $L1$; (b) avec une distance $L2$.

Chaque région de cette partition doit être approximée par projection sur une base de polynômes. Les bases polynomiales sont définies sur un domaine rectangulaire. Chaque région du diagramme de Voronoï est donc placée dans une boîte englobante.

L'objectif de l'approximation fondée sur les points singuliers est de pouvoir obtenir une approximation de qualité des régions proches de ces points. On peut constater sur la figure FIG 4.18 que le diagramme de Voronoï sépare bien les différentes régions les unes des autres. Cependant, les régions s'étendent souvent jusqu'aux bords du domaine spatial du champ. L'approximation n'est donc pas de très bonne qualité autour de la singularité. Nous proposons donc d'ajouter des sites de manière à ce que la partition obtenue par un diagramme de Voronoï fournisse des zones peu étendues autour des sites d'origine.

Nous supposons que chaque site est entouré par une boîte englobante à l'intérieur de laquelle le mouvement entourant le site peut être modélisé dans une base de degré donné. Les dimensions de cette boîte sont associées à la position du site. Si l'on se place dans le cas où il n'y a qu'un site donc une seule boîte dans le champ, il faut ajouter 1 point dans chacune des directions cardinales (Nord, Sud, Est et Ouest) à une distance égale au double du côté de la boîte englobante. Si on se place dans le cas général, c'est-à-dire qu'il peut y avoir plusieurs boîtes englobantes, il ne faut pas qu'un site ajouté pour partitionner une boîte ait de l'influence sur une autre zone. Pour chaque site, nous définirons une zone dans laquelle il est impossible d'ajouter des sites. Ces zones sont appelées zones de réserve et sont délimitées par une boîte englobante aux dimensions doubles de la zone d'intérêt. La figure FIG 4.19a présente les zones de réserve associées au

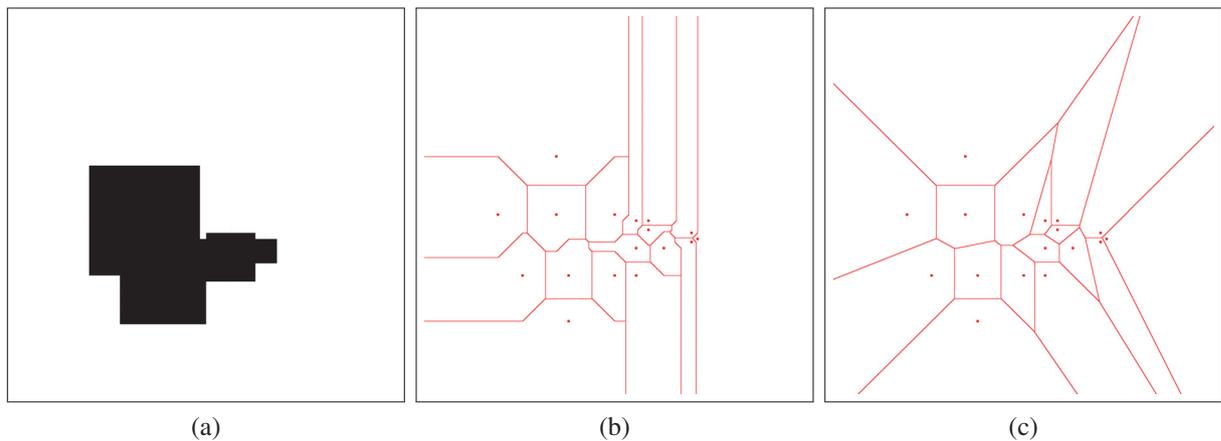


FIGURE 4.19 – Illustration de l'ajout de site au diagramme de Voronoï ; (a) Zones de réserve associées au diagramme de Voronoï présenté FIGURE.4.18 ; (b) Diagramme de Voronoï après l'ajout de nouveaux sites.

diagramme de Voronoï présenté figure FIG 4.18.

L'algorithme pour ajouter des sites est le suivant :

1. Définir les zones de réserve pour tous les sites ;
2. Pour chaque site, ajouter des nouveaux sites sur les directions cardinales à une distance égale au double du côté de la boîte englobante associée à ce site si le nouveau site n'est pas dans une zone de réserve ;
3. Recalculer le diagramme de Voronoï.

À partir de cette partition, nous proposons une approximation par base de polynômes. Chaque partition est associée à une boîte englobante orientée ou non. Le champ à l'intérieur de cette boîte englobante est ensuite projeté sur la base de polynômes.

La figure FIG 4.20 permet de comparer l'approximation à partir d'une partition de Voronoï avec une distance $L1$ et avec une distance $L2$. Cette figure présente les cartes d'erreurs quadratiques obtenues pour différents degrés avec chacune des distances. La distance $L1$ permet d'obtenir une partition qui correspond mieux à la forme rectangulaire des boîtes englobantes. L'approximation est donc meilleure pour les diagrammes obtenus avec la distance $L1$.

Afin d'améliorer les résultats obtenus avec la distance $L2$, nous utilisons les boîtes orientées. La figure FIG 4.21 présente la carte d'erreurs d'approximations obtenue avec une approximation de chaque région avec une boîte non orientée en FIG 4.21(a) , (c) et (e) et avec une boîte orientée

en FIG 4.21(b), (d) et (f). Les résultats sont présentés pour des bases de polynômes de degré 1, 5 et 9. On peut constater que l'approximation globale est meilleure avec les boîtes orientées, bien que pour les degrés 5 et 9 l'erreur maximale est plus importante qu'avec les boîtes non-orientées. Quelque soit le type de boîtes utilisées, l'approximation avec une distance $L2$ est moins bonne qu'avec la distance $L1$.

4.7 Conclusion

Le tableau TAB 4.2 rappelle les mesures d'erreur du tableau TAB 4.1 auquel nous avons ajouté les résultats de l'approximation par boîtes hiérarchiques, et par subdivision issue du diagramme de Voronoï par boîtes englobantes non orientées et par boîtes englobantes orientées. Tous les résultats sont présentés pour le même champ 131 de la séquence *Cylindre-Carré* (cf. ANNEXE A). Nous pouvons constater que les méthodes de subdivision telles que les grilles régulières, le quadtree ou le kdtree permettent d'améliorer l'approximation tout en conservant des bases de degré faible.

La technique d'approximation par combinaison de la modélisation globale et des boîtes englobantes locales ne permet pas d'obtenir une bonne qualité d'approximation mais reste intéressante du point de vue de l'analyse.

L'utilisation du diagramme de Voronoï avec une distance $L1$ permet d'obtenir une qualité d'approximation comparable à une modélisation globale pour un même taux de compression. La subdivision fournie par le diagramme de Voronoï permet d'utiliser des bases de degré plus faible. Cependant, avec la distance $L2$ la subdivision par diagramme de Voronoï produit une erreur quadratique moyenne plus élevée que par modélisation globale, tout en fournissant malgré tout, une erreur angulaire moins élevée. Les boîtes orientées ne présentent pas d'intérêt avec la distance $L1$ puisque l'amélioration obtenue pour la EQM est de l'ordre 10^{-3} . Cette méthode est donc intéressante du point de vue de l'analyse puisque les zones d'intérêt, situées autour des sites de Voronoï, sont bien modélisées ce qui est l'objectif de cette méthode. La distance $L1$ est à privilégier. L'erreur quadratique moyenne obtenue est élevée en raison de la répartition de l'énergie des vecteurs connus sur les vecteurs inconnus lors de la reconstruction. C'est pour cette raison que les résultats obtenus sont meilleurs avec la distance $L1$ qui fournit des régions plus rectangulaires. Pour la distance $L2$, la technique des boîtes englobantes orientées permet d'améliorer sensiblement les résultats.

Les difficultés d'approximation rencontrées avec la partition obtenue par diagramme de Voronoï démontrent l'importance de la densité du champ lors de son approximation par projection sur une base de polynômes. Nous consacrons donc le chapitre suivant à l'approximation des champs épars.

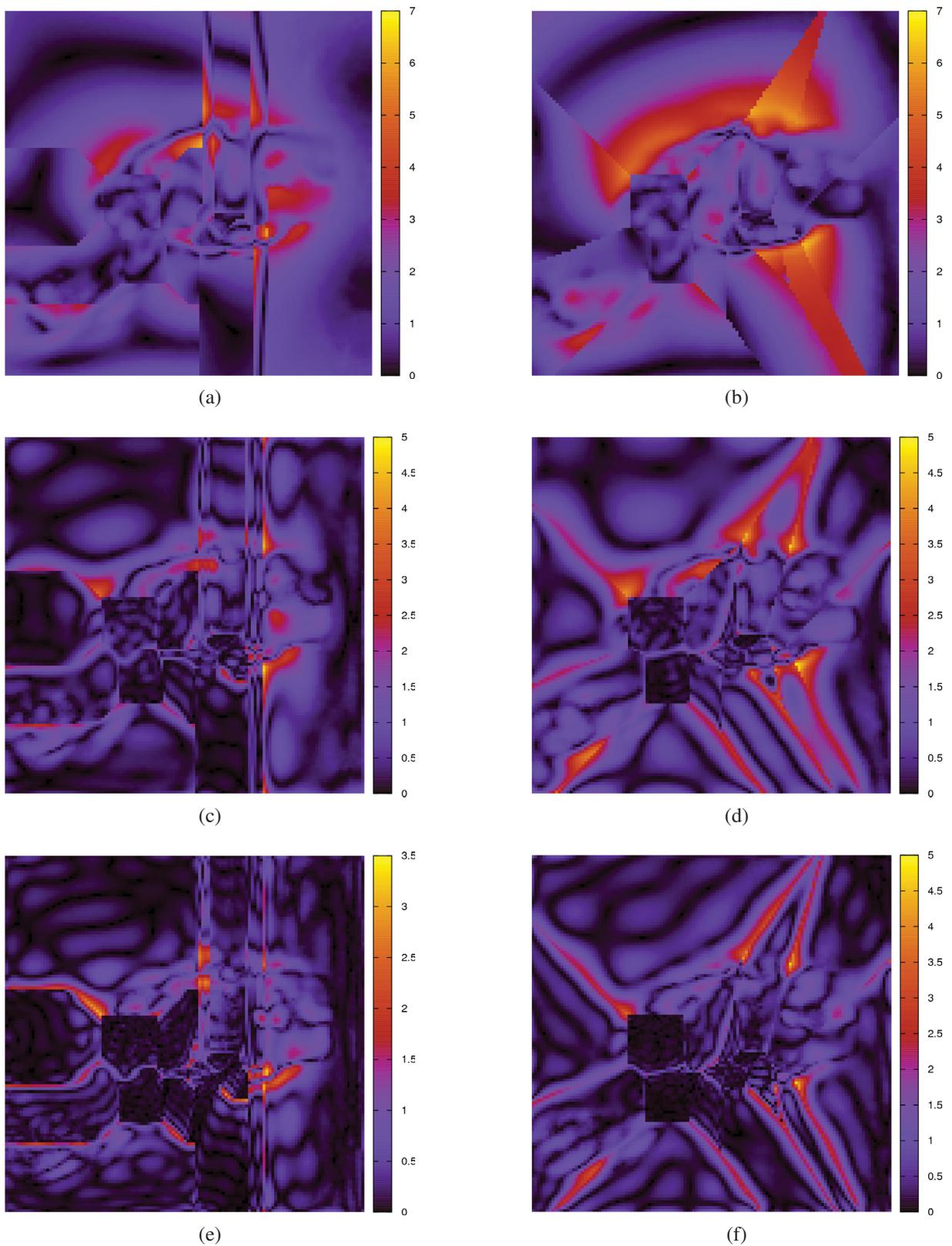


FIGURE 4.20 – Carte d’erreurs quadratiques associée à l’approximation du champ test obtenue à partir du diagramme de Voronoï avec une distance $L1$ et avec une distance $L2$ pour des boîtes englobantes non-orientées (a) distance $L1$ et degré 1 ; (b) distance $L2$ et degré 1 ; (c) distance $L1$ et degré 5 ; (d) distance $L2$ et degré 5 ; (e) distance $L1$ et degré 9 ; (f) distance $L2$ et degré 9 ;

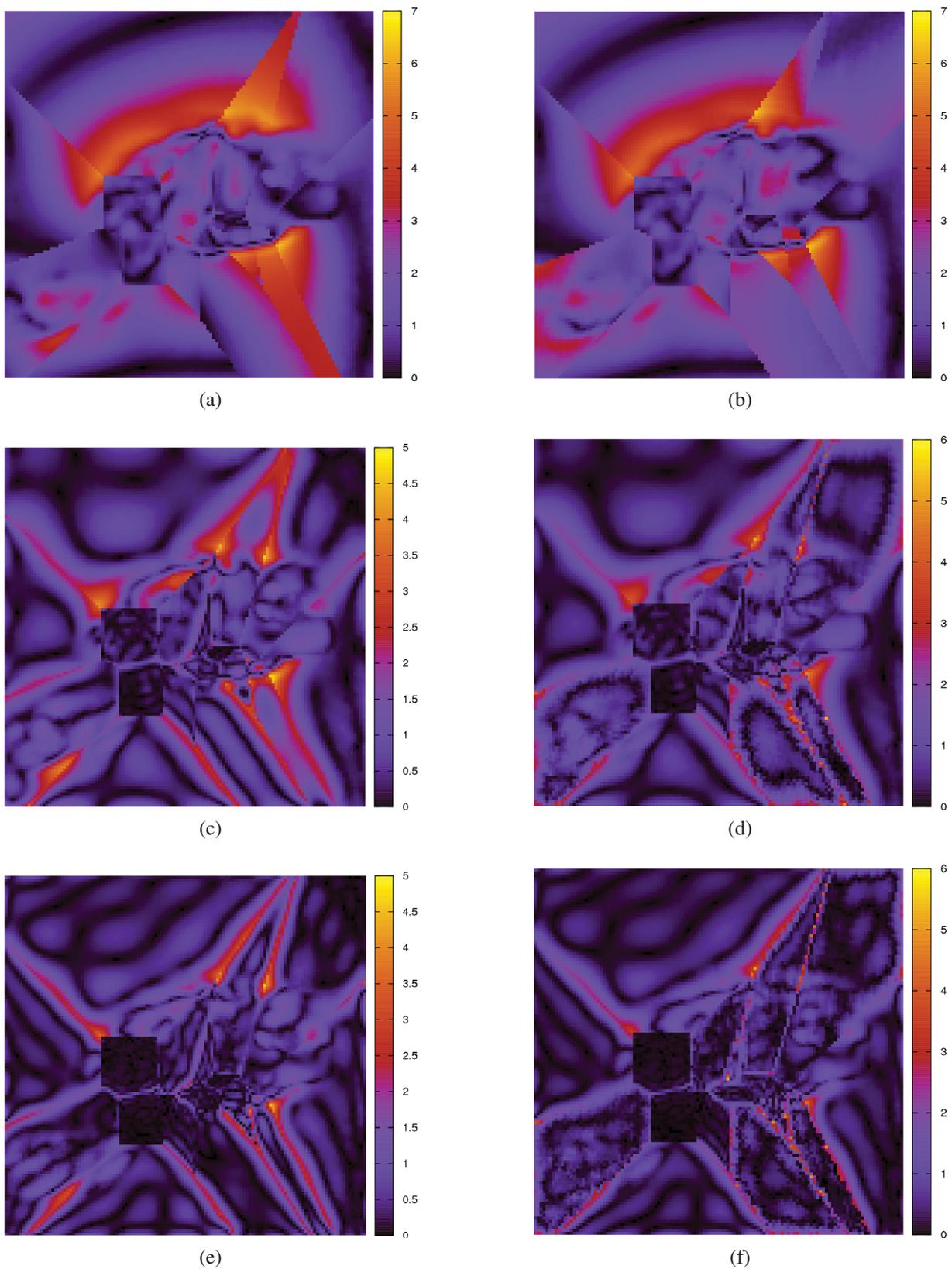


FIGURE 4.21 – Carte d’erreurs quadratiques associée à l’approximation du champ test obtenue à partir du diagramme de Voronoï avec une distance $L2$ pour des boîtes englobantes non-orientées (a), (c) et (e) ou orientées (b), (d) et (f) ; (a) boîtes non orientées (degré 1) ; (b) boîtes orientées (degré 1) ; (c) boîtes non orientées (degré 5) ; (d) boîtes orientées (degré 5) ; (e) boîtes non orientées (degré 9) ; (f) boîtes orientées (degré 9) ;

Méthode	Degré	Niveau	Gain	EQM	E_{Θ}
globale	5	0	99.863	1.167	0.210
globale	9	0	99.642	0.631	0.112
globale	16	0	99.005	0.396	0.080
globale	20	0	98.498	0.308	0.069
grille régulière	9	3	77.107	0.020	0.005
grille régulière	5	4	65.036	0.014	0.004
grille régulière	1	5	80.021	0.041	0.011
quadtree	9	3	87.838	0.034	0.007
quadtree	5	4	87.981	0.036	0.008
quadtree	1	5	92.254	0.072	0.015
kd-tree median	9	3	91.058	0.040	0.008
kd-tree median	5	4	91.669	0.045	0.009
kd-tree median	1	5	92.683	0.081	0.015
kd-tree irrégulier	9	3	92.488	0.053	0.009
kd-tree irrégulier	5	4	92.215	0.050	0.009
kd-tree irrégulier	1	5	96.449	0.135	0.032
boîte (global,boîte)	(16,1)	0	98.868	0.389	0.077
boîte (global,boîte)	(9,5)	0	98.803	0.532	0.079
Voronoi $L1$ (boîtes non orientées)	9	14 sites	94.992	0.305	0.043
Voronoi $L1$ (boîtes non orientées)	5	14 sites	98.088	0.517	0.065
Voronoi $L1$ (boîtes non orientées)	1	14 sites	99.727	1.170	0.150
Voronoi $L2$ (boîtes non orientées)	9	14 sites	94.992	0.562	0.044
Voronoi $L2$ (boîtes non orientées)	5	14 sites	98.088	0.824	0.071
Voronoi $L2$ (boîtes non orientées)	1	14 sites	99.727	1.691	0.149
Voronoi $L2$ (boîtes orientées)	9	14 sites	94.992	0.585	0.048
Voronoi $L2$ (boîtes orientées)	5	14 sites	98.088	0.821	0.076
Voronoi $L2$ (boîtes orientées)	3	14 sites	99.090	1.076	0.106
Voronoi $L2$ (boîtes orientées)	1	14 sites	99.727	1.609	0.155

TABLEAU 4.2 – Evaluation de l’erreur de modélisation et du ration de compression pour les méthodes globale, de subdivision par grille régulière, par quadtree, par kd-tree, par boîtes englobantes et par diagramme de Voronoï

CHAPITRE 5

APPROXIMATION HIÉRARCHISÉE DES CHAMPS ÉPARS

5.1 Difficultés d'approximation des champs épars

Selon la méthode utilisée pour estimer le mouvement ou la nature même du mouvement, le champ de déplacements peut être composé de vecteurs nuls. Les méthodes telles que le tenseur de structure [Augereau 05], par exemple, ne fournissent pas des champs de vecteurs denses.

Lors de l'approximation (projection et reconstruction) d'un champ non dense avec une base de polynômes, les vecteurs ont la même importance qu'ils soient nuls ou non. Lors de la reconstruction, l'énergie des vecteurs non nuls est répartie sur l'ensemble du domaine de définition du champ. Nous illustrons ce problème avec la figure FIG 5.1. En FIG 5.1(a), nous présentons un champ de déplacements épars. Ce champ ne comporte que 45 vecteurs non-nuls sur un domaine de 32×32 vecteurs. Ce champ a été obtenu à partir d'une rotation affine définie sur le domaine de définition complet dont une partie des vecteurs a été annulée par la suite. En figure FIG 5.1(b), nous présentons l'approximation du champ FIG 5.1(a) par projection sur une base de polynômes de degré 1. On peut constater sur cet exemple que le mouvement reconstruit représente bien une rotation mais la norme est très faible sur l'ensemble du domaine de définition. En fait, l'énergie des 45 vecteurs non nuls a été répartie sur le domaine de définition complet.

L'objet de ce chapitre est de proposer des solutions permettant de modéliser un champ de déplacements épars avec une approximation de qualité sans utiliser une base de degré élevé. Après avoir vérifié que les solutions étudiées dans les chapitres précédents ne permettent pas de résoudre ce problème, nous étudions, dans la section suivante, la génération des bases polynomiales définies sur un support épars. À partir de cette nouvelle méthode de génération des bases polynomiales, nous pouvons à nouveau utiliser les différentes méthodes de subdivisions étudiées au chapitre précédent.

Nous illustrons sur la figure FIG 5.2 que l'augmentation du degré de la base ne suffit pas pour régler ce problème. Nous présentons, en FIG 5.2(b), une approximation du champ FIG

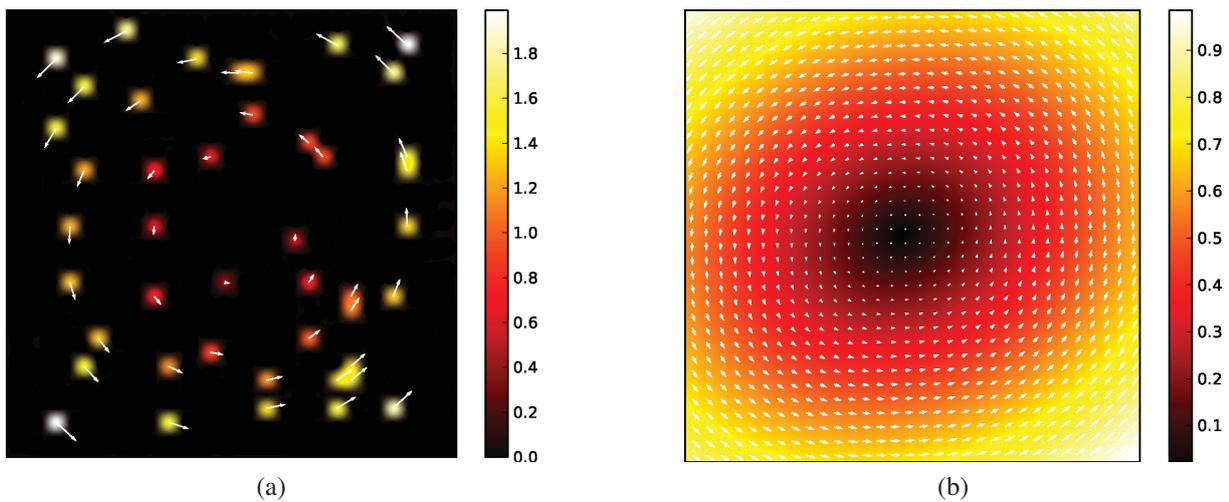


FIGURE 5.1 – Illustration de l’approximation globale d’une rotation affine éparse avec une base de polynômes dense de degré 1 ; (a) Exemple de rotation affine éparse ; (b) Résultat de l’approximation globale du champ (a) avec une base de degré 1 (l’amplitude des vecteurs est multipliée par 10) ;

5.2(a) par projection sur une base de polynômes de degré 16. Le résultat n’est pas satisfaisant, nous nous tournons donc vers l’étude d’autres méthodes.

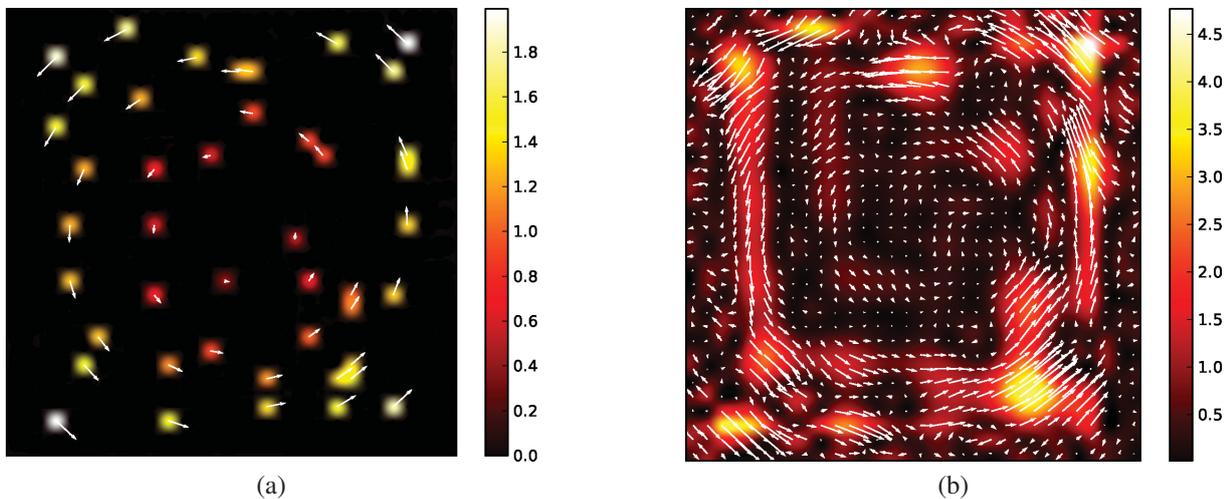


FIGURE 5.2 – Illustration de l’approximation globale d’une rotation affine éparse avec une base de polynômes dense de degré 16 ; (a) Exemple de rotation affine éparse ; (b) Approximation globale d’une rotation affine éparse avec une base de degré 16 (l’amplitude des vecteurs est multipliée par 10).

Nous avons étudié au chapitre précédent différentes méthodes de subdivision qui ont permis d’approximer avec une qualité acceptable des champs complexes sans utiliser des bases de degré élevé. Nous étudions donc la méthode de subdivision par quadtree afin de vérifier si elle permet de résoudre le problème de l’approximation des champs denses.

Il n'est pas intéressant de présenter cette méthode sur un champ de petite dimension comme le champ de rotation affine défini sur 32×32 vecteurs. Nous définissons donc 2 champs de déplacements épars à partir du champ 131 de la séquence cylindre-carré utilisé pour les tests du chapitre précédent (cf. figure FIG 5.3(a)).

Nous appelons le premier champ test *Cylindre-carré épars 1*. Il est construit à partir du champ cylindre-carré auquel des vecteurs ont été annulés sur une seule zone circulaire de taille importante. Ce champ est présenté en figure FIG 5.3(b).

Nous appelons le deuxième champ test *Cylindre-carré épars 2*. Il est construit à partir du champ cylindre-carré auquel des vecteurs ont été annulés sur plusieurs zones de petite taille réparties aléatoirement suivant une loi uniforme sur l'ensemble du domaine de définition. Ce champ est présenté en figure FIG 5.3(c).

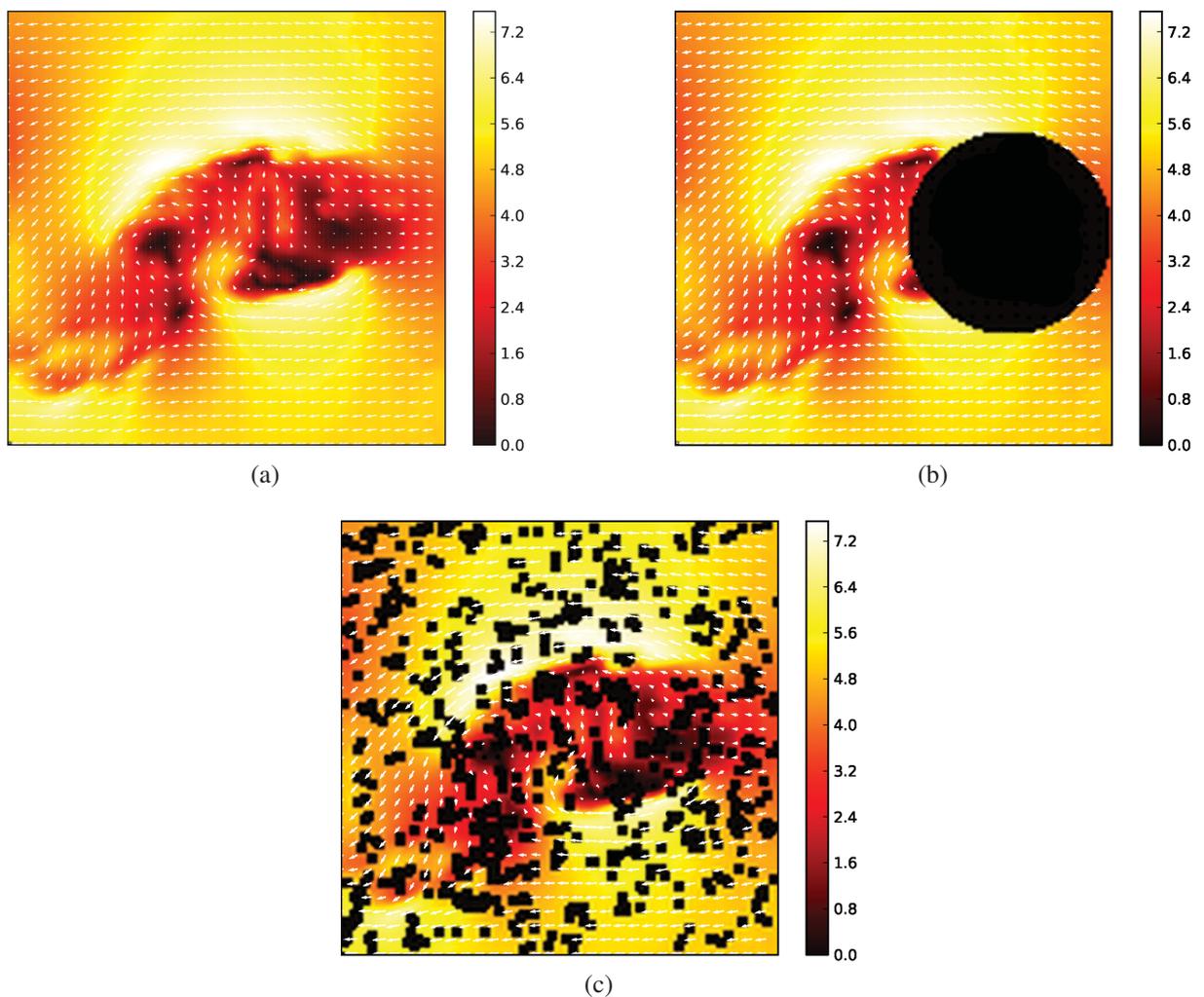


FIGURE 5.3 – Champs tests épars pour les tests de subdivisions par quadtree ; (a) Champ 131 (dense) de la séquence Cylindre-Carré ; (b) Champ *Cylindre-carré épars 1* construit à partir du Cylindre-Carré dense auquel des vecteurs ont été annulés sur une seule zone circulaire de taille importante ; (c) Champ *Cylindre-carré épars 2* construit à partir du Cylindre-Carré dense auquel des vecteurs ont été annulés sur plusieurs zones de petite taille réparties aléatoirement suivant une loi uniforme sur l'ensemble du domaine de définition.

Ces 2 champs sont approximés avec un quadtree. L'algorithme est présenté en section 4.3. Les résultats d'approximation obtenus avec un quadtree sur 5 niveaux et une base de degré 1 sont présentés en figure FIG 5.4(a) pour le champ *Cylindre-carré épars 1* et figure FIG 5.4(b) pour le champ *Cylindre-carré épars 2*. On peut constater que le quadtree permet d'obtenir une bonne approximation du premier champ en limitant le nombre de vecteurs nuls à modéliser. Cependant, les vecteurs du bord de la zone nulle sont entachés d'une erreur d'approximation importante. L'approximation du deuxième champ n'est satisfaisante ni du point de vue de la qualité d'approximation puisque l'erreur locale reste importante sur l'ensemble du domaine de définition, ni du point de vue de l'analyse car la subdivision obtenue correspond presque à une grille régulière.

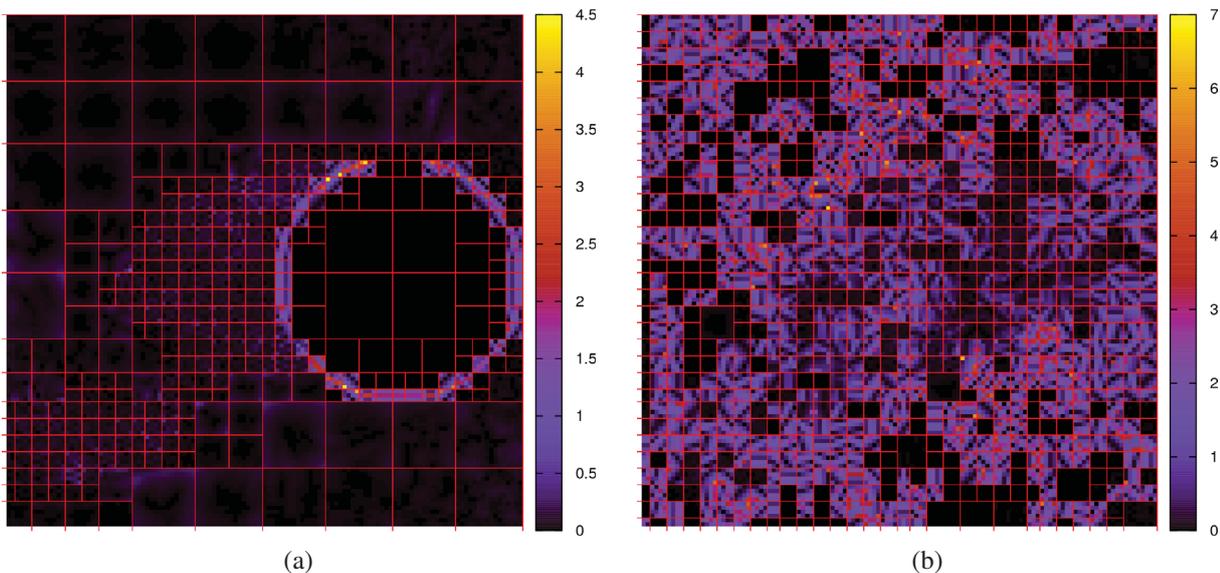


FIGURE 5.4 – Approximation de deux champs épars avec un quadtree de degré 1 sur 5 niveaux ; (a) Approximation du champ *Cylindre-carré épars 1* ; (b) Approximation du champ *Cylindre-carré épars 2*

la gestion du domaine de définition. Ainsi, nous ne prenons en compte que les vecteurs pour lesquels le mouvement est défini. Nous verrons que cette solution ouvre des perspectives intéressantes, comme l'identification du mouvement dans les champs épars. Enfin, nous intégrons ces nouvelles bases dans les méthodes d'approximation par subdivision du domaine étudiées au chapitre précédent. Nous revenons sur la génération des bases de polynômes bi-variables en prenant en compte la densité du champ lors de la génération de la base.

5.2 Bases de polynômes orthogonaux définies sur un support épars

Comme nous avons pu constater dans la section précédente, la modélisation d'un champ épars n'est pas simple. Même en utilisant les techniques de subdivision, le résultat de la modélisation ne permet pas une approximation précise d'un champ de déplacements.

Dans cette section, nous revenons sur la génération des bases de polynômes bi-variables. Nous avons vu au chapitre 3 comment générer une base de polynômes par la procédure dite de

la récurrence à trois termes. Nous avons au préalable défini un produit scalaire dans lequel nous faisons apparaître une fonction de pondération $w(x_1, x_2)$. Cette fonction de pondération permet de définir le type de base générée. Ici nous assignons à cette fonction une notion d'appartenance au sous domaine Ω^* sur lequel notre champ est défini. Ce nouveau domaine de définition w^* est défini comme suit :

$$w^*(x_1, x_2) = \delta_{\Omega^*} w(x_1, x_2) \quad (5.1)$$

avec

$$\delta_{\Omega^*} = \begin{cases} 1 & \text{si } x \in \Omega \\ 0 & \text{si } x \notin \Omega \end{cases} \quad (5.2)$$

Avec une telle fonction de pondération, nous ne pouvons pas garantir que la base B^* produite par la procédure de récurrence à trois termes soit orthogonale. C'est pourquoi nous orthogonalisons la base obtenue en utilisant la procédure d'orthogonalisation de GRAMM-SCHMIDT. Pour cela, nous devons définir un ordre sur les indices des polynômes. Cet ordre est défini par la relation suivante :

$$(i, j) \prec (k, l) \text{ si } \begin{cases} i + j < k + l \\ i > k \text{ si } i + j = k + l \end{cases} \quad (5.3)$$

Il est alors possible de définir la procédure de Gramm-Schmidt de façon générique par :

$$\begin{cases} C_{i,j}(x_1, x_2) = P_{i,j}^*(x_1, x_2) - \sum_{(pq) \prec (i,j)} \langle P_{i,j}^* | P_{p,q} \rangle_{w^*} P_{p,q}(x_1, x_2) \\ P_{i,j}(x_1, x_2) = \frac{C_{i,j}(x_1, x_2)}{\sqrt{\langle C_{i,j}^* | C_{i,j} \rangle_{w^*}}} \end{cases} \quad (5.4)$$

Nous pouvons alors maintenant obtenir une base dont le domaine est défini en fonction de la densité du champ que l'on souhaite modéliser. Dans la suite du document, nous appellerons par abus de langage ce type de base : *base éparse*, alors que les bases présentées jusqu'alors seront appelées : *bases denses*. Nous insistons bien sur le fait que c'est bien le domaine de définition du support de la base qui est *dense* ou *épars*.

Nous proposons en figure FIG 5.5 une approximation du mouvement de rotation affine éparse (ce mouvement est présenté en figure FIG 5.1) avec une base éparse de degré 1. Dans ce cas, l'erreur d'approximation obtenue est nulle. Il est en effet possible d'obtenir une approximation à erreur nulle des champs affines de degré 1 même lorsque le champ est épars.

Nous avons vu qu'il est parfois nécessaire d'utiliser des bases de degré supérieur à 1. Nous vérifions les résultats obtenus sur un champ épars de degré 16. À partir d'un champ polynomial dense de degré 16 dont le domaine spatial est de 124×124 vecteurs, nous créons un champ épars en annulant de manière régulière l'amplitude de deux vecteurs sur trois horizontalement et verticalement. Ce champ, présenté en figure FIG 5.6, est appelé par la suite *Champ polynomial_{1/3}*. Le champ n'est pas redimensionné, son domaine spatial est toujours de 124×124 vecteurs mais certains vecteurs sont d'amplitude nulle. Comme pour les champs affines de degré 1, l'erreur d'approximation par projection sur base polynomiale éparse produit une erreur nulle.

Dans la pratique, les champs ne sont pas toujours polynomiaux. Les champs *Cylindre-carré épars 1* et *Cylindre-carré épars 2* sont des champs non polynomiaux. Les figures FIG 5.7 et FIG 5.8 rappellent ces champs respectivement en (a), leur approximation globale avec une *base dense* de degré 16 en (b) et avec une *base éparse* de même degré (c). Les cartes d'erreur d'approximation obtenues avec la base dense et avec la base éparse sont présentées respectivement en (d) et (e). On constate l'amélioration apportée par les *bases éparses* sur l'approximation de tels champs.

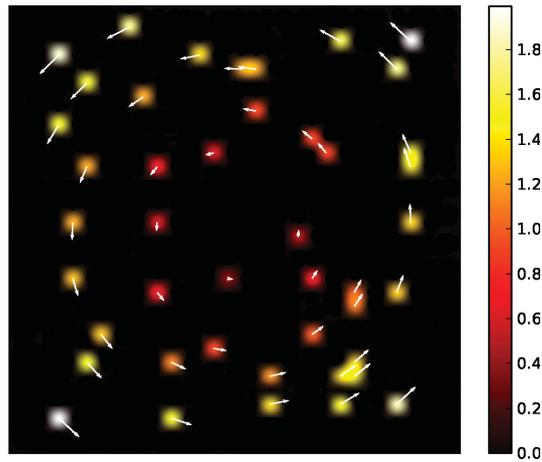
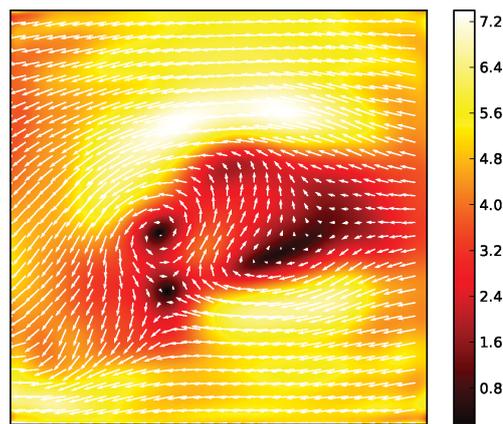
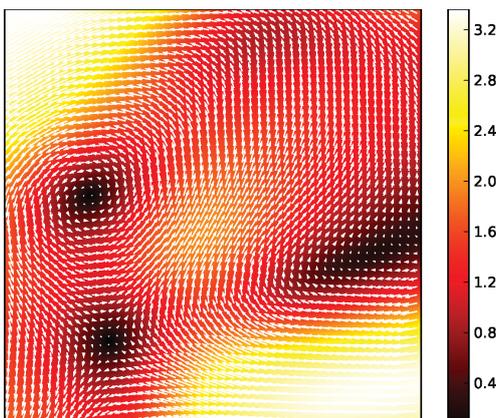


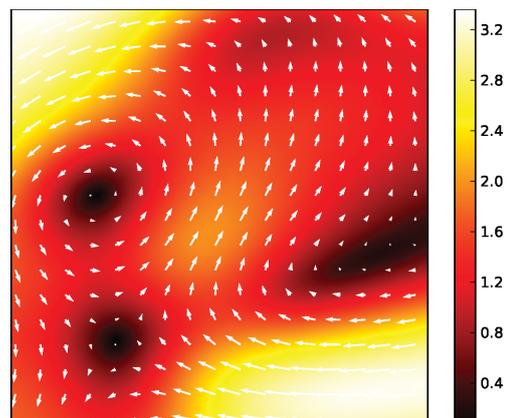
FIGURE 5.5 – Approximation du champ présenté en figure FIG 5.1 avec une base éparse de degré 1



(a)



(b)



(c)

FIGURE 5.6 – *Champ polynomial*_{1/3}; (a) Champ polynomial obtenu par projection du champ 131 de la séquence *Cylindre-Carre* (cf. ANNEXE A) sur une base de degré 16 (124×124 vecteurs); (b) Extrait du champ (a); (c) Extrait du *champ polynomial*_{1/3} obtenu en annulant l'amplitude de deux vecteurs sur trois du champ (a) horizontalement et verticalement.

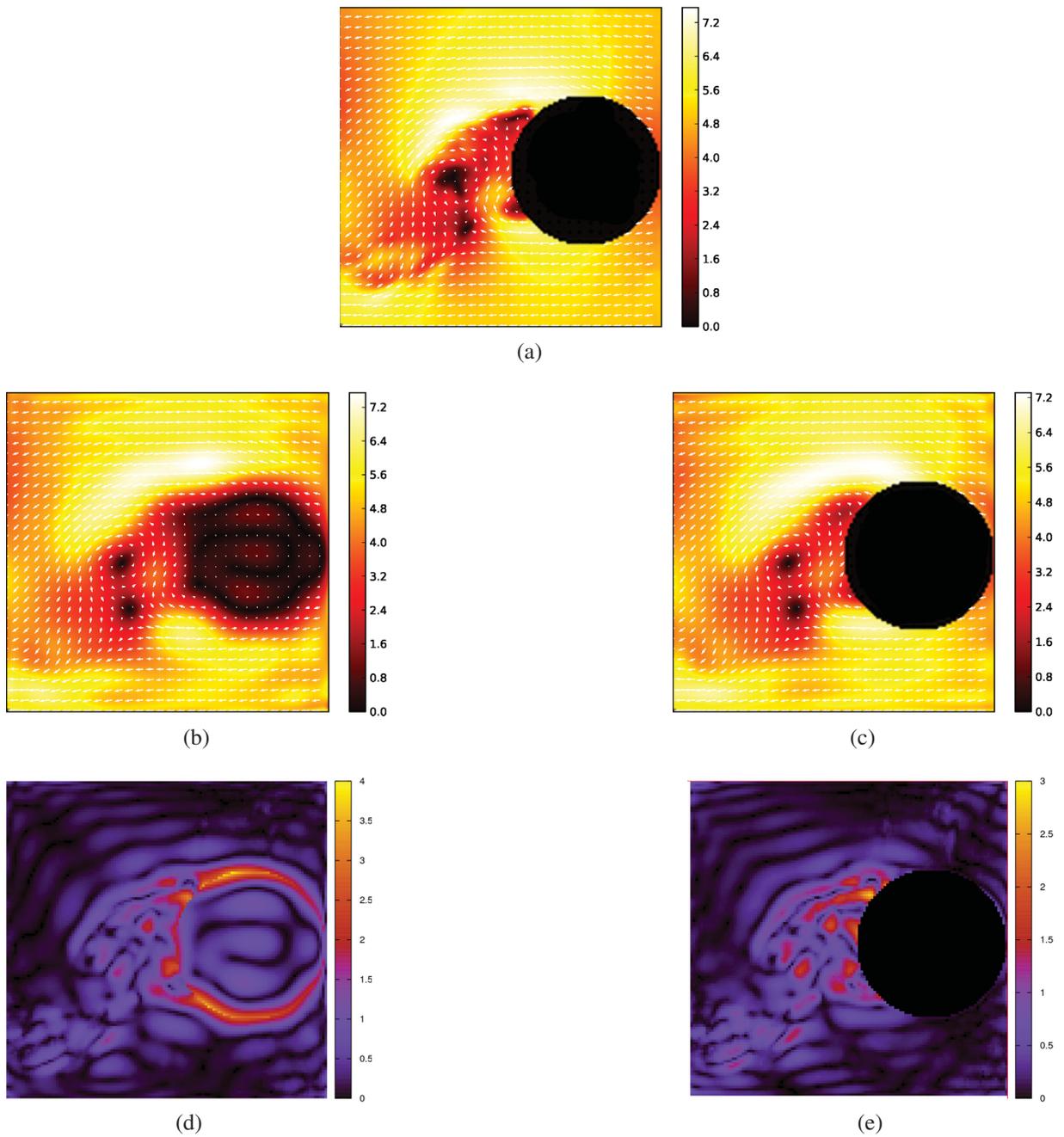


FIGURE 5.7 – Modélisation du mouvement *Cylindre-carré épars* 1 : comparaison entre une base dense et une base non-dense ; (a) Champ *Cylindre-carré épars* 1 ; (b) Modélisation globale du champ (a) avec une base dense de degré 16 ; (c) Modélisation globale du champ (a) avec une base éparse de degré 16 ; (d) Carte d'erreurs quadratiques associée à la modélisation (b) du champ (a) ; (e) Carte d'erreurs quadratiques associée à la modélisation (c) du champ (a).

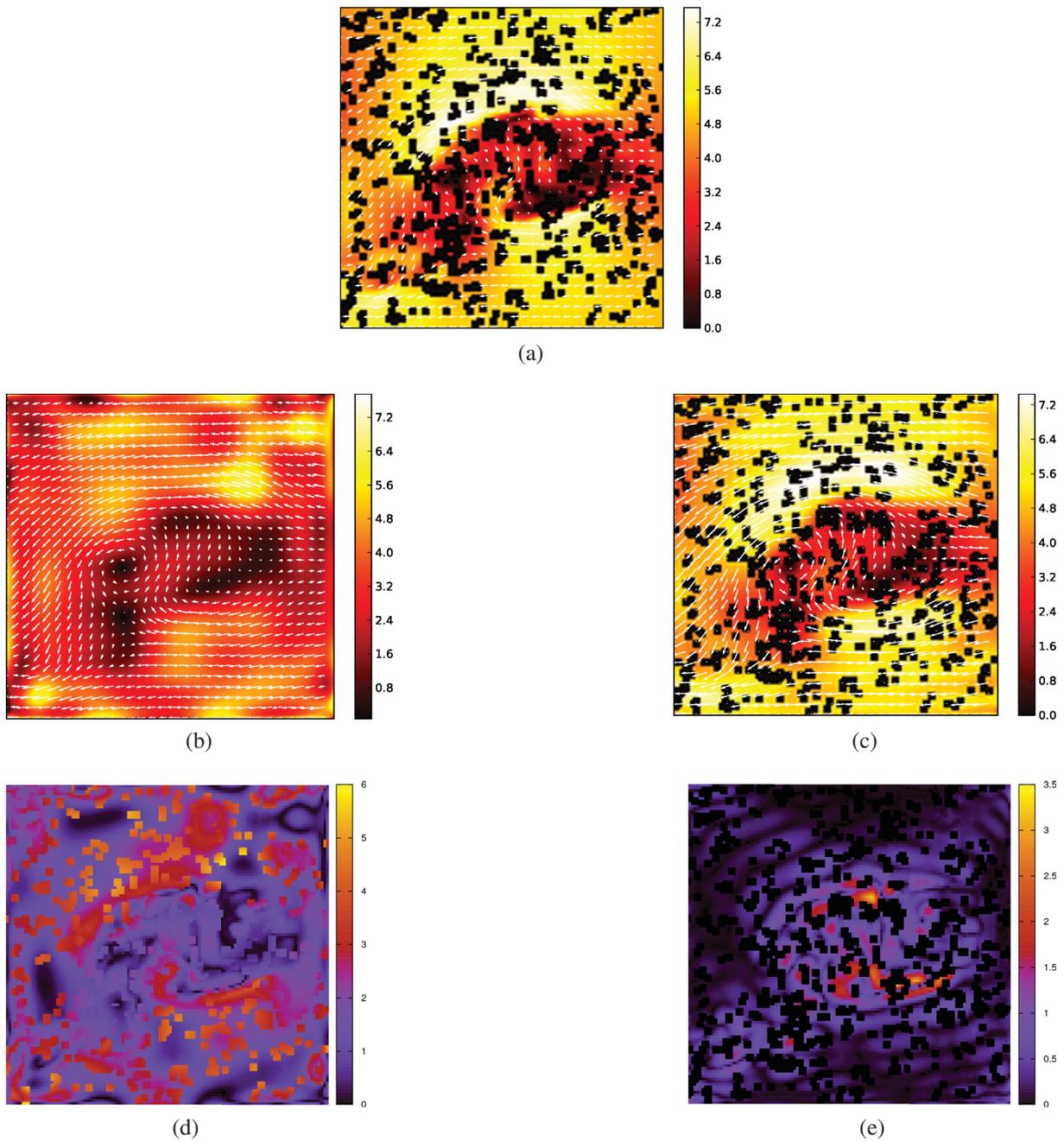


FIGURE 5.8 – Modélisation du mouvement *Cylindre-carré épars 2* : comparaison entre une base dense et une base non-dense ; (a) Champ *Cylindre-carré épars 2* ; (b) Modélisation globale du champ (a) avec une base dense de degré 16 ; (c) Modélisation globale du champ (a) avec une base éparse de degré 16 ; (d) Carte d'erreurs quadratiques associée à la modélisation (b) du champ (a) ; (e) Carte d'erreurs quadratiques associée à la modélisation (c) du champ (a).

5.3 Identification

Les bases éparées permettent d'approximer sans erreur les champs polynomiaux éparés. La capacité d'approximation est un des intérêts des bases de polynômes. Cependant, il est aussi important que les bases polynomiales permettent une analyse des mouvements à partir des coefficients polynomiaux. Cependant, les coefficients de deux bases éparées définies pour deux densités spatiales différentes n'ont pas la même signification. Nous illustrons cela par l'exemple suivant. À partir du champ de déplacements d'une rotation affine dense, présenté sur la figure FIG 5.9(a), nous définissons deux rotations affines de densités spatiales différentes, présentées sur les figures FIG 5.9(b) et FIG 5.9(c). Nous générons alors deux bases de polynômes éparées, puis chaque champ est projeté sur la base de densité spatiale correspondante. Les coefficients obtenus par projection du champ FIG 5.9(b) sont :

$$\begin{aligned} P_U &= 5.651 \times 10^{-2} \times P_{0,0} - 6.458 \times P_{0,1} + 2.253 \times 10^{-6} \times P_{1,0} \\ P_V &= 4.60 \times 10^{-1} \times P_{0,0} + 0.413 \times P_{0,1} + 5.947 \times P_{1,0} \end{aligned} \quad (5.5)$$

Les coefficients obtenus par projection du champ FIG 5.9(c) sont :

$$\begin{aligned} P_U &= -5.330 \times 10^{-1} \times P_{0,0} - 5.180 \times P_{0,1} + 1.868 \times 10^{-6} \times P_{1,0} \\ P_V &= -5.933 \times 10^{-1} \times P_{0,0} + 0.625 \times P_{0,1} + 4.930 \times P_{1,0} \end{aligned} \quad (5.6)$$

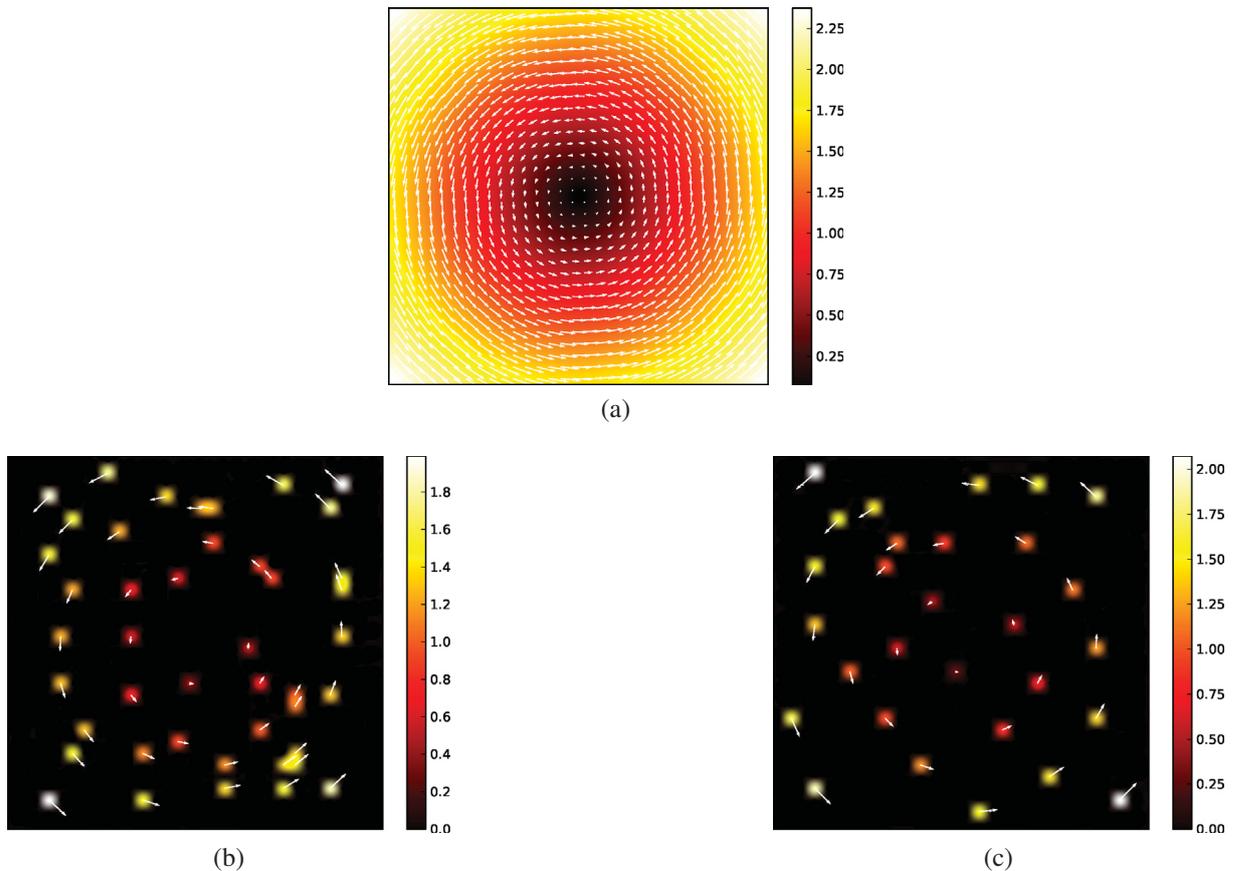


FIGURE 5.9 – Identification de deux rotations affines éparées extraites d'une rotation affine dense ; (a) rotation affine dense ; (b) premier exemple de rotation affine éparse ; (c) deuxième exemple de rotation affine éparse.

On constate que le même mouvement est modélisé différemment en fonction de la densité du champ. Cela peut donc rendre plus compliqué l'analyse. En effet, la base est orthogonalisée précisément en fonction de la densité spatiale. Pour deux densités différentes, on obtient donc deux bases de polynômes différentes. Nous aboutissons donc à des bases définies *a posteriori* comme pour une décomposition en valeurs singulières. Le choix de notre modèle polynomial s'est orienté vers les bases de polynômes en partie pour éviter ce problème.

En fait, il est possible après projection sur une base de polynômes épars de retrouver la valeur des coefficients dans toute autre base à partir du moment où cette autre base est possible à générer. Voici la procédure permettant de trouver la valeur des coefficients dans la base canonique à partir des coefficients connus dans une base B^* . Le raisonnement suivant étant similaire pour \mathcal{U} et \mathcal{V} , nous ne le présentons que pour \mathcal{U} .

La base canonique est la base constituée des polynômes $P_{i,j}^c(x) = x_1^i x_2^j$. Cette base n'est pas orthogonale sauf pour les degrés 0 et 1. Dans un premier temps, nous supposons que le degré D de la base polynomiale est égal au degré de la composante polynomiale, ici \mathcal{U} . D'après l'équation (3.17), on peut écrire :

$$\tilde{u}_{i,j} \langle P_{i,j} | P_{i,j} \rangle = \langle \mathcal{U} | P_{i,j} \rangle \quad (5.7)$$

Or en supposant que \mathcal{U} soit de forme polynomiale, $\langle \mathcal{U} | P_{i,j} \rangle$ peut s'écrire :

$$\langle \mathcal{U} | P_{i,j} \rangle = \sum_{(k,l) \in [0;D]^2}^{k+l \leq D} \gamma_{k,l} \langle x_1^k x_2^l | P_{i,j} \rangle \quad (5.8)$$

On peut écrire l'égalité matricielle suivante :

$$\begin{pmatrix} \langle 1 | P_{0,0} \rangle & \dots & \langle x_1^k x_2^l | P_{0,0} \rangle & \dots & \langle x_2^d | P_{0,0} \rangle \\ \vdots & & \vdots & & \vdots \\ \langle 1 | P_{i,j} \rangle & \dots & \langle x_1^k x_2^l | P_{i,j} \rangle & \dots & \langle x_2^d | P_{i,j} \rangle \\ \vdots & & \vdots & & \vdots \\ \langle 1 | P_{0,d} \rangle & \dots & \langle x_1^k x_2^l | P_{0,d} \rangle & \dots & \langle x_2^d | P_{0,d} \rangle \end{pmatrix} \begin{pmatrix} \gamma_{0,0} \\ \vdots \\ \gamma_{k,l} \\ \vdots \\ \gamma_{0,d} \end{pmatrix} = \begin{pmatrix} \langle \mathcal{U} | P_{0,0} \rangle \\ \vdots \\ \langle \mathcal{U} | P_{i,j} \rangle \\ \vdots \\ \langle \mathcal{U} | P_{0,d} \rangle \end{pmatrix} \quad (5.9)$$

Nous réécrivons cette équation sous la forme suivante :

$$X_{\mathbb{B}} \Gamma = P_{\mathbb{B}} \quad (5.10)$$

À partir de là, il est possible de retrouver les coefficients polynomiaux dans la base canonique :

$$\Gamma = X_{\mathbb{B}}^{-1} P_{\mathbb{B}} \quad (5.11)$$

Cependant, il est nécessaire de se poser la question de l'inversibilité de la matrice $X_{\mathbb{B}}$. Pour cela, nous nous intéressons à l'expression des produits scalaires $\langle x_1^k x_2^l | P_{i,j} \rangle$. Le terme $x_1^k x_2^l$ étant lui-même un polynôme bi-variable, il peut s'écrire comme une combinaison linéaire des éléments de la base :

$$x_1^k x_2^l = \sum_{(p,q) \in [0;D]^2}^{p+q \leq D} \alpha_{p,q}^{k,l} P_{p,q}(x_1, x_2) = P_{k,l}(x_1, x_2) + \sum_{(p,q) \in [0;D]^2}^{p+q < k+l} \alpha_{p,q}^{k,l} P_{p,q}(x_1, x_2) \quad (5.12)$$

On peut en déduire que

$$\langle x_1^k x_2^l | P_{i,j} \rangle = \langle P_{k,l} | P_{i,j} \rangle + \sum_{(p,q) \in [0;D]^2, p+q < k+l} \alpha_{p,q}^{k,l} \langle P_{p,q} | P_{i,j} \rangle \quad (5.13)$$

Étudions les différents cas possibles. Si $k + l < i + j$, alors par la propriété d'orthogonalité, tous les produits scalaires $\langle P_{k,l} | P_{i,j} \rangle$ et $\langle P_{p,q} | P_{i,j} \rangle$ vont être nuls puisque aucun couple (i, j) ou (p, q) n'est égal au couple i, j . On obtient donc la relation

$$\forall(k, l), \forall(i, j) \quad k + l < i + j \text{ alors } \langle x_1^k x_2^l | P_{i,j} \rangle = 0 \quad (5.14)$$

Si $k + l = i + j$ avec $(k, l) \neq (i, j)$, alors pour les mêmes raisons que dans le cas précédent on obtient

$$\forall(k, l), \forall(i, j) \quad k + l = i + j \text{ avec } (k, l) \neq (i, j) \text{ alors } \langle x_1^k x_2^l | P_{i,j} \rangle = 0 \quad (5.15)$$

Si $(k, l) = (i, j)$, alors seul le produit scalaire $\langle P_{k,l} | P_{i,j} \rangle$ sera non nul, on obtient

$$\forall(k, l), \forall(i, j) \quad (k, l) = (i, j) \text{ alors } \langle x_1^k x_2^l | P_{i,j} \rangle = \langle P_{i,j} | P_{i,j} \rangle \quad (5.16)$$

Enfin, si $k + l > i + j$ alors on peut affirmer que $\langle P_{k,l} | P_{i,j} \rangle$ sera nul. De plus, la seule possibilité pour que $\langle P_{p,q} | P_{i,j} \rangle$ soit non nul est que (p, q) soit égal à (i, j) d'où la relation

$$\forall(k, l), \forall(i, j) \quad k + l > i + j \text{ alors } \langle x_1^k x_2^l | P_{i,j} \rangle = \alpha_{i,j}^{k,l} \langle P_{i,j} | P_{i,j} \rangle \quad (5.17)$$

Ces différentes observations et l'ordonnancement choisi, nous permettent de conclure que la matrice $X_{\mathbb{B}}$ est triangulaire supérieure et que les éléments de sa diagonale sont tous non nuls. Dans ce cas, la matrice $X_{\mathbb{B}}$ est donc inversible.

À partir de l'équation (5.11), il est donc possible de trouver les coefficients polynomiaux dans la base canonique dense à partir des coefficients d'une base éparse. Les coefficients ainsi obtenus ne sont plus dépendants du domaine de définition de la base et il est alors possible de comparer des champs ayant des supports différents. Nous revenons sur l'exemple présenté sur la figure FIG 5.9, où nous avons créé deux rotations éparse à partir d'une seule rotation dense. Après passage dans la base canonique des coefficients présentés en équation (5.5) et (5.6), nous obtenons les coefficients polynomiaux suivants dans les deux cas.

$$\begin{aligned} P_{\mathcal{U}} &= -3.375 \times 10^{-8} \times P_{0,0} - 1.679 \times P_{0,1} + 6.360 \times 10^{-7} \times P_{1,0} \\ P_{\mathcal{V}} &= 3.002 \times 10^{-7} \times P_{0,0} - 4.713 \times 10^{-7} \times P_{0,1} + 1.679 \times P_{1,0} \end{aligned} \quad (5.18)$$

Nous étudions maintenant un exemple d'identification à partir un champ polynomial de degré 16. On réalise alors une identification des coefficients polynomiaux du *champ polynomial*_{1/3} (cf. figure FIG 5.6). L'écriture littérale des coefficients telle que nous l'avons fait pour le champ affine de degré 1 est trop lourde sous cette forme. Nous exprimons donc simplement l'erreur d'identification par une mesure relative à l'énergie de chaque coefficient polynomial, définie par :

$$E_{\gamma} = \frac{|\tilde{\gamma}_{k,l} - \gamma_{k,l}| \times 100}{|\gamma_{k,l}|} \quad (5.19)$$

où $\gamma_{k,l}$ est le coefficient connu et $\tilde{\gamma}_{k,l}$ son identification. L'identification des coefficients du *champ polynomial*_{1/3} produit les résultats suivants :

- $E_{\gamma_{min}} = 2.609 \times 10^{-8}\%$ le coefficient obtenant l'erreur minimale
- $E_{\gamma_{max}} = 1.757 \times 10^{-3}\%$ le coefficient obtenant l'erreur maximale
- $E_{\gamma_{moy}} = 2.947 \times 10^{-4}\%$ l'erreur moyenne des coefficients

L'erreur obtenue est relativement faible et permet bien de vérifier qu'il s'agit du même mouvement. Il existe donc un moyen à partir d'une base de polynômes épars (définie a posteriori) de se ramener à une base dense (définie a priori). Cependant nous avons un moyen de vérifier qu'il s'agit bien du même mouvement. En effet, après identification, nous obtenons l'expression d'un mouvement épars dans une base dense. Il est alors possible de reconstruire le mouvement polynomial dense. La figure FIG 5.10 présente les résultats obtenus pour la rotation affine épars en FIG 5.10(a) et pour le *champ polynomial*_{1/3} en FIG 5.10(b). Les champs reconstruits par interpolation sont identiques aux champs denses originaux. On peut donc interpolier un mouvement polynomial à partir d'une identification d'un mouvement épars dans la base canonique.

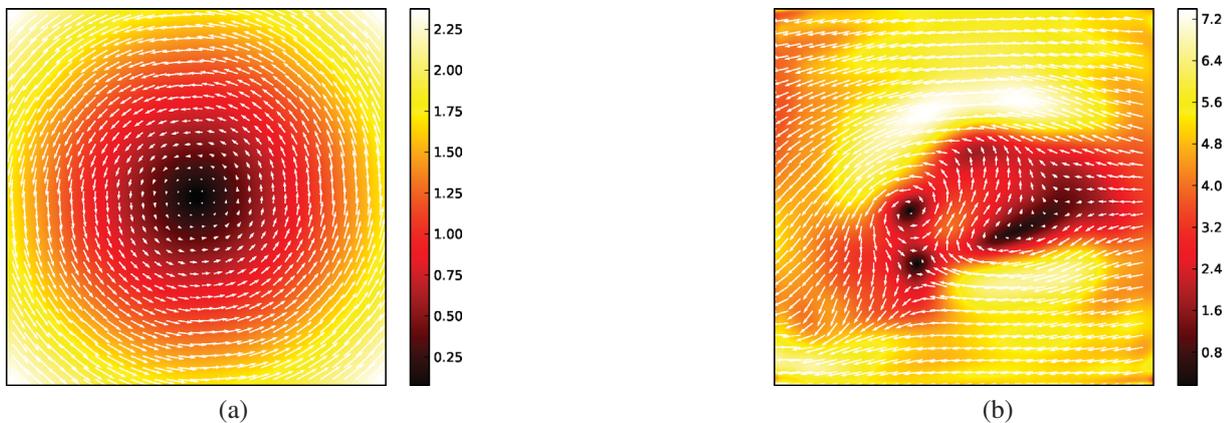


FIGURE 5.10 – Résultats d'interpolation obtenus à partir d'une identification dans la base canonique ; (a) Interpolation d'une rotation affine épars (cf. figure FIG 5.9) (b) Interpolation du *champ polynomial*_{1/3} (cf. figure FIG 5.6).

Comme il est possible d'effectuer une approximation à erreur nulle (ou presque) d'un champ polynomial épars grâce aux bases éparées, nous étudions une approximation par base de polynômes épars sur un champ non polynomial. À partir d'un champ expérimental dense dont le domaine spatial est de 124×124 vecteurs (cf. figure FIG 5.11(a) et FIG 5.11(b)), nous créons un champ épars en annulant de manière régulière l'amplitude de deux vecteurs sur trois horizontalement et verticalement. Ce champ, présenté en figure FIG 5.11(c), est appelé par la suite *cylindre-carré* $\mathcal{C}_{1/3}$. Le champ n'est pas redimensionné, son domaine spatial est toujours de 124×124 vecteurs mais certains vecteurs sont d'amplitude nulle. Nous comparons alors l'approximation globale avec une base dense du champ dense à l'approximation obtenue par identification des coefficients de la base canonique du champ épars correspondant. Les approximations obtenues ainsi que la carte d'erreurs quadratiques entre ces deux approximations sont présentées sur la figure FIG 5.12. On constate qu'une version épars du champ ne conservant que 11, 11% des vecteurs permet d'obtenir une approximation dense du champ de qualité comparable à celle obtenue sur le champ dense. Il est alors possible d'envisager d'utiliser cette technique pour combler les champs lorsque certains vecteurs ne sont pas connus. En effet, certaines techniques d'estimation du mouvement ne fournissent pas des champs denses. Il est alors possible de combler les trous du champ de déplacements à partir d'interpolation effectuée par projection sur une base de polynômes.

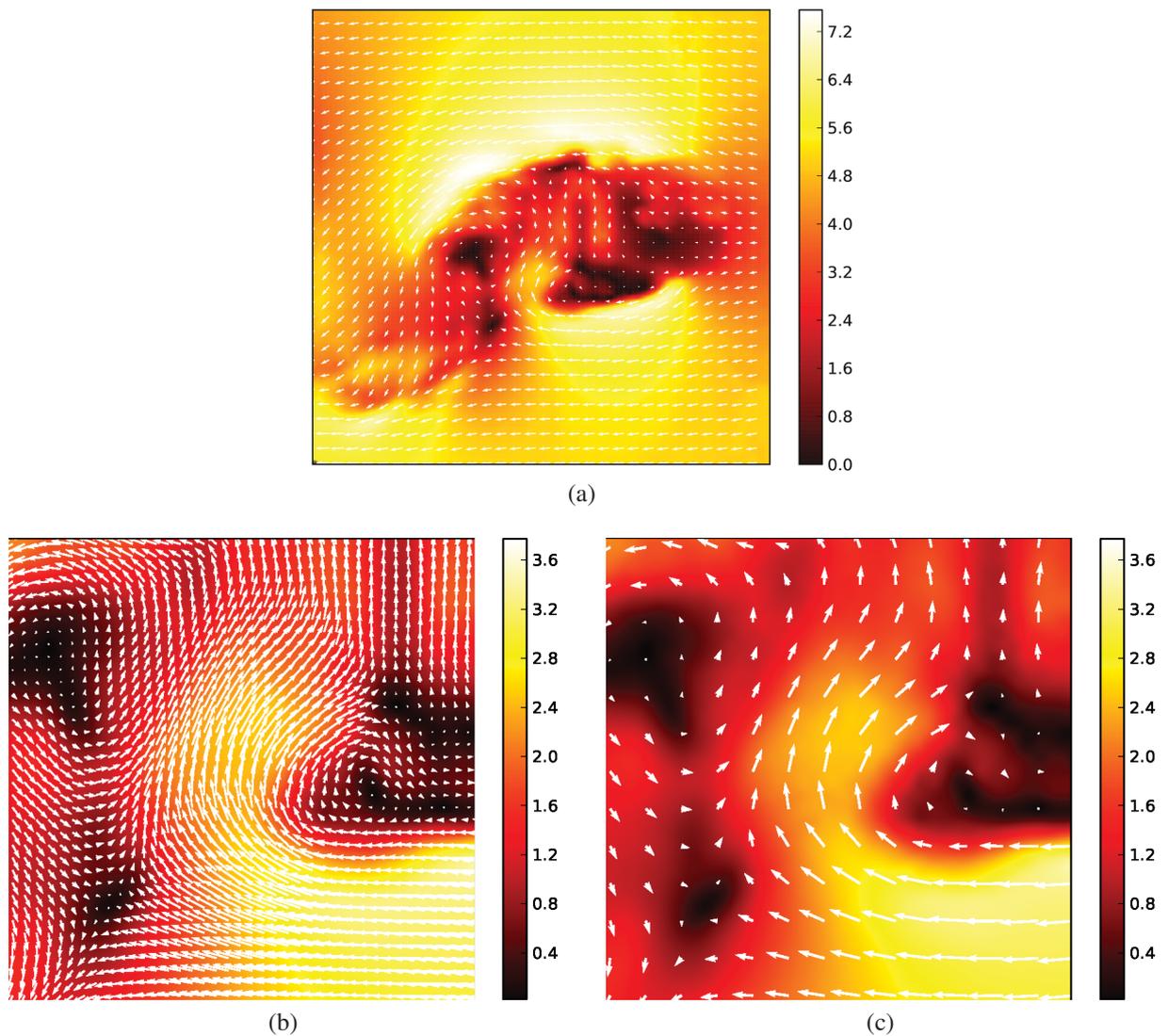


FIGURE 5.11 – Champ cylindre-carré $\mathcal{C}_{1/3}$; (a) Champ (dense de 124×124 vecteurs) 131 de la séquence *Cylindre-Carre* (cf. ANNEXE A); (b) Extrait du champ (a); (c) Extrait du champ *cylindre-carré* $\mathcal{C}_{1/3}$ obtenu en annulant l'amplitude de deux vecteurs sur trois du champ (a) horizontalement et verticalement.

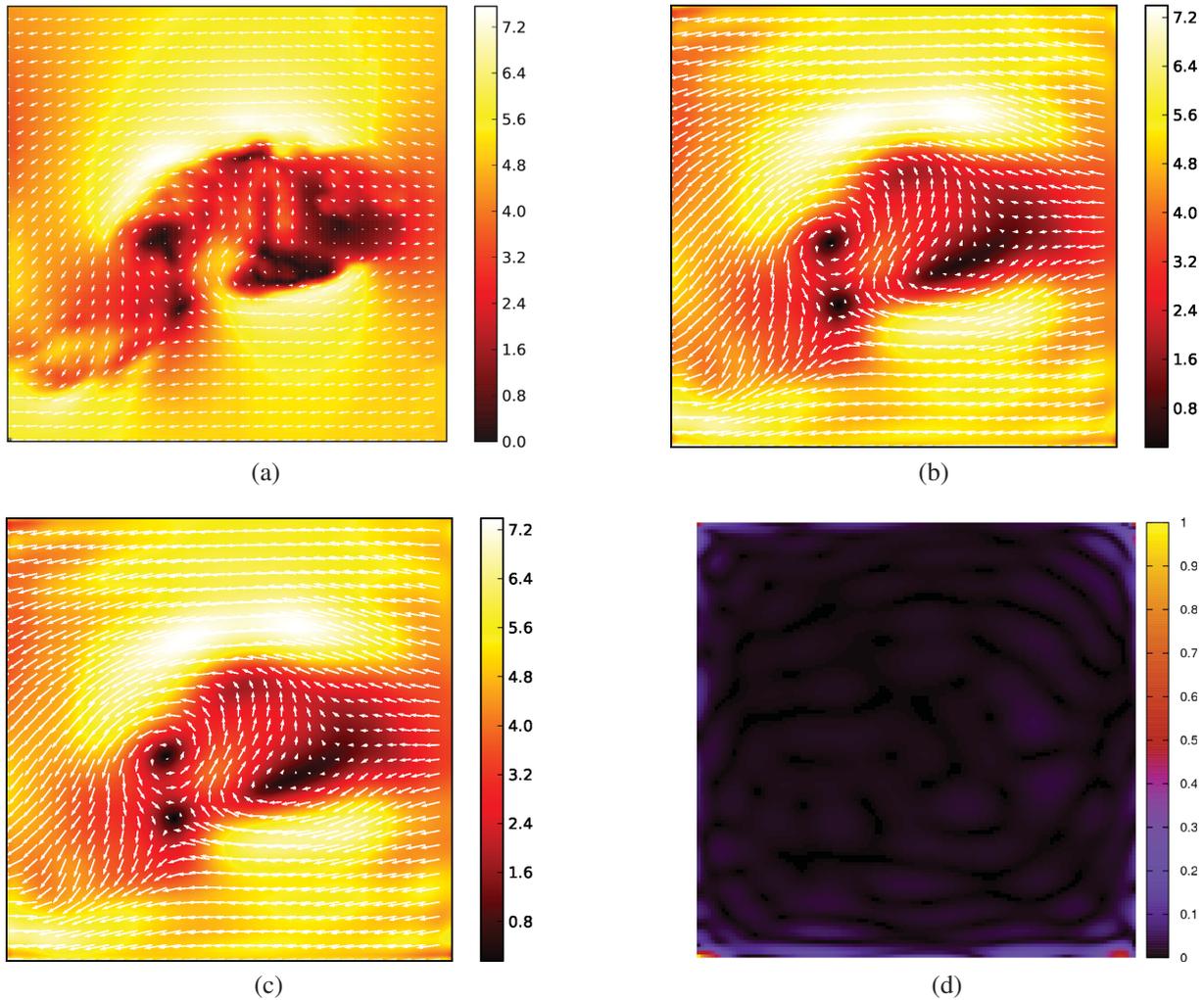


FIGURE 5.12 – Comparaison entre l’approximation globale d’un champ non polynomial dense (en (a)) et l’approximation d’une version éparse du même champ dont l’amplitude de deux vecteurs sur trois a été annulée horizontalement et verticalement ; (a) Champ Cylindre-Carré dense ; (b) Approximation obtenue par projection de (a) sur une base polynomiale dense de degré 16 ; (c) Approximation obtenue par projection de *cylindre-carré* $\mathcal{C}_{1/3}$ sur une base polynomiale éparse de degré 16, après identification et interpolation ; (d) Carte d’erreurs quadratiques entre (b) et (c).

5.4 Approximation hiérarchique avec les bases éparées

Nous avons étudié au chapitre précédent des méthodes d'approximation par subdivision du domaine spatial. Ces méthodes permettent d'obtenir une approximation des champs complexes avec une erreur faible en utilisant des bases polynomiales de degré peu élevé. Nous ne réévaluons pas toutes les techniques proposées avec les bases denses. Parmi les techniques de subdivisions étudiées figure le quadtree qui permet d'obtenir une bonne qualité d'approximation tout en étant très simple à mettre en oeuvre. Nous présentons des résultats d'approximation pour une modélisation par quadtree avec les bases éparées. Nous améliorons aussi les résultats obtenus au chapitre précédent pour la méthode de subdivision relative à l'utilisation du diagramme de Voronoï (distance $L1$ et $L2$) grâce aux bases éparées.

5.4.1 Quadtree avec les bases éparées

Nous utilisons maintenant les bases éparées à l'intérieur de chaque bloc du quadtree. La base doit alors être générée séparément pour chaque bloc en fonction du domaine spatial qu'elle couvre. Le coût calculatoire est donc supérieur à celui obtenu avec les bases denses. Cependant, l'utilisation du quadtree est intéressante avec des bases de degré faible. La figure FIG 5.13 présente une comparaison entre une modélisation par quadtree avec des *bases denses* et avec un quadtree utilisant des *bases éparées* sur le champ *Cylindre-carré éparé 1*. La même étude est effectuée sur le champ *Cylindre-carré éparé 2* en figure FIG 5.14. On peut constater dans les deux cas que nous obtenons une approximation du champ de déplacements éparés de bien meilleure qualité qu'avec les bases denses.

Nous présentons une illustration de l'utilisation du quadtree et des bases de polynômes éparées. Nous avons vu qu'il est possible à partir d'une identification des coefficients dans la base canonique de combler les trous d'un champ éparé. L'algorithme suivant basé sur l'utilisation d'un quadtree et de base éparse permet de combler les trous d'un champ de déplacements éparés.

1. *Modéliser le champ non-dense avec un quadtree et une base de degré D en utilisant le domaine particulier correspondant aux vecteurs non nuls du champ ;*
2. *Pour chaque feuille du quadtree :*
3. *Vérifier si la densité n'est pas nulle :*
 - *si elle n'est pas nulle, effectuer le passage des coefficients de la base éparse vers la base canonique ET reconstruire le champ par interpolation uniquement là où le mouvement n'est pas connu ;*
 - *sinon remonter au noeud père de cette feuille et retourner à l'étape 3 ;*

Nous présentons sur la figure FIG 5.15 une illustration de cet algorithme. Pour plus de clarté, nous utilisons un champ de petites dimensions. Il s'agit d'un extrait du champ *cylindre-carré*, présenté en (a) auquel nous retirons seulement quelques blocs de vecteurs afin d'obtenir le champ présenté en (b). Le résultat d'interpolation de ce champ obtenu par l'algorithme ci-dessus est présenté en (d). Pour effectuer une comparaison, nous présentons aussi le résultat d'une interpolation globale de champ en (c). Nous comparons les résultats d'interpolation obtenus à partir de l'erreur d'interpolation calculé entre (a) et (c) en (e) et entre (a) et (d) en (f). L'utilisation du quadtree permet une meilleure interpolation du mouvement. De plus, il est plus rapide de calculer l'interpolation avec un quadtree associé à une base de degré 1 que de réaliser l'interpolation de manière globale avec une base de degré 16.

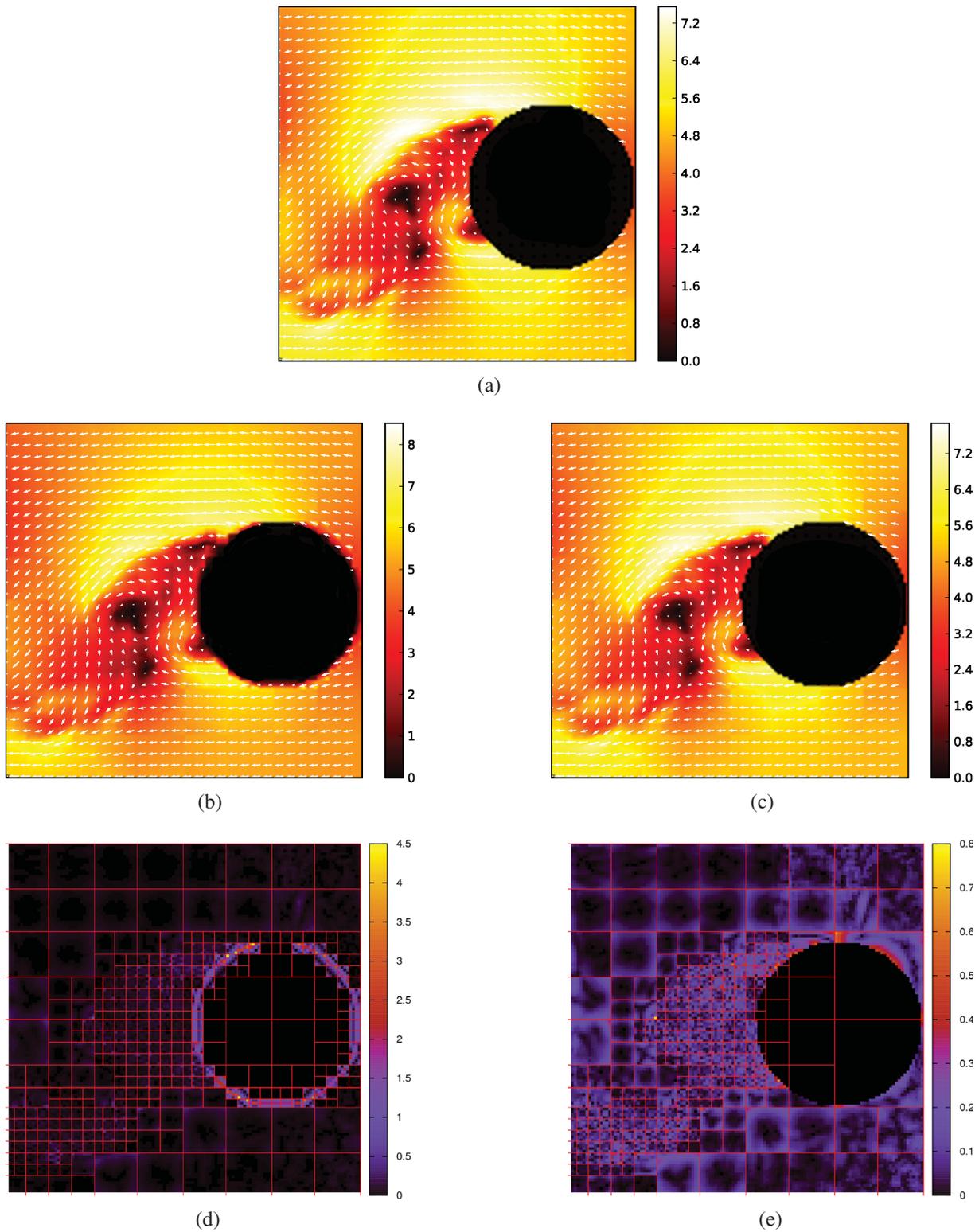


FIGURE 5.13 – Modélisation d'un mouvement complexe non-dense (exemple 1) avec un quadtree associé à une base éparse; (a) Champ non-dense issu de la séquence Cylindre-carré (exemple 1); (b) Modélisation du champ (a) avec un quadtree sur 5 niveaux et une base dense de degré 1; (c) Modélisation du champ (a) avec un quadtree sur 5 niveaux et une base éparse de degré 1; (d) Carte d'erreurs quadratiques associée à la modélisation (b) du champ (a); (e) Carte d'erreurs quadratiques associée à la modélisation (c) du champ (a);

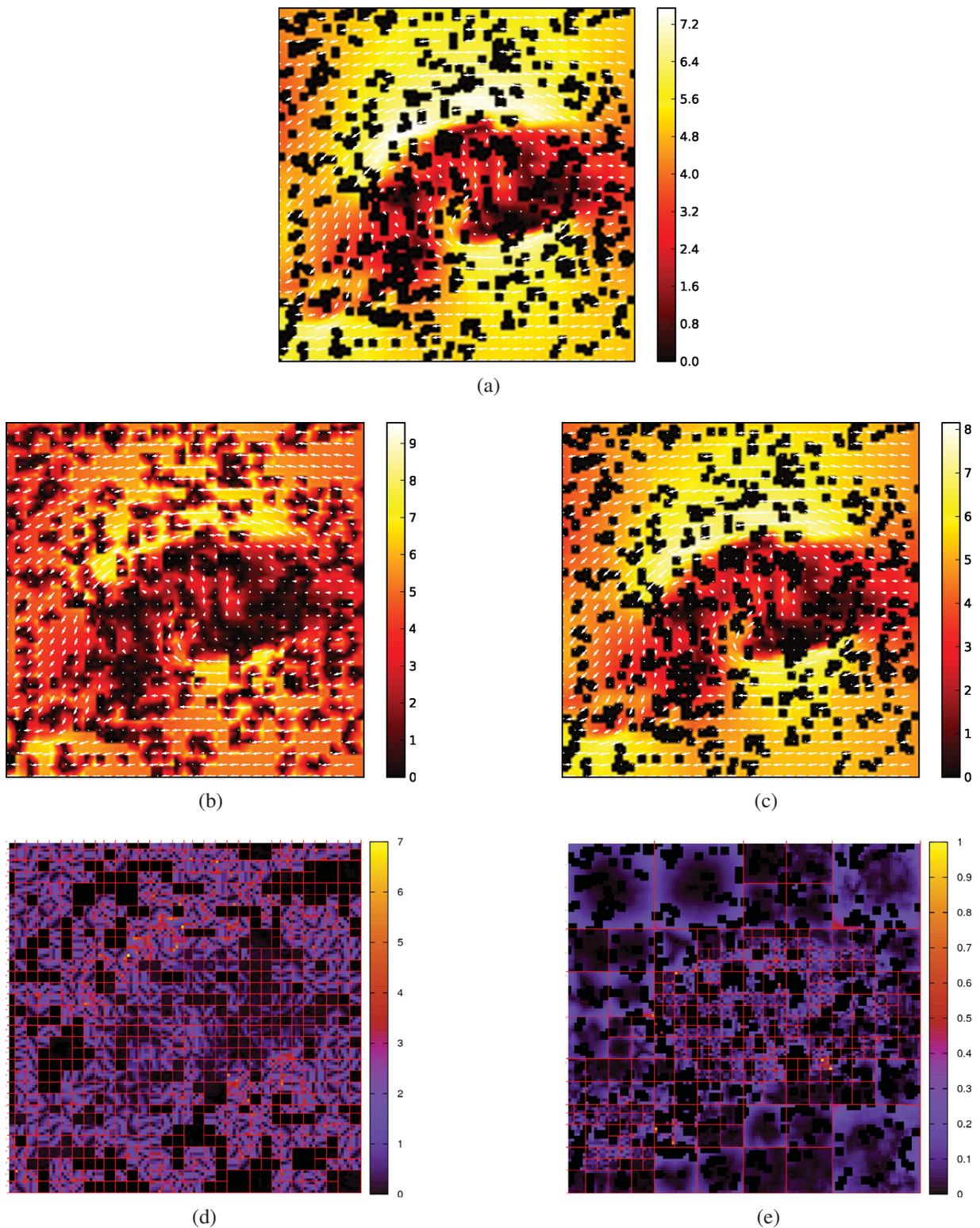


FIGURE 5.14 – Modélisation d'un mouvement complexe non-dense (exemple 1) avec un quadtree associé à une base éparse ; (a) Champ non-dense issu de la séquence Cylindre-carré (exemple 1) ; (b) Modélisation du champ (a) avec un quadtree sur 5 niveaux et une base dense de degré 1 ; (c) Modélisation du champ (a) avec un quadtree sur 5 niveaux et une base éparse de degré 1 ; (d) Carte d'erreurs quadratiques associée à la modélisation (b) du champ (a) ; (e) Carte d'erreurs quadratiques associée à la modélisation (c) du champ (a) ;

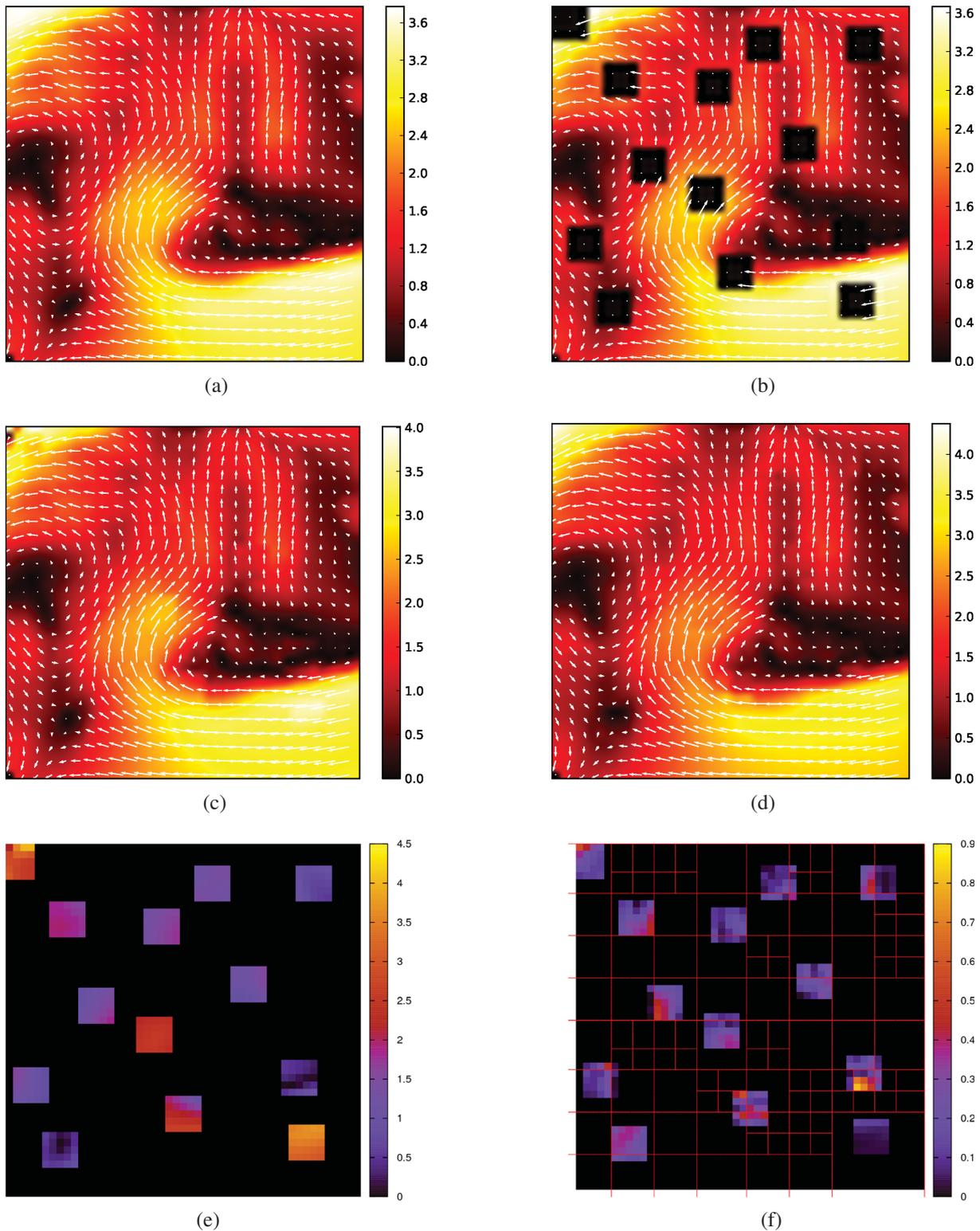


FIGURE 5.15 – Exemple d’interpolation sur un extrait du champ Cylindre-carré non dense ; (a) Extrait du champ cylindre-carré ; (b) Extrait du champ cylindre-carré auquel nous retirons quelques blocs de vecteurs ; (c) Résultat de l’interpolation du champ (b) à partir d’une modélisation globale avec une base éparsée de degré 16 ; (d) Résultat de l’interpolation du champ (b) à partir de l’algorithme d’interpolation utilisant le quadtree, ici sur 5 niveaux et avec une base non-dense de degré 1 ; (e) Carte d’erreurs quadratiques d’interpolation associée à l’interpolation du champ (a) présentée en (c) ; (f) Carte d’erreurs quadratiques d’interpolation associée à l’interpolation du champ (a) présentée en (d).

5.4.2 Diagramme de Voronoï avec les bases éparées

Une autre technique de subdivision vue au chapitre précédent repose sur l'utilisation d'un diagramme de Voronoï. La subdivision est obtenue à partir de points particuliers du champ de déplacements. Cela permet d'obtenir une approximation précise autour des points ayant un intérêt pour l'analyse. Pour la construction du diagramme de Voronoï, plusieurs distances peuvent être utilisées. Dans notre étude, nous avons présenté des résultats avec la distance $L1$ et avec la distance $L2$.

Au chapitre précédent, la distance $L1$ nous a permis d'obtenir de meilleurs résultats d'approximation. Cependant, quelque soit la distance utilisée, l'approximation de chaque région du diagramme s'est avérée difficile. En effet, cette technique produit des régions non rectangulaires, ce qui induit des difficultés d'approximation lorsque la région est modélisée sur une base de polynômes ayant un support dense rectangulaire. Les bases éparées n'ont pas de telles limitations, nous revenons donc sur cette modélisation. En figure FIG 5.16, nous étudions l'approximation d'un champ dense avec une subdivision par diagramme de Voronoï. Sur la figure FIG 5.16(a) et sur la figure FIG 5.16(b), nous présentons le champ à approximer en le superposant aux diagrammes de Voronoï avec les distances $L1$ et $L2$. Nous présentons en figure FIG 5.16(c) et FIG 5.16(d) les cartes d'erreurs obtenues au chapitre précédent avec des boîtes englobantes (non-orientées) et des bases denses de degré 1. Finalement, les figures FIG 5.16(e) et FIG 5.16(f) présentent les cartes d'erreur obtenues avec des bases éparées de degré 1. Les approximations avec les bases éparées sont meilleures que les précédentes. L'erreur quadratique globale obtenue avec les bases éparées et la distance $L1$ est de $EQM = 0.974042$ et de $EQM = 0.816251$ pour la distance $L2$. On peut remarquer que l'erreur est plus faible avec la distance $L2$. Cela montre donc que la distance $L2$ permet d'obtenir une subdivision qui a plus de sens au niveau du mouvement. Le taux de compression obtenu est le même que celui des boîtes englobantes orientées ce qui donne une qualité d'approximation du même ordre de grandeur qu'une approximation globale de degré 9 avec la simplicité d'analyse liée à l'utilisation de base de degré 1. On en déduit que dans le cadre de l'utilisation des bases de polynômes bi-variables denses, il faut privilégier la distance $L1$ puisqu'elle fournit des régions plus rectangulaires, tandis que dans le cadre de l'utilisation des bases de polynômes bi-variables éparées, il faut privilégier la distance $L2$.

5.5 Conclusion

Nous savons maintenant comment réaliser l'approximation d'un champ de déplacements épars. Lorsque l'objectif est d'obtenir une approximation, la projection du champ sur une base de polynômes non dense permet de reconstruire les champs polynomiaux sans erreur et d'obtenir une approximation intéressante des champs non-polynomiaux. Lorsqu'il s'agit de permettre une analyse, il est alors intéressant de réaliser l'identification des coefficients dans la base canonique. Cela permet d'exprimer des champs de densités différentes dans une même base afin de les comparer. De plus, à partir des coefficients exprimés dans la base canonique, il est possible de réaliser une interpolation des vecteurs manquants. Ces bases de polynômes peuvent être utilisées avec les différentes techniques d'approximation par subdivision étudiées au chapitre précédent. Elles permettent même d'obtenir de meilleurs résultats pour la technique utilisant le diagramme de Voronoï. Enfin, à partir de l'approximation par quadtree et des bases de polynômes éparées, nous avons proposé une technique d'interpolation des vecteurs nuls d'un champ épars.

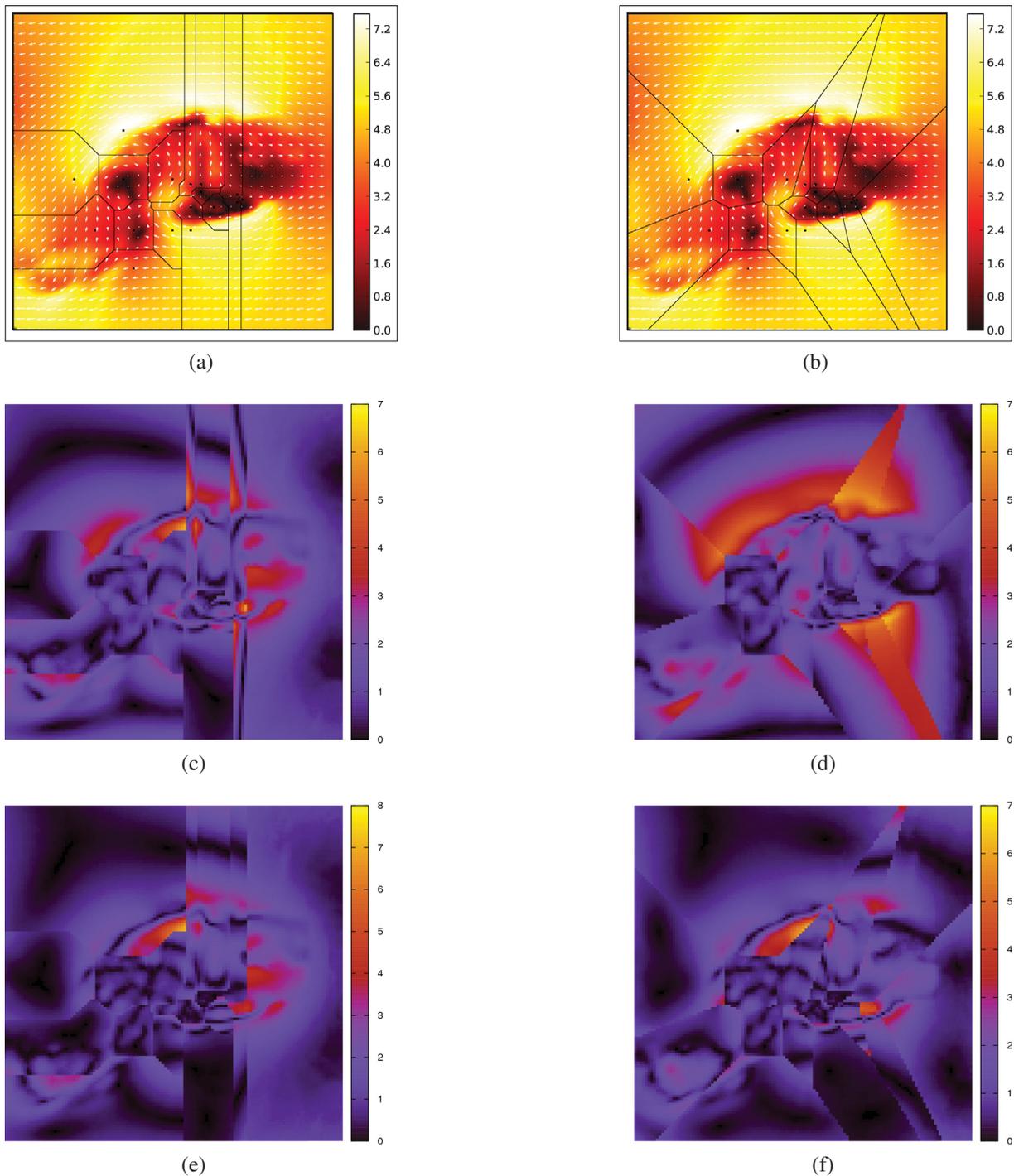


FIGURE 5.16 – Utilisation des bases de polynômes éparses pour une approximation basée sur le diagramme de Voronoï ; (a) Champ test superposé à la subdivision obtenue par diagramme de Voronoï avec une distance $L1$; (b) Champ test superposé à la subdivision obtenue par diagramme de Voronoï avec une distance $L2$; (c) Approximation de la subdivision (b) avec des bases de polynômes denses de degré 1 ; (d) Approximation de la subdivision (c) avec des bases de polynômes denses de degré 1 ; (e) Approximation de la subdivision (b) avec des bases de polynômes éparses de degré 1 ; (f) Approximation de la subdivision (c) avec des bases de polynômes éparses de degré 1 ;

CHAPITRE 6

BASES DE POLYNÔMES A-COMPLÈTES

6.1 Vers une approximation à erreur nulle

Nous avons proposé au cours des chapitres précédents différentes méthodes d'approximation utilisant les bases de polynômes orthogonaux. Nous avons d'abord étudié la modélisation globale d'un champ de déplacements en le projetant sur une base de polynômes. Cette méthode permet d'obtenir un très bon taux de compression pour une qualité d'approximation comparable à celle d'une approximation avec les premiers vecteurs d'une décomposition aux valeurs propres. Cependant, il est souvent nécessaire d'utiliser des bases de degrés élevés. Dans ce cas, les bases peuvent être longues à générer et l'analyse des coefficients devient complexe quand le degré de la base augmente.

Afin d'obtenir une meilleure qualité d'approximation, tout en limitant l'augmentation du degré de la base, nous avons proposé différentes techniques de subdivision du champ. Parmi ces méthodes, le quadtree permet d'obtenir une approximation de qualité du champ de déplacements tout en conservant un fort taux de compression. Le quadtree est rapide et simple à mettre en œuvre. Une autre méthode de subdivision basée sur le diagramme de Voronoï est proposée. L'avantage de cette technique est de permettre une approximation de qualité de certaines zones d'intérêt tout en simplifiant l'analyse du champ de déplacements.

Ces techniques d'approximation sont efficaces sur les champs denses, cependant l'approximation des champs épars reste difficile même avec ces méthodes. Nous avons donc proposé au chapitre 5, une méthode pour générer des bases prenant en compte la densité du champ. À partir de ces nouvelles bases, il a été possible d'approximer les champs épars avec les différentes techniques de subdivision étudiées.

Cependant, dans tous ces cas, seuls les champs polynomiaux peuvent être approximés avec une erreur nulle. L'objet de ce chapitre est de montrer que les bases de polynômes peuvent permettre d'obtenir une erreur d'approximation nulle. En nous appuyant sur le polynôme d'interpolation de Lagrange, nous proposons une méthode pour générer une nouvelle famille de bases polynomiales. Nous comparons ensuite cette méthode d'approximation avec la Transformée en Cosinus Discret (DCT) ainsi qu'avec la Décomposition en Valeurs Singulières (SVD).

Enfin, nous utilisons ces bases polynomiales pour réaliser une analyse multi-résolutions d'un champ de déplacements.

6.2 Du Polynôme d'interpolation de Lagrange aux bases de polynômes A-complètes

6.2.1 Polynôme d'interpolation de Lagrange

Soit une fonction \mathcal{U} définie sur un domaine discret $\Omega = \{x(i)\}_{i=0\dots n}$ par

$$\begin{aligned} \mathcal{U} : \quad \mathbb{R} &\rightarrow \mathbb{R} \\ (x(i)) &\mapsto \mathcal{U}(x(i)) \end{aligned} \quad (6.1)$$

Nous cherchons s'il existe un polynôme tel que :

$$L(x(i)) = \mathcal{U}(x(i)), \forall i \in \Omega \quad (6.2)$$

Ce polynôme existe, c'est le polynôme d'interpolation de Lagrange défini par :

$$L(x) = \sum_{i=0}^n \left(\prod_{k \neq i} \frac{x - x(k)}{x(i) - x(k)} \right) \mathcal{U}(x(i)) \quad (6.3)$$

Ce polynôme est l'unique polynôme de degré au plus n vérifiant l'équation (6.2).

On peut l'écrire sous la forme :

$$L(x) = \sum_{i=0}^n l_i(x) \mathcal{U}(x(i)) \quad (6.4)$$

où les polynômes $l_i(x)$ sont définis par l'équation (6.5). Ces polynômes forment une base.

$$l_i(x) = \prod_{k \neq i} \frac{x - x(k)}{x(i) - x(k)} \quad (6.5)$$

Il est alors possible d'obtenir une approximation à erreur nulle à partir d'une autre base de polynômes bivariés. Nous démontrons dans la section suivante comment obtenir le polynôme d'interpolation de Lagrange par projection sur une base de polynômes.

6.2.2 Les Bases A-complètes

Commençons par raisonner en dimension 1 sur le domaine discret $\Omega = x(i)_{i=0\dots n}$. Dans ce cas, le produit scalaire de deux fonctions \mathcal{U} et \mathcal{V} est défini par

$$\langle \mathcal{U} | \mathcal{V} \rangle_w = \sum_{i=0}^n \mathcal{U}(x(i)) \mathcal{V}(x(i)) w(x(i)) \quad (6.6)$$

où apparaît une fonction de pondération w vérifiant $\forall i, w(x(i)) > 0$.

Pour générer une base orthonormale, nous avons vu au chapitre 3 une méthode utilisant la procédure de récurrence à trois termes. Dans le cas unidimensionnel, la récurrence à trois termes est obtenue par :

$$\begin{aligned} T_i(x) &= (x - \langle xP_{i-1} | P_{i-1} \rangle_w) P_{i-1}(x) - P_{i-2}(x) \\ P_i(x) &= \frac{T_i(x)}{|T_i|_w} \end{aligned} \quad (6.7)$$

Définition 3 Une base A-Complète, relative à un certain produit scalaire, est la base orthogonale définie sur Ω par la famille $P_i(x)_{i=0\dots n}$

En posant $r_k = \sqrt{w(x(k))}$, on peut former la matrice suivante :

$$R_{\mathbb{B}} = \begin{pmatrix} r_0 P_0(x(0)) & \dots & r_j P_0(x(j)) & \dots & r_n P_0(x(n)) \\ \vdots & & \vdots & & \vdots \\ r_0 P_i(x(0)) & \dots & r_j P_i(x(j)) & \dots & r_n P_i(x(n)) \\ \vdots & & \vdots & & \vdots \\ r_0 P_n(x(0)) & \dots & r_j P_n(x(j)) & \dots & r_n P_n(x(n)) \end{pmatrix} \quad (6.8)$$

En posant λ_i et λ_j deux vecteurs ligne de cette matrice, tel qu'ils sont présentés en équation (6.9), on peut montrer que la matrice $R_{\mathbb{B}}$ est orthogonale.

$$\begin{aligned} \lambda_i &= r_0 P_i(x(0)) \dots r_j P_i(x(j)) \dots r_n P_i(x(n)) \\ \lambda_j &= r_0 P_j(x(0)) \dots r_j P_j(x(j)) \dots r_n P_j(x(n)) \end{aligned} \quad (6.9)$$

En effet, leur produit scalaire au sens classique du terme donne :

$$\lambda_i \cdot \lambda_j = \sum_k r_k^2 P_i(x(k)) P_j(x(k)) \quad (6.10)$$

soit, par définition

$$\lambda_i \cdot \lambda_j = \langle P_i | P_j \rangle_w = \delta_{i,j} \quad (6.11)$$

L'orthogonalité de $R_{\mathbb{B}}$ implique :

$$R_{\mathbb{B}} R_{\mathbb{B}}^T = R_{\mathbb{B}}^T R_{\mathbb{B}} = \mathbb{I}_{n+1} \quad (6.12)$$

où \mathbb{I}_{n+1} est la matrice identité. Ces relations induisent les deux conséquences suivantes :

1. à partir de $R_{\mathbb{B}} R_{\mathbb{B}}^T = \mathbb{I}_{n+1}$, on retrouve la propriété initiale d'orthogonalité puisque chaque élément de $R_{\mathbb{B}} R_{\mathbb{B}}^T$ peut s'écrire

$$\sum_k r_k^2 P_i(x(k)) P_j(x(k)) \text{ soit } \langle P_i | P_j \rangle_w = \delta_{i,j} \quad (6.13)$$

2. à partir de $R_{\mathbb{B}}^T R_{\mathbb{B}} = \mathbb{I}_{n+1}$, chaque élément de $R_{\mathbb{B}}^T R_{\mathbb{B}}$ peut s'écrire

$$\sum_{i=0}^n r_j r_k P_i(x(j)) P_i(x(k)) = \delta_{j,k} \quad (6.14)$$

on obtient alors la propriété

$$\sum_{i=0}^n P_i(x(j))P_i(x(k)) = \frac{\delta_{j,k}}{r_j r_k} \quad (6.15)$$

L'approximation de la fonction \mathcal{U} dans une base de polynômes s'obtient par projection

$$\tilde{\mathcal{U}}(x) = \sum_{i=0}^n \langle \mathcal{U} | P_i \rangle_w P_i(x) = \sum_{i=0}^n \left(\sum_{j=0}^n \mathcal{U}(x(j)) P_i(x(j)) w(x(j)) \right) P_i(x) \quad (6.16)$$

soit, en inversant les sommations

$$\tilde{\mathcal{U}}(x) = \sum_{j=0}^n \left(\sum_{i=0}^n P_i(x(j)) P_i(x) \right) \mathcal{U}(x(j)) w(x(j)) \quad (6.17)$$

Afin d'établir le lien avec l'interpolation de Lagrange, on calcule $\tilde{\mathcal{U}}(x(k))$ pour $k = 0 \dots n$, soit

$$\tilde{\mathcal{U}}(x(k)) = \sum_{j=0}^n \left(\sum_{i=0}^n P_i(x(j)) P_i(x(k)) \right) \mathcal{U}(x(j)) w(x(j)) \quad (6.18)$$

D'après la seconde conséquence de l'orthogonalité de $R_{\mathbb{B}}$, l'équation (6.18) devient :

$$\tilde{\mathcal{U}}(x(k)) = \sum_{j=0}^n \frac{\delta_{jk}}{\sqrt{w(x(j))w(x(k))}} w(x(j)) \mathcal{U}(x(j)) \quad (6.19)$$

On obtient finalement

$$\tilde{\mathcal{U}}(x(k)) = \mathcal{U}(x(k)) \quad (6.20)$$

Ceci montre que $\tilde{\mathcal{U}}$ est un polynôme de degré au plus n vérifiant $\tilde{\mathcal{U}}(x(i)) = \mathcal{U}(x(i))$ pour tout i . Le degré de la base $P_i(x)_{i=0 \dots n}$ étant n , on peut alors en déduire que $\tilde{\mathcal{U}}$ n'est autre que le polynôme d'interpolation de Lagrange.

6.2.3 Bases A-complètes en dimension 2

En dimension 2, le domaine discret est

$$\Omega = x(i, j) = (x_1(i), x_2(j))_{i \in [0, n_1], j \in [0, n_2]} \quad (6.21)$$

Le produit scalaire devient

$$\langle \mathcal{U} | \mathcal{V} \rangle_w = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \mathcal{U}(x(i, j)) \mathcal{V}(x(i, j)) w(x(i, j)) \quad (6.22)$$

où Ω est toujours une fonction de pondération. Cette fonction de pondération permet de déterminer le type de base utilisée. Le tableau TAB 6.1 rappelle quelques familles de polynômes en donnant la fonction de pondération associée.

À partir de ce produit scalaire, une base A-complète en dimension 2 peut être générée soit par l'extension de la récurrence à trois termes (cf. chapitre 3), soit par produit tensoriel de bases

Famille	Ω	$w(x_1, x_2)$
Legendre	$[-1; 1]^2$	1
Tchebychev 1	$[-1; 1]^2$	$\frac{1}{\sqrt{(1-x_1)^2(1-x_2)^2}}$
Tchebychev 2	$[-1; 1]^2$	$\sqrt{(1-x_1)^2(1-x_2)^2}$
Laguerre	$[0; \infty]^2$	$e^{-(x_1+x_2)}$
Hermite	$[-\infty; \infty]^2$	$e^{-\frac{x_1^2+x_2^2}{2}}$

TABLEAU 6.1 – Quelques fonctions de pondération w permettant d'obtenir différentes familles de polynômes

en dimension 1, soit par orthonormalisation de la base canonique $K_{i,j}(x) = x_1^i x_2^j$ en utilisant comme ordre sur les couples d'indices, l'ordre lexicographique suivant

$$(i, j) \prec (l, m) \equiv \begin{cases} i + j < l + m \\ j < m \text{ si } i + j = l + m \end{cases} \quad (6.23)$$

Ici, nous utilisons la formule de récurrence à trois termes que nous avons utilisé dans les chapitres précédents et que nous avons présenté en équations (3.14) et (3.15). Cependant, nous générons maintenant tous les polynômes $P_{i,j}$ tel que $i < N$ et $j < N$. Dans les chapitres précédents, le degré D de la base définissait le nombre N_p de polynômes de cette base par la relation $N_p = \frac{(D+1) \times (D+2)}{2}$. Maintenant c'est simplement le domaine qui détermine le nombre de polynômes de la base. Ces bases sont appelées dans la suite du document : bases polynomiales *A-complètes*.

Définition 4 Une base *A-complète* en dimension 2, relative à un produit scalaire donné, est la base orthonormale définie sur Ω par la famille

$$P_{i,j}(x)_{i=0\dots n_1, j=0\dots n_2} \quad (6.24)$$

De même qu'en dimension 1, en posant $r_{i,j} = \sqrt{w(x(i, j))}$ on obtient la représentation matricielle de la famille $P_{i,j}(x)$

$$R_{\mathbb{B}} = \begin{pmatrix} r_{0,0}P_{0,0}(x(0,0)) & \dots & r_{i,j}P_{0,0}(x(i,j)) & \dots & r_{n_1,n_2}P_0(x(n_1,n_2)) \\ \vdots & & \vdots & & \vdots \\ r_{0,0}P_{i,j}(x(0,0)) & \dots & r_{i,j}P_{i,j}(x(i,j)) & \dots & r_{n_1,n_2}P_i(x(n_1,n_2)) \\ \vdots & & \vdots & & \vdots \\ r_{0,0}P_{n_1,n_2}(x(0,0)) & \dots & r_{i,j}P_{n_1,n_2}(x(i,j)) & \dots & r_{n_1,n_2}P_n(x(n_1,n_2)) \end{pmatrix} \quad (6.25)$$

On retrouve les propriétés d'orthogonalité suivantes :

$$R_{\mathbb{B}} R_{\mathbb{B}}^T = R_{\mathbb{B}}^T R_{\mathbb{B}} = \mathbb{I}_{(n_1+1) \times (n_2+1)} \quad (6.26)$$

Comme en dimension 1, ces propriétés nous permettent de vérifier que l'on obtient une approximation à erreur nulle par projection sur une base A-complète.

Nous rappelons deux méthodes de collocation déjà présentées au chapitre 3. La première, la plus classique, est la collocation uniforme présentée en équation 6.27 et la seconde est la collocation de Gauss-Tchebychev présentée en équation 6.28. La collocation de Gauss-Tchebychev

correspond aux zéros des polynômes de Tchebychev et est optimale au sens de la quadrature de Gauss.

$$x(i)_k = -1 + k \frac{2}{N-1} \quad (6.27)$$

$$x(i)_k = -\cos\left(\frac{2k+1}{2N}\pi\right) \quad (6.28)$$

Afin de s'assurer qu'il est possible de reconstruire une image avec une erreur d'approximation nulle quelque soit l'image, nous projetons sur notre base A-complète une image en niveau de gris représentant un bruit défini par une loi uniforme. La figure FIG 6.1 représente l'image originale représentant ce bruit en FIG 6.1(a) et le résultat de sa reconstruction après projection dans une base de polynômes orthogonaux A-complète en FIG 6.1(b). L'erreur d'approximation obtenue est nulle.

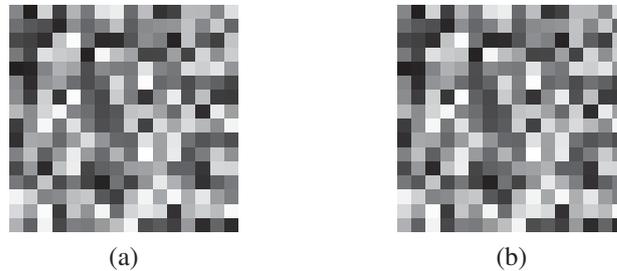


FIGURE 6.1 – Reconstruction parfaite d'une image après projection dans une base de polynômes bi-variables A-complète ; (a) Image d'origine représentant un bruit généré par une loi uniforme ; (b) Image reconstruite à partir de la projection dans une base A-complète de polynômes orthogonaux (erreur d'approximation nulle)

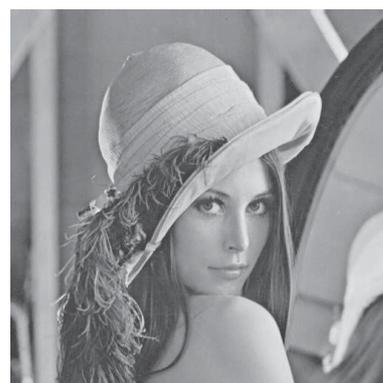
6.3 Bases A-complètes et approximation à erreur nulle

Dans l'exemple présenté en figure FIG 6.1, l'image est de 16×16 pixels. Il est donc simple et rapide de générer une base ayant autant de polynômes que de pixels dans l'image. Cependant, générer une base polynomiale à la dimension d'un champ de déplacements ou d'une image réelle est très coûteux en temps de calcul. En effet, il n'est pas rare d'obtenir des champs de déplacements de plus de 1024×1024 vecteurs. Les images peuvent maintenant dépasser plusieurs dizaines de mégapixels. Dans ce cas, il est nécessaire d'utiliser une méthode de subdivision telle qu'une grille régulière composée de blocs ayant une taille raisonnable. Cela permet d'obtenir une approximation à erreur nulle par bloc en conservant des temps de calcul acceptables. Cette technique a été utilisée avec succès dans le format de compression d'image fixe jpeg où la transformée en Cosinus Discret est appliquée sur des blocs de 8×8 pixels.

Nous présentons, sur la figure FIG 6.2, l'approximation d'une image à partir des bases A-complètes avec une grille de 2×2 pixels. Nous utilisons donc une base contenant 4 éléments qui sont les éléments $P_{0,0}$, $P_{0,1}$, $P_{1,0}$ et $P_{1,1}$. Sur la figure FIG 6.2(a), nous présentons l'image test et en FIG 6.2(b), le résultat de l'approximation. On constate que l'erreur d'approximation est nulle. Ce résultat est identique quelque soit la grille utilisée.



(a)



(b)

FIGURE 6.2 – Approximation de l’image Lena (512×512 pixels) avec les bases A-complètes avec une grille régulière de 2×2 pixels (a) image Lena originale ; (b) image reconstruite

6.3.1 Comparaison avec la Transformée en Cosinus Discrète (DCT)

Nous comparons les bases de polynômes A-complètes avec la Transformée en Cosinus Discrète (DCT). La DCT a été choisie comme transformée dans la norme JPEG. La norme JPEG définit le format d’enregistrement et l’algorithme de décodage pour la compression des images numériques. C’est pour cette raison que nous souhaitons nous comparer à cette transformée. Nous ne nous intéressons pas ici à la compression des données de la norme JPEG mais uniquement à la DCT. Nous n’étudions donc pas la quantification et les différents codages (RLE, Huffman, etc).

Nous avons vu en figure FIG 6.2 qu’il est possible de reconstruire parfaitement une image à partir des coefficients polynomiaux d’une base A-complète. Nous étudions maintenant la qualité de reconstruction lorsque l’on ne conserve que quelques coefficients. Nous effectuons nos tests sur l’image “bikes” présentée sur la figure FIG 6.3. Cette image est extraite de la base LIVE Image Quality Database [Sheikh 03], [Sheikh 06], [Wang 04].



FIGURE 6.3 – Image “bikes” extraite de la base LIVE Image Quality Database [Sheikh 03], [Sheikh 06], [Wang 04]

Nous comparons la qualité d’approximation obtenue avec une base A-complète et avec une DCT lorsque la même quantité d’énergie est conservée dans chaque bloc. Nous évaluons la

qualité des images par le rapport signal maximum sur bruit (PSNR pour “Peak Signal to Noise Ratio”) défini par l’équation (6.29) :

$$PSNR = 10 \cdot \log_{10} \frac{d}{EQM} \quad (6.29)$$

où d est la dynamique du signal, pour une image sur 8 bits $d = 256$. Les images sont approximées avec une grille régulière pour les deux transformées étudiées. Sur la figure FIG 6.4, nous présentons les résultats obtenus avec une grille de 8×8 pixels en FIG 6.4(a) et avec une grille de 16×16 pixels en FIG 6.4(b). Ici la base A-complète utilisée est générée avec une base de Legendre et une fonction de collocation uniforme. Nous représentons la qualité de l’image (PSNR) en fonction de la quantité d’énergie conservée pour chaque bloc. Pour chaque bloc, les coefficients sont triés en fonction de leur énergie. On conserve alors les coefficients les plus énergétiques jusqu’à obtenir par exemple 90% de l’énergie totale du bloc. Les autres coefficients sont alors annulés. Le bloc correspondant dans l’image résultat est alors reconstruit à partir des coefficients conservés. On constate que la qualité d’approximation obtenue par ces deux transformées est très proche.

Nous avons aussi effectué la comparaison entre une base A-complète de polynômes de Legendre pour une fonction de collocation de Gauss-Tchebychev (cf équation (6.28)) et une DCT. Nous constatons que les résultats d’approximation sont identiques pour ces deux bases. De plus, les coefficients obtenus par les deux bases sont très proches en valeur absolue. Les tableaux TAB 6.2 et TAB 6.3 présentent respectivement les coefficients de la base A-complète et de la DCT pour un bloc de 4×4 pixels choisi au hasard dans l’image “bikes”.

Les liens entre la transformée en cosinus discret et les polynômes de Tchebychev ont été démontrés dans [Neagoe 90], [WANG 96], [Baszenski 97], [Chu 00]. Dans le cas $1D$, les coefficients polynomiaux sont obtenus à partir des coefficients DCT par la relation suivante :

$$\tilde{u}_k = (-1)^k DCT(k) \quad (6.30)$$

où $DCT(k)$ est le $k^{\text{ème}}$ coefficient obtenu par DCT.

La modification est un changement de signe. Dans notre cas, en $2D$, il suffit alors de changer le signe de certains coefficients par rapport à la règle suivante. Pour les lignes impaires, on doit changer le signe de tous les coefficients des colonnes paires et pour les lignes paires, on doit changer le signe de tous les coefficients des colonnes impaires. Cela se traduit par la relation

$$\tilde{u}_{k,l} = (-1)^k (-1)^l DCT(k, l) \quad (6.31)$$

où $DCT(k, l)$ est le coefficient DCT de la ligne k et de la colonne l .

Ainsi, les bases A-complètes permettent d’obtenir une approximation de qualité comparable, voire identique sous certaines conditions, à celle obtenue avec une DCT. Cela met en exergue l’intérêt d’une telle transformée pour des aspects de compression de données.

6.3.2 Comparaison avec la SVD

La SVD que nous avons présenté au chapitre 2.3 est une méthode de compression de données optimale au sens de l’énergie. Nous comparons donc l’approximation d’une image avec une SVD et par base A-complète avec une grille régulière. Nous continuons à utiliser l’image “bikes” (cf. figure FIG 6.3) pour présenter nos résultats.

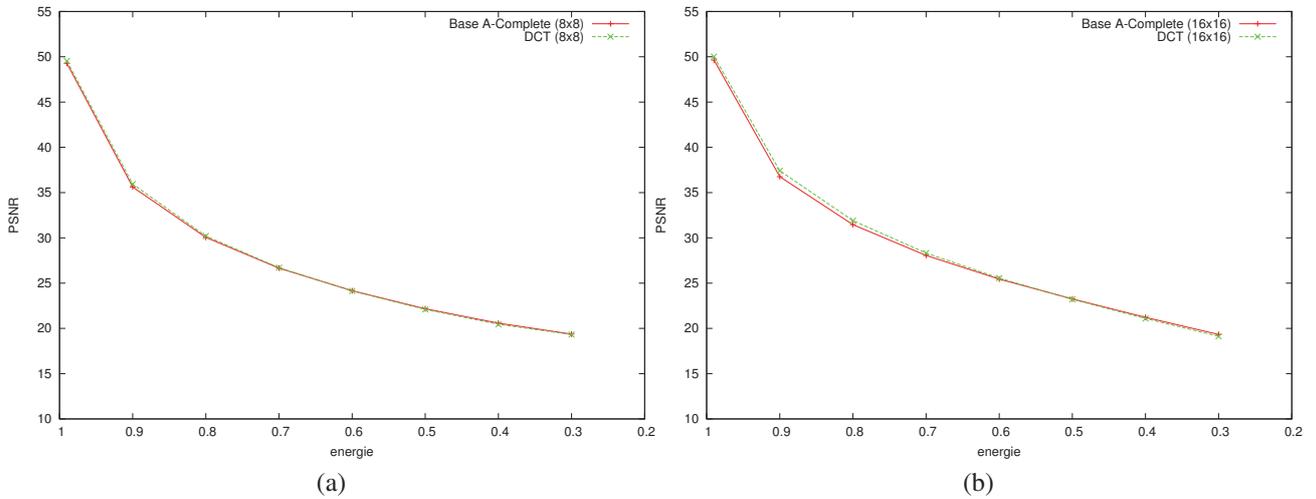


FIGURE 6.4 – Evolution du PSNR en fonction de la quantité d'énergie conservée pour les coefficients DCT ou polynomiaux de chaque bloc ; Les courbes en trait continu rouge correspondent aux bases de polynômes A-complètes et les traits discontinus vert correspondent aux DCT ; (a) La grille est de 8×8 pixels ; (b) La grille est de 16×16 pixels.

+1.461	+1.206 $\times 10^{-2}$	+3.922 $\times 10^{-3}$	+3.494 $\times 10^{-3}$
-2.518 $\times 10^{-2}$	-9.061 $\times 10^{-3}$	-5.12378 $\times 10^{-3}$	-5.714 $\times 10^{-3}$
+1.373 $\times 10^{-2}$	-5.435 $\times 10^{-3}$	+1.80411 $\times 10^{-16}$	+2.251 $\times 10^{-3}$
-4.427 $\times 10^{-3}$	+1.682 $\times 10^{-4}$	-2.122 $\times 10^{-3}$	-7.425 $\times 10^{-4}$

TABLEAU 6.2 – Tableau représentant les coefficients polynomiaux d'un bloc de 4×4 pixels choisi au hasard dans l'image "bikes" ; Ce bloc est le même que celui utilisé pour le tableau TAB 6.3 ; Les coefficients en gras sont ceux dont le signe change par rapport au tableau TAB 6.3. Ici la base A-complète est une base de polynômes de Legendre avec une collocation de Gauss-Tchebychev.

+1.461	-1.206 $\times 10^{-2}$	+3.922 $\times 10^{-3}$	-3.494 $\times 10^{-3}$
+2.518 $\times 10^{-2}$	-9.061 $\times 10^{-3}$	+5.12378 $\times 10^{-3}$	-5.714 $\times 10^{-3}$
+1.373 $\times 10^{-2}$	+5.435 $\times 10^{-3}$	+1.80411 $\times 10^{-16}$	-2.251 $\times 10^{-3}$
+4.427 $\times 10^{-3}$	+1.682 $\times 10^{-4}$	+2.122 $\times 10^{-3}$	-7.425 $\times 10^{-4}$

TABLEAU 6.3 – Tableau représentant les coefficients DCT d'un bloc de 4×4 pixels choisi au hasard dans l'image "bikes" ; Ce bloc est le même que celui utilisé pour le tableau TAB.6.2 ; Les coefficients en gras sont ceux dont le signe change par rapport au tableau TAB.6.2.

Nous étudions, pour la SVD et pour les bases A-complètes, l'évolution du PSNR (cf. équation (6.29)) en fonction du gain de compression τ de coefficients conservés. Ce gain de compression est calculé par l'équation (6.32).

$$\tau = 1 - \frac{\text{Nombre de coefficients conservés}}{\text{Nombre total de pixels dans l'image}} \quad (6.32)$$

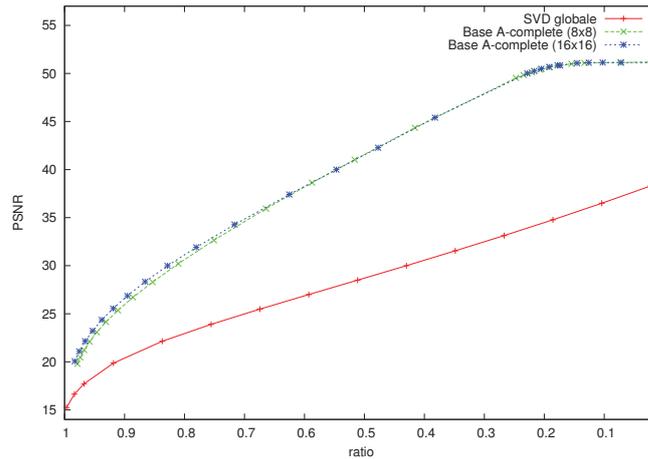


FIGURE 6.5 – Evolution du PSNR en fonction du gain de compression pour une SVD globale et pour une approximation obtenue par projection sur une base A-complète avec une grille régulière ; La courbe en trait continu rouge représente le résultat obtenu par SVD ; La courbe en trait discontinu vert représente la base A-complète avec une grille de 8×8 pixels ; La courbe en pointillés bleu représente la base A-complète avec une grille de 16×16 pixels ;

Dans un premier temps, sur le graphique de la figure FIG 6.5, nous comparons l'approximation obtenue avec une SVD appliquée à l'image entière et l'approximation obtenue par projection sur une base A-complète avec une grille régulière de 8×8 pixels et pour une grille régulière de 16×16 pixels.

Afin de comparer les deux méthodes dans le même contexte, nous appliquons la SVD sur une grille régulière. Pour chaque bloc, on ne conserve qu'un nombre réduit de vecteurs afin de ne pas avoir un gain de compression supérieur à 1. En effet, dans le cas de la SVD, les vecteurs propres dépendent des données, il est donc important de les conserver pour pouvoir reconstruire le signal. La figure FIG 6.6 présente l'évolution du PSNR en fonction du gain de compression. Nous présentons en FIG 6.6(a), les résultats sur une grille de 8×8 pixels et en FIG 6.6(b) pour une grille de 16×16 pixels.

Pour ces différents résultats, on peut constater que les bases de polynômes A-complètes donnent des résultats plus intéressants que la SVD en terme de qualité d'approximation. On peut donc en conclure que l'approximation par grille régulière permet d'obtenir une bonne approximation d'une image. Cependant, nous avons vu au chapitre 4 que la grille régulière est une technique très limitée pour l'analyse du domaine. Nous souhaitons donc obtenir dans la section suivante une modélisation hiérarchique du domaine spatial.

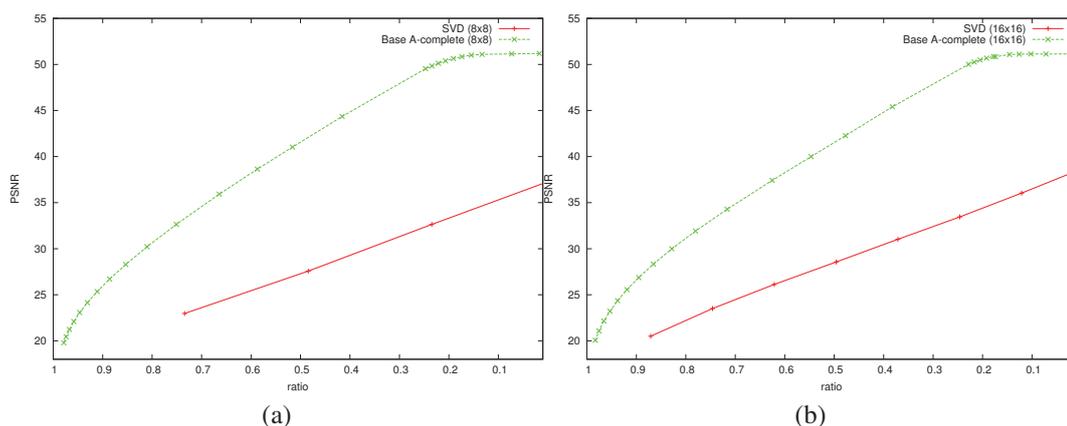


FIGURE 6.6 – Evolution du PSNR en fonction du gain de compression pour une base A-complète et une SVD toutes les deux appliquées sur une grille régulière ; (a) Résultats obtenus avec une grille composée de blocs de 8×8 pixels ; (b) Résultats obtenus avec une grille composée de blocs de 16×16 pixels

6.3.3 Multi-résolutions

Nous proposons de construire une analyse multi-résolutions fondée sur les coefficients polynomiaux obtenus par projection des blocs de pixels sur une base A-complète.

6.3.3.1 Réorganisation des coefficients en imagettes

Afin d’obtenir une modélisation multi-résolutions, nous réorganisons les coefficients issus de l’approximation. Une fois que tous les blocs ont été projetés sur la base A-complète de dimension choisie, nous obtenons un nombre de blocs dépendants des dimensions de l’image et des blocs choisis.

Pour un polynôme donné de la base, par exemple le polynôme $P_{0,0}(x)$, nous regroupons tous les coefficients associés à ce polynôme dans les différents blocs obtenus après projection. En répétant cette étape pour tous les polynômes de la base A-complète, on obtient alors une représentation des différents coefficients de chaque polynôme de la base que l’on appelle imagette. Pour des blocs de $n \times n$ pixels, nous obtenons $n \times n$ imagettes correspondantes aux $n \times n$ polynômes de la base. La figure FIG 6.7 illustre cette technique avec une image synthétique composée de 4×4 pixels. La grille utilisée est composée de blocs de 2×2 pixels, il y a donc 4 blocs. Ces blocs sont notés $b^{0,0}$ pour le bloc en haut à gauche, $b^{0,1}$ pour le bloc en haut à droite, $b^{1,0}$ pour le bloc en bas à gauche, $b^{1,1}$ pour le bloc en bas à droite. La base de polynômes A-complète est constituée des 4 polynômes suivants : $P_{0,0}$, $P_{0,1}$, $P_{1,0}$ et $P_{1,1}$. Nous notons donc $U_{0,0}^{0,1}$ le coefficient obtenu par projection du bloc $b^{0,1}$ sur le polynôme $P_{0,0}$. Il est à noter que les coefficients $P_{0,1}$ et $P_{1,0}$ permettent de “calculer” les dérivées selon x_1 et x_2 du bloc concerné.

La figure FIG 6.8 présente le résultat de la réorganisation des coefficients polynomiaux obtenus par projection des blocs de l’image sur une base polynomiale A-complète. En FIG 6.8(a), se trouve l’image Lena, en FIG 6.8(b), le résultat avec des blocs de 2×2 pixels et en FIG 6.8(c), avec des blocs de 8×8 pixels. On constate que la décomposition d’une image avec une base de polynômes A-complète et une grille de 2×2 pixels est équivalente à une décomposition avec l’ondelette de Haar. En effet, en étudiant les filtres utilisés pour l’ondelette de Haar et les

coefficients polynomiaux obtenus pour une base A-complète sur un domaine 2×2 pixels, on obtient les mêmes filtres à un facteur de normalisation près.

Il est possible de recommencer le processus sur l'imagette de chaque polynôme de la base pour effectuer une analyse multi-résolutions. Cependant, en fonction de la taille des blocs utilisés, le nombre de décompositions est limité. En effet, pour une image de $N \times N$ pixels, si les blocs sont de dimension $n \times n$ pixels, les imagettes obtenues par regroupement des coefficients sont de dimension $\frac{N}{n} \times \frac{N}{n}$ pixels.

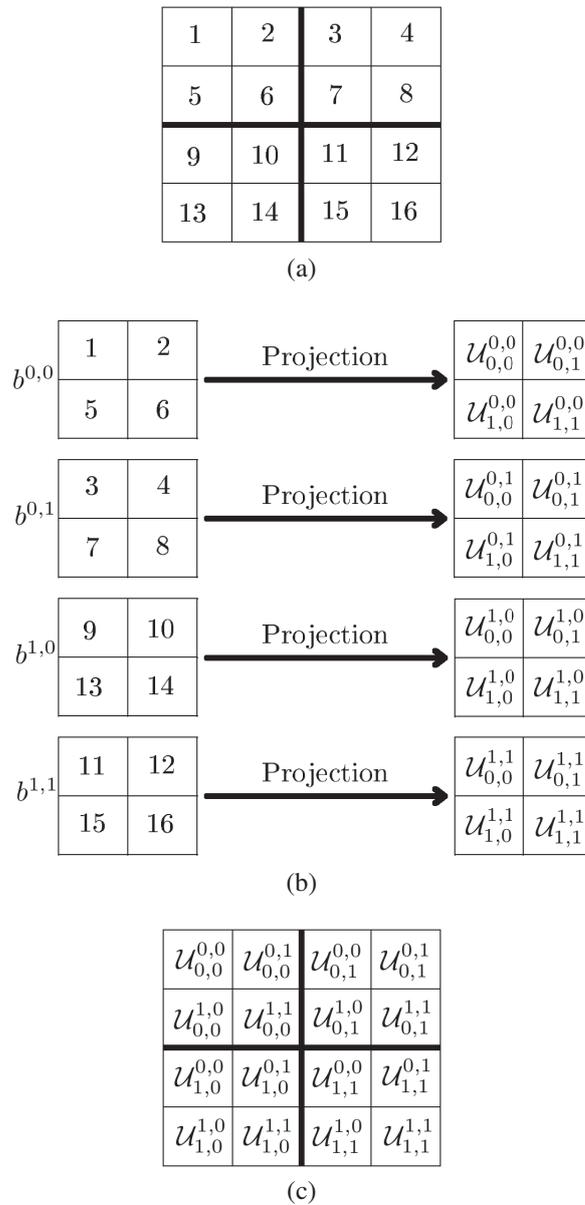


FIGURE 6.7 – Construction du premier niveau de l'analyse d'une image en niveau de gris avec une base polynomiale A-complète ; (a) petite image synthétique de 4×4 pixels ; (b) Projection de chaque bloc sur la base de polynômes A-complète de 2×2 polynômes ; (c) Regroupement des coefficients polynomiaux en imagettes.



FIGURE 6.8 – (a) Exemple de regroupement des coefficients issus de la projection dans un même élément de la base pour des bases A-complètes de 4 éléments (bloc de 2×2 pixels). (b) Exemple de regroupement des coefficients polynomiaux en imgettes avec une base A-complète de 64 éléments (bloc de 8×8 pixels) ; Pour l’affichage, les images (a) et (b) ont été égalisées.

6.3.3.2 Décomposition dyadique

Nous proposons donc un schéma permettant d’obtenir une décomposition dyadique de l’image. D’une échelle à l’autre, les dimensions de l’image sont divisées par 2 selon l’horizontale et la verticale. À partir des imgettes obtenues par regroupement des coefficients polynomiaux, on reconstruit une image de plus faible résolution. Pour cela, nous utilisons uniquement les $(\frac{n}{2} + 1) \times (\frac{n}{2} + 1)$ imgettes correspondantes aux coefficients de plus faible degré. Afin de construire cette image de dimension $\frac{N}{2} \times \frac{N}{2}$, il faut générer une deuxième base A-complète composée de $\frac{n}{2} \times \frac{n}{2}$ polynômes, ayant pour support spatial des blocs de $\frac{n}{2} \times \frac{n}{2}$ pixels. Cette base est appelée base de niveau 2. Afin de faciliter les explications, la base de polynômes A-complète composée de $n \times n$ polynômes est maintenant appelée base de niveau 1 lorsque cela est nécessaire. Cette décomposition dyadique nous permet d’obtenir une image plus petite, sur laquelle on peut alors appliquer à nouveau la décomposition. Cela permet d’obtenir une décomposition multi-résolutions d’une image à partir des bases de polynômes A-complètes. Nous illustrons cette méthode avec la figure FIG 6.9.

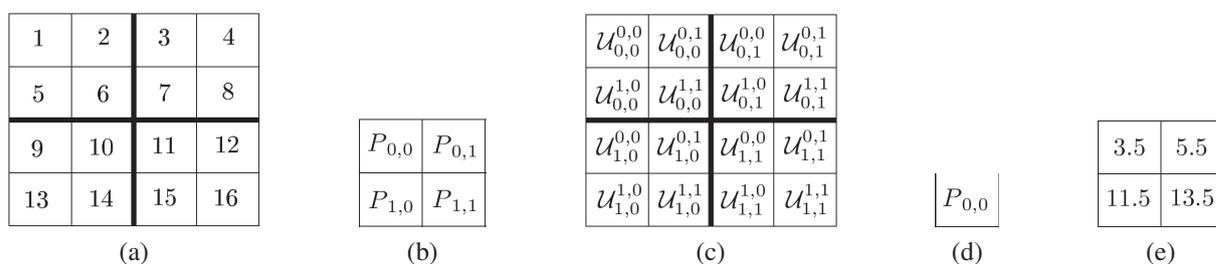


FIGURE 6.9 – Représentation des coefficients conservés d’une échelle à la suivante ; (a) Image synthétique de 4×4 pixels ; (b) Base polynomiale A-complète de niveau 1 ; (c) Représentation des coefficients polynomiaux regroupés en imgettes ; (d) Base polynomiale A-complète de niveau 2 ; (e) Image obtenue par décomposition dyadique à la résolution 1 à partir des coefficients de (c) et de la base A-complète de niveau 2.

Nous présentons un exemple de décomposition dyadique sur deux échelles, appliquée à l'image "bikes" sur la figure FIG 6.10.



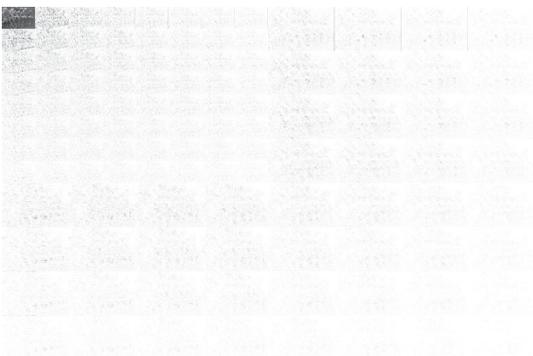
(a)



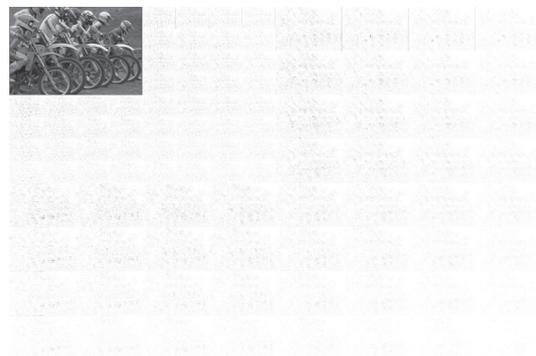
(b)



(c)



(d)



(e)

FIGURE 6.10 – Décomposition dyadique de l'image "bikes" par base de polynômes A-complète de 8×8 polynômes ; (a) Image "bikes" originale (échelle 0) ; (b) Regroupement en imasettes des coefficients polynomiaux ; (c) Décomposition à l'échelle 1 ; (d) Regroupement en imasettes des coefficients polynomiaux à partir de l'image à l'échelle 1 ; (e) Décomposition à l'échelle 2 ;

6.3.3.3 Reconstruction à l'échelle supérieure

À partir de la décomposition dyadique à une résolution donnée, il est alors possible de reconstruire l'image à la résolution supérieure. Pour la reconstruction, les blocs de l'image sont alors projetés sur la base A-complète de niveau 2. À partir des $\frac{n}{2} \times \frac{n}{2}$ coefficients obtenus,

nous souhaitons reconstruire les blocs avec la base de niveau 1. Si l'image est reconstruite uniquement à partir de ces coefficients, l'image obtenue ne contient pas tous les détails. Nous devons ajouter les coefficients polynomiaux manquants. Ces coefficients ont été stockés dans les imagerie lors de la décomposition. Pour chaque bloc, il est donc possible de retrouver les coefficients polynomiaux manquants dans les imagerie. De cette manière, il est possible d'obtenir une reconstruction parfaite de l'image originale. Sur la figure FIG 6.11, nous présentons la reconstruction de l'image "bikes" à partir du résultat de la décomposition dyadique sur deux échelles de l'image "bike" (cf. figure FIG 6.10(c)). La base A-complète de niveau 1 est une base composée de 8×8 polynômes.

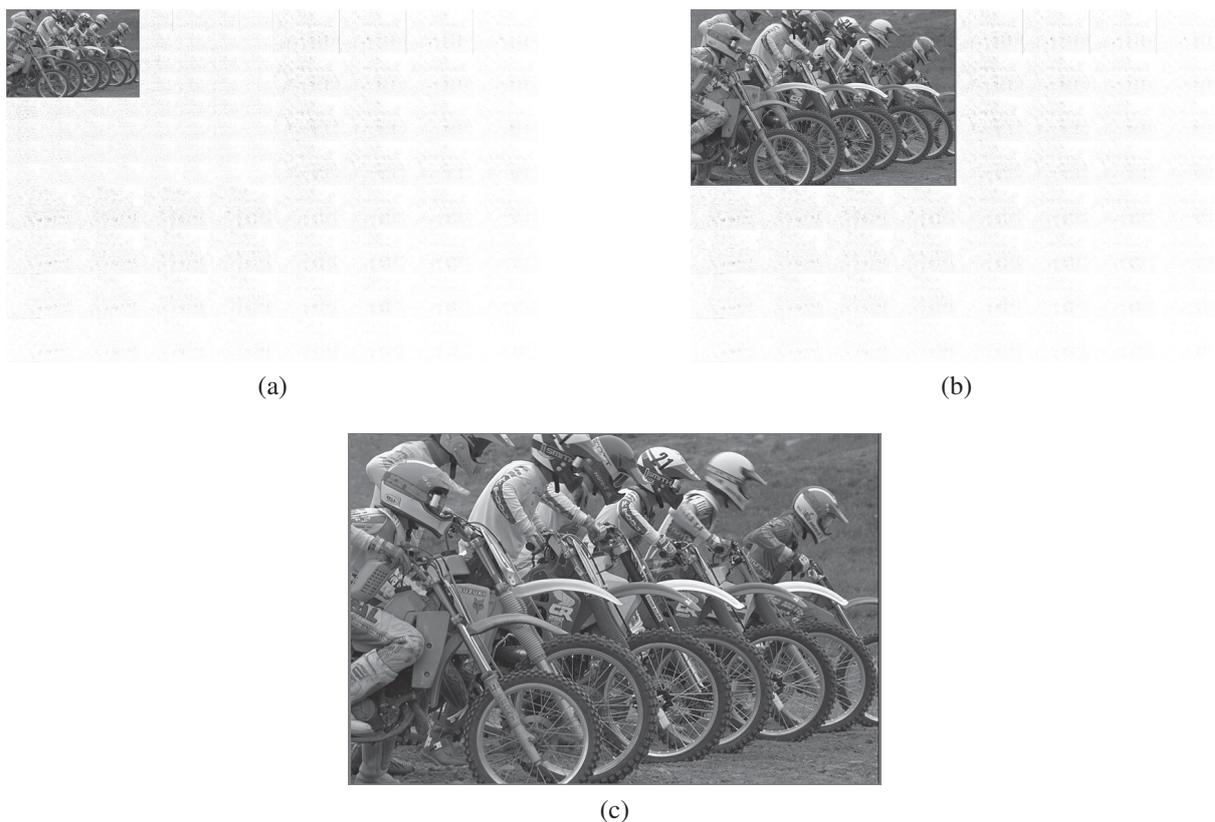


FIGURE 6.11 – Reconstruction dyadique de l'image "bikes" par base de polynômes A-complète à partir du résultat obtenu en 6.10; (a) Décomposition dyadique à l'échelle 2 de l'image "bikes" par base polynomiale A-complète; (b) Reconstruction à l'échelle 1; (c) Reconstruction à l'échelle 0.

Nous constatons donc que nous pouvons obtenir une analyse multi-résolutions dyadique fondée sur l'utilisation des bases de polynômes A-complètes. Nous souhaitons utiliser les bases polynomiales A-complètes pour compresser l'information. Nous proposons de reconstruire l'image à partir de la résolution la plus élevée (image de plus petite dimension) sans utiliser les coefficients de degrés supérieurs à $\frac{n}{2} \times \frac{n}{2}$. On peut considérer qu'il s'agit d'interpoler les pixels de l'image obtenue à la plus haute résolution jusqu'à obtenir l'image à la résolution d'origine. Pour ces tests, nous utilisons trois images de la base de test "LIVE". Ces trois images sont présentées sur la figure FIG 6.12. Pour chacune de ces images, nous calculons l'erreur quadratique moyenne et le gain de compression τ (cf. équation (6.32)). Les graphiques de la figure FIG 6.13 présentent les résultats obtenus. Ces graphiques représentent l'évolution du gain

de compression τ sur les axes des abscisses et l'erreur quadratique moyenne sur les axes des ordonnées. Pour chaque image, les tests sont effectués pour des bases de niveau 1 de 4×4 , 8×8 , 16×16 , 32×32 polynômes. Les bases de niveau 2 sont alors respectivement de 2×2 , 4×4 , 8×8 , 16×16 polynômes.



(a)



(b)



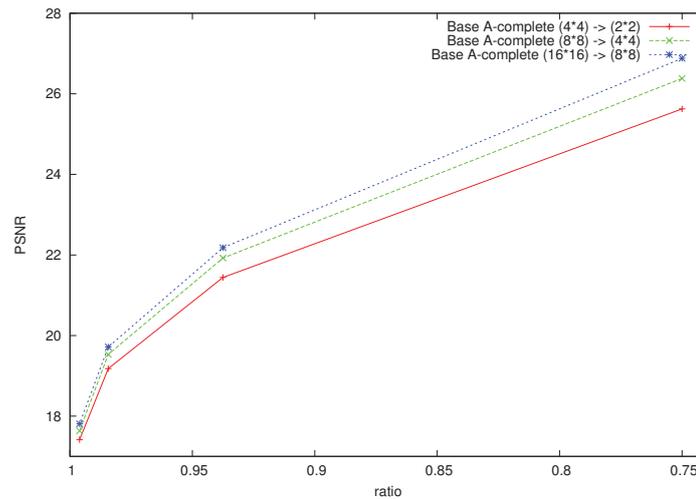
(c)

FIGURE 6.12 – Images tests issues de la base LIVE ; (a) Image "bikes" ; (b) Image "lighthouse2" ; (c) Image "parrots".

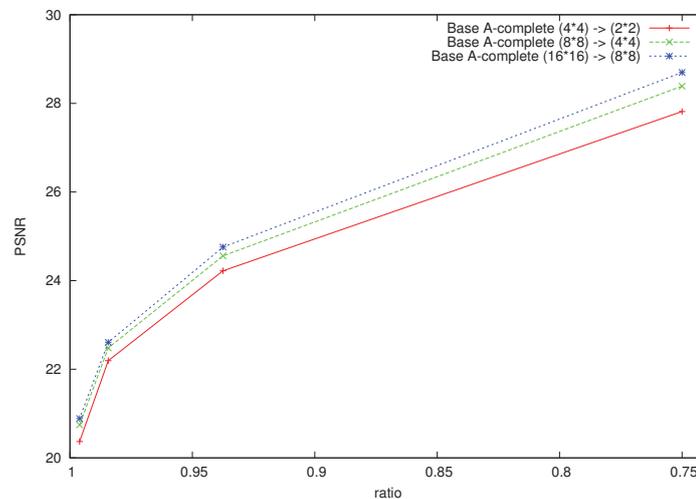
Afin d'avoir une représentation visuelle des défauts générés par cette méthode, nous présentons sur la figure FIG 6.14 les images obtenues à partir de la décomposition dyadique de l'image "parrots". Pour cet exemple, chacune des résolutions (de 1 à 4) est décomposée avec une grille composée de blocs de 8×8 pixels. On constate qu'un effet de bloc apparaît en raison de la suppression des polynômes de plus haut degré qui modélisent les détails. La taille des blocs de ce défaut augmente avec la résolution. Si la base de niveau 1 est de $n \times n$ polynômes, à la résolution r (pour $r > 0$), les blocs visibles sont de $(n \times 2^{r-1}) \times (n \times 2^{r-1})$ pixels.

6.3.3.4 Décomposition multi-résolutions non-dyadique

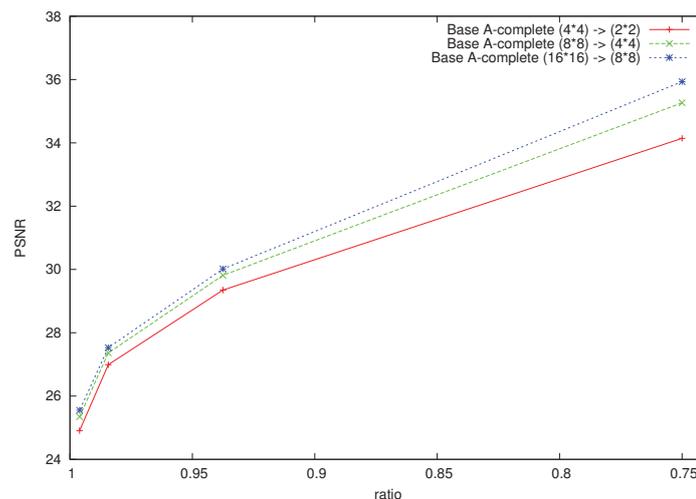
Le cas dyadique n'est qu'un cas particulier. Sur le même principe, on peut décider de reconstruire une image en conservant d'une échelle à l'autre un nombre de polynômes compris entre 1 et $n \times n$. En effet, les dimensions de l'image de faible résolution que l'on souhaite obtenir d'une échelle à l'autre dépendent du nombre de points de collocation et donc du nombre de polynômes utilisés.



(a)



(b)



(c)

FIGURE 6.13 – Evolution du PSNR en fonction du gain de compression pour une décomposition multi-résolutions dyadique par base de polynômes A-complète. Les bases de niveau 1 sont de 4×4 , 8×8 , 16×16 , 32×32 polynômes. Les bases de niveau 2 sont alors respectivement de 2×2 , 4×4 , 8×8 , 16×16 polynômes. (a) Graphique obtenu pour l'image "bikes"; (b) Graphique obtenu pour l'image "lighthouse2"; (c) Graphique obtenu pour l'image "parrots".



(a)



(b)



(c)



(d)



(e)

FIGURE 6.14 – Visualisation de la qualité de l'image parrots en fonction de la résolution de l'approximation par décomposition dyadique ; (a) Image "parrots" originale ; (b) Image obtenue à la résolution 1 ; (c) Image obtenue à la résolution 2 ; (d) Image obtenue à la résolution 3 ; (e) Image obtenue à la résolution 4 ;

Il est possible d'obtenir une décomposition non-dyadique à partir du nombre de polynômes qui composent la base de niveau 2. C'est en effet cela qui détermine la dimension des blocs et donc de l'image d'une échelle à l'autre. En reconstruisant une image à partir des $(n-1) \times (n-1)$ coefficients de plus faible degré, il est possible d'obtenir une décomposition avec plus d'échelles que la décomposition dyadique. Les dimensions d'une image ne sont pas nécessairement des multiples de n et de $n-1$. Dans ce cas, il faut gérer les conditions aux bords d'une échelle à l'autre. La figure FIG 6.15 présente un exemple de l'utilisation de la reconstruction d'une échelle à l'autre avec les 7×7 polynômes de plus faible degré pour des blocs de dimension 8×8 , sur l'image "bikes". Nous présentons les résultats obtenus à l'échelle 1 et 2.



(a)



(b)



(c)



(d)



(e)

FIGURE 6.15 – Décomposition de l'image "bikes" avec une base de niveau 1 de 8×8 polynômes et une base de niveau 2 de 7×7 polynômes ; (a) Image "bikes" originale ; (b) Imagerie à l'échelle 1 ; (c) Décomposition à l'échelle 1 ; (d) Imagerie à l'échelle 2 ; (e) Décomposition à l'échelle 2.

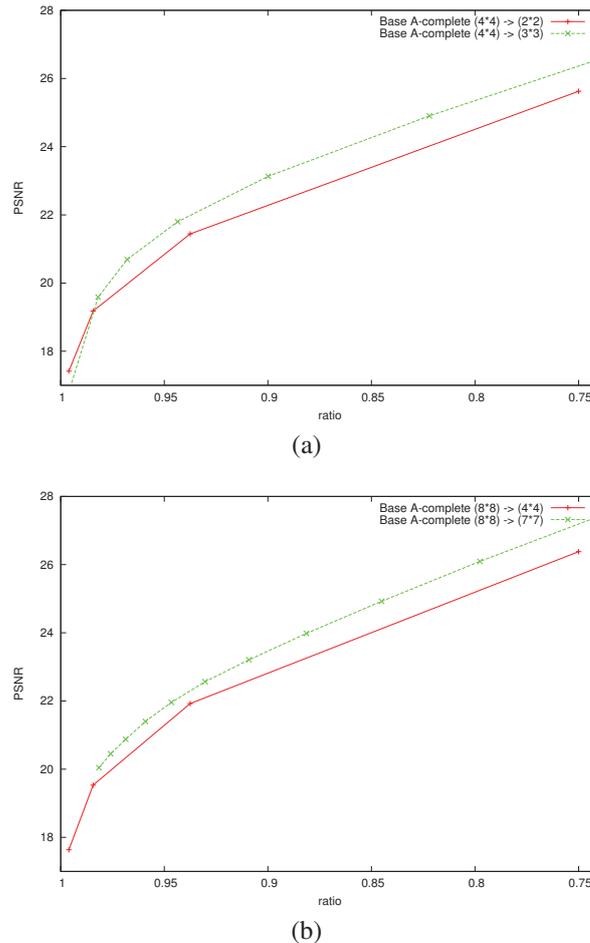
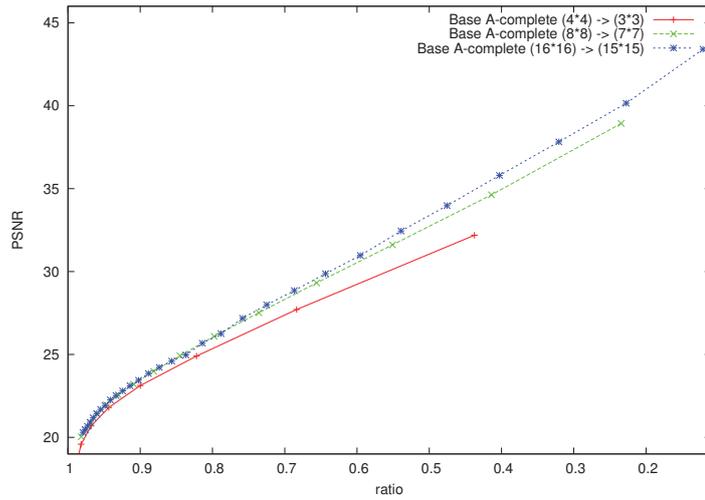


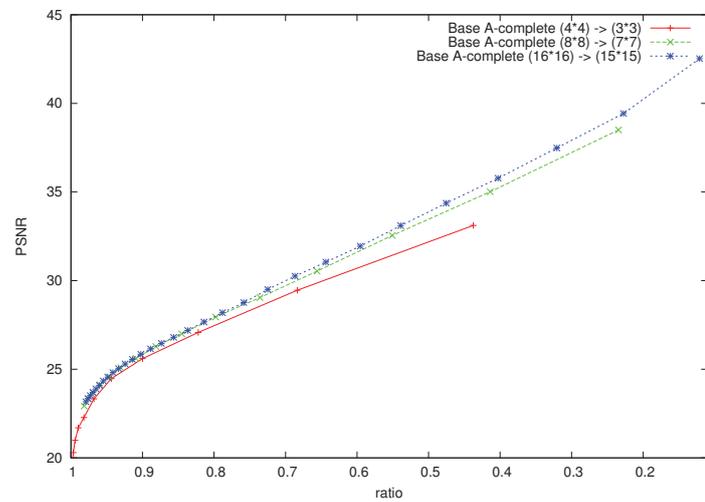
FIGURE 6.16 – Comparaison de l’évolution du PSNR en fonction du gain de compression entre une décomposition multi-résolutions dyadique et une décomposition multi-résolutions non-dyadique à partir de l’image “parrots”. (a) Graphique obtenu pour des bases de niveau 1 de 4×4 polynômes ; (b) Graphique obtenu pour des bases de niveau 1 de 8×8 polynômes ;

Afin de visualiser l’apport de la décomposition non-dyadique, nous comparons sur la figure FIG 6.16, les résultats obtenus avec une décomposition dyadique à ceux obtenus avec une décomposition non dyadique. Nous effectuons cette comparaison pour des blocs de 4×4 pixels en FIG 6.16(a) et de 8×8 pixels en FIG 6.16(b). On constate que pour un gain de compression donné, il est plus intéressant d’utiliser la décomposition non-dyadique afin d’obtenir une meilleure qualité d’approximation.

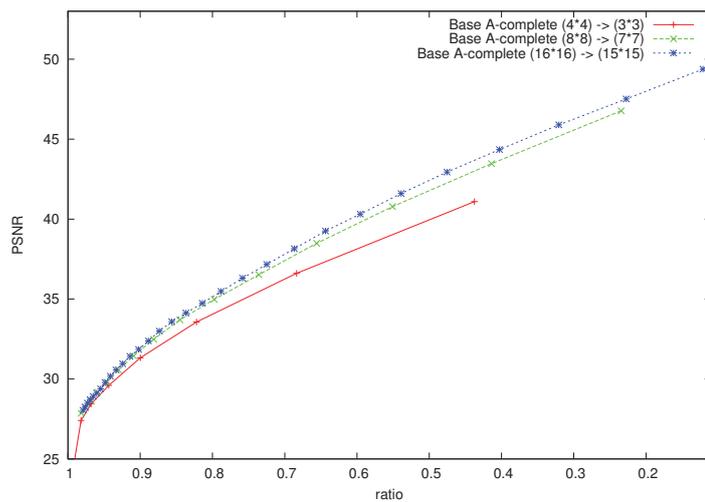
Nous présentons sur la figure FIG 6.17, pour chaque image, l’évolution du PSNR en fonction du gain de compression τ pour le cas non dyadique. Le graphique présenté en FIG 6.17(a) présente les résultats obtenus sur l’image “bikes”, en FIG 6.17(b) les résultats obtenus sur l’image “lighthouse” et en FIG 6.17(c) les résultats obtenus sur l’image “parrots”. Nous constatons qu’il est possible d’obtenir beaucoup plus de décompositions qu’avec la décomposition dyadique. Cela permet d’ajouter plus de scalabilité au gain de compression du modèle. Nous constatons aussi que le choix de la dimension des blocs a une influence sur la qualité de l’approximation. Cependant, pour des gains de compression proches de 1, ce choix semble être moins important. Plus les dimensions des blocs sont grandes, plus il est possible d’obtenir de résolutions. De plus, pour un gain de compression donné, la qualité d’approximation est



(a)



(b)



(c)

FIGURE 6.17 – Evolution du PSNR en fonction du gain de compression pour une décomposition multi-résolutions dyadique par base de polynômes A-complète. Les bases de niveau 1 sont de 4×4 , 8×8 , 16×16 , 32×32 polynômes. Les bases de niveau 2 sont alors respectivement de 2×2 , 4×4 , 8×8 , 16×16 polynômes. (a) Graphique obtenu pour l'image "bikes"; (b) Graphique obtenu pour l'image "lighthouse2"; (c) Graphique obtenu pour l'image "parrots".

meilleure avec des grand blocs. Cependant, il est important de noter que le surcoût calculatoire entre une grille composée de blocs de 8×8 et une grille composée de blocs de 16×16 est non négligeable.

6.4 Application aux champs de déplacements

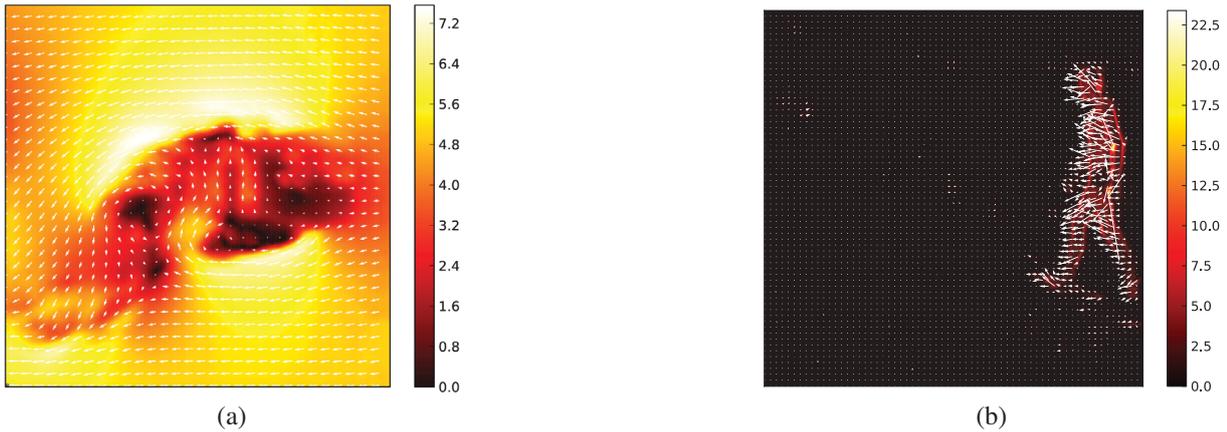
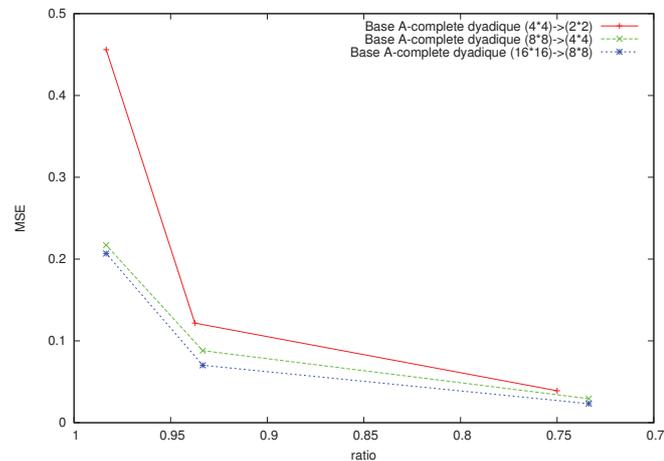


FIGURE 6.18 – Présentation des champs tests ; (a) Champ 131 de la séquence Cylindre-carré (cf. ANNEXE A ;(b) Champ Marche.

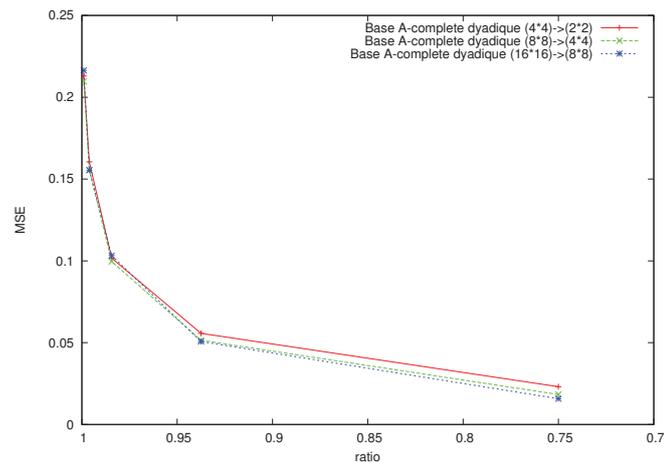
Nous avons étudié les bases de polynômes A-complètes sur des images en niveau de gris afin de comparer ces bases avec la DCT et la SVD sur des exemples plus visuels que les champs de déplacements. Nous utilisons maintenant les bases A-complètes sur des champs de déplacements comme nous l’avons fait dans les chapitres précédents. Pour les différents tests, nous utilisons deux champs tests de nature différente. Le premier champ est le champ 131 de la séquence “cylindre-carré” qui, nous le rappelons, est issu d’une expérience de mécanique des fluides expérimentale. Il est présenté sur la figure FIG 6.18(a). Ce champ est obtenu par PIV (vélocimétrie par imagerie de particules cf. chapitre 2.1). Le deuxième champ est un champ destiné à l’étude du mouvement humain. Il s’agit du mouvement d’une personne qui marche. Le champ est obtenu par flot optique avec l’algorithme de HORN et SHUNCK. Nous appelons ce champ “Marche”. Il est présenté sur la figure FIG 6.18(b).

Comme pour les images, il est possible d’obtenir une approximation à erreur nulle d’un champ de déplacements grâce aux bases A-complètes utilisées avec l’une des méthodes proposées. Nous nous intéressons à la qualité d’approximation en fonction du nombre de polynômes conservés. Nous présentons l’erreur d’approximation par la mesure de l’erreur quadratique moyenne telle qu’elle est présentée aux chapitre 3, 4 et 5. Nous nous intéressons au cas d’une approximation dyadique. Pour une grille composée de blocs de $n \times n$ vecteurs, soit une base A-complète de $n \times n$ polynômes, à chaque résolution, nous ne conservons donc que les $\frac{n}{2} \times \frac{n}{2}$ polynômes de plus faible degré. Le graphique de la figure FIG 6.19 présente les résultats obtenus. Ce graphique représente l’évolution de l’erreur quadratique moyenne en fonction du gain de compression. Ici la base A-complète de niveau 1 est composée de 8×8 polynômes et la base de niveau 2 est composée de 4×4 polynômes. Nous présentons sur la FIG 6.19(a), les résultats obtenus sur le champ 131 de la séquence cylindre-carré et sur la figure FIG 6.19(b), les résultats obtenus sur le champ “Marche”. En raison des dimensions du champ “cylindre-carré”

(124×124 vecteurs), il n'est pas intéressant de calculer plus de 3 résolutions. On constate que le gain de compression correspond à 0.75^{res} où res est la résolution. L'évolution trop rapide du gain de compression d'une résolution à l'autre produit une dégradation importante de la qualité d'approximation, comme nous l'avons constaté sur les images dans la section précédente.



(a)



(b)

FIGURE 6.19 – Evolution de l'erreur quadratique moyenne en fonction du gain de compression pour la décomposition dyadique d'un champ de déplacements ; (a) Résultats pour le champ 131 de la séquence cylindre-carré ; (b) Résultats pour le champ "Marche"

Afin de pouvoir obtenir plus de résolutions, nous utilisons la version non-dyadique de la méthode multi-résolutions. Nous présentons les résultats obtenus sur la **FIGURE.6.20** avec en (a) les résultats avec le champ 131 de la séquence cylindre-carré et en (b) avec le champ "Marche". Nous utilisons des bases A-complètes de niveau 1 composées de 4×4 , 8×8 , 16×16 polynômes. Les bases de niveau 2 sont alors composées de 3×3 , 7×7 , 15×15 polynômes.

Nous pouvons ainsi montrer l'intérêt de la décomposition multi-résolutions dans le cas non-dyadique, puisque celle-ci permet un plus grand nombre de résolutions. Cela nous permet de mieux gérer le rapport entre le gain de compression et l'erreur quadratique moyenne. Dans la section suivante, nous comparons ces résultats à ceux obtenus aux chapitres précédents.

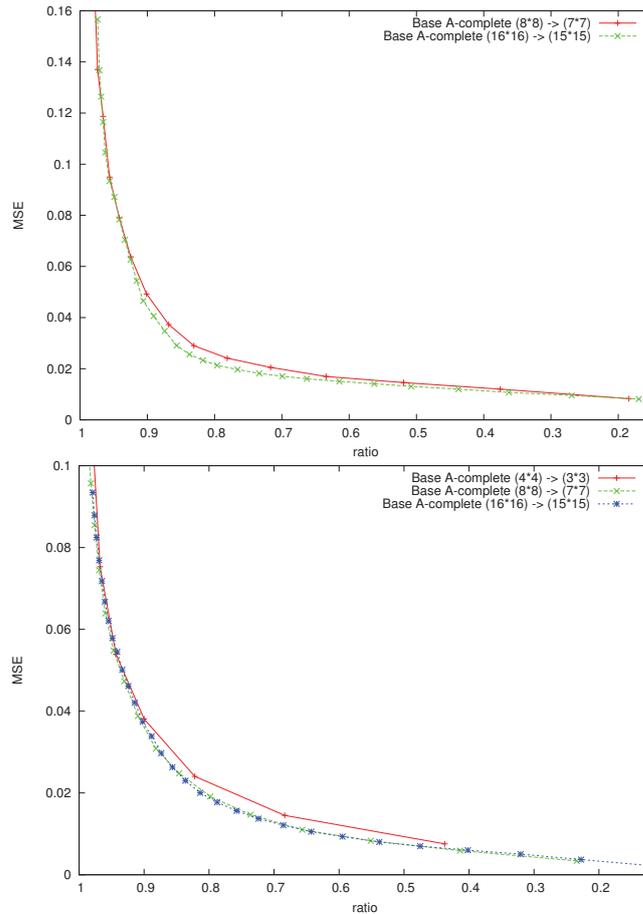


FIGURE 6.20 – Evolution de l’erreur quadratique moyenne en fonction du gain de compression pour la décomposition dyadique d’un champ de déplacements ; (a) Résultats pour le champ 131 de la séquence cylindre-carré ; (b) Résultats pour le champ “Marche”

6.5 Comparaison de l’approximation par base A-complète aux méthodes précédentes

Nous proposons dans le tableau **TAB.6.4**, une synthèse des différents résultats d’approximation obtenus au cours des chapitres précédents sur le champ 131 de la séquence cylindre-carré. Ce tableau regroupe les résultats du tableau **TAB.4.2** auxquels nous ajoutons un exemple de décomposition dyadique et deux exemples de décompositions non-dyadiques. Pour la décomposition dyadique, nous arrêtons la décomposition à la résolution 3 en raison des dimensions du champ. La base de polynômes A-complète de niveau 1 utilisée est composée de 8×8 polynômes et la base de niveau 2 de 4×4 polynômes. Pour les décompositions non-dyadiques, utilisons d’abord, une base de polynômes A-complète de niveau 1 composée de 8×8 polynômes et la base de niveau 2 de 7×7 polynômes. Dans ce cas, nous arrêtons la décomposition à la résolution 11 puisque le résultat obtenu est comparable, au niveau du gain de compression et de l’erreur quadratique moyenne, à celui avec un quadtree ou un kd-tree et une base de polynômes dense de degré 1. Ensuite, nous utilisons une base de polynômes A-complète de niveau 1 composée de 16×16 polynômes et la base de niveau 2 de 15×15 polynômes. En arrêtant la décomposition à la résolution 23, le résultat est à nouveau comparable, au niveau du gain de compression et de

6.5. Comparaison de l'approximation par base A-complète aux méthodes précédentes 117

l'erreur quadratique moyenne, à celui avec un quadtree ou un kd-tree et une base de polynômes dense de degré 1.

Méthode	Degré	Niveau	Gain	EQM	E_θ
globale	5	0	99.863	1.167	0.210
globale	9	0	99.642	0.631	0.112
globale	16	0	99.005	0.396	0.080
globale	20	0	98.498	0.308	0.069
grille régulière	9	3	77.107	0.020	0.005
grille régulière	5	4	65.036	0.014	0.004
grille régulière	1	5	80.021	0.041	0.011
quadtree	9	3	87.838	0.034	0.007
quadtree	5	4	87.981	0.036	0.008
quadtree	1	5	92.254	0.072	0.015
kd-tree median	9	3	91.058	0.040	0.008
kd-tree median	5	4	91.669	0.045	0.009
kd-tree median	1	5	92.683	0.081	0.015
kd-tree irrégulier	9	3	92.488	0.053	0.009
kd-tree irrégulier	5	4	92.215	0.050	0.009
kd-tree irrégulier	1	5	96.449	0.135	0.032
boîte (global,boîte)	(16,1)	0	98.868	0.389	0.077
boîte (global,boîte)	(9,5)	0	98.803	0.532	0.079
Voronoi L1 (boîtes non orientées)	9	14 sites	94.992	0.305	0.043
Voronoi L1 (boîtes non orientées)	5	14 sites	98.088	0.517	0.065
Voronoi L1 (boîtes non orientées)	1	14 sites	99.727	1.170	0.151
Voronoi L2 (boîtes non orientées)	9	14 sites	94.992	0.562	0.044
Voronoi L2 (boîtes non orientées)	5	14 sites	98.088	0.824	0.071
Voronoi L2 (boîtes non orientées)	1	14 sites	99.727	1.692	0.149
Voronoi L2 (boîtes orientées)	9	14 sites	94.992	0.585	0.0475376
Voronoi L2 (boîtes orientées)	5	14 sites	98.088	0.821	0.076
Voronoi L2 (boîtes orientées)	3	14 sites	99.090	1.076	0.106
Voronoi L2 (boîtes orientées)	1	14 sites	99.727	1.609	0.155
Décomposition dyadique	8×8	rés = 3	98.335	0.217	0.059
Décomposition non-dyadique	8×8	rés = 11	92.482	0.064	0.017
Décomposition non-dyadique	16×16	rés = 23	94.147	0.078	0.021

TABLEAU 6.4 – Evaluation de l'erreur de modélisation et du gain de compression pour les méthodes globale, de subdivision par grille régulière, par quadtree, par kd-tree, par boîtes englobantes, par diagramme de Voronoï, par décomposition dyadique avec une base polynomiale A-complète et par décomposition non-dyadique avec une base polynomiale A-complète

Nous utilisons la méthode de décomposition non-dyadique dans le chapitre 7 pour la détection de points singuliers dans un champ de déplacements. Il est important de signaler qu'il est possible d'étendre les bases de polynômes A-complètes aux dimensions supérieures à 2.

6.6 Conclusion

Dans ce chapitre, nous avons étudié une méthode d'approximation à erreur nulle à partir des bases de polynômes. Pour cela, nous avons utilisé une famille particulière de polynômes. Cette méthode de modélisation permet par décimation de réaliser une compression d'images comparable à une DCT. De plus, cette méthode fournit des images de meilleure qualité (mesurée par le PSNR) que la SVD pour un même gain de compression. Finalement, nous avons proposé deux algorithmes multi-résolutions l'un dyadique l'autre non. L'algorithme dyadique est comparable à une décomposition en ondelette avec une ondelette de Haar dans le cas où l'on utilise une base de 2×2 polynômes. L'algorithme non-dyadique permet d'obtenir plus de résolutions et fournit donc une meilleure scalabilité. En ce qui concerne l'approximation d'un champ de déplacements, l'algorithme non-dyadique fournit des résultats comparables en terme de qualité et de gain de compression à ceux obtenus avec l'algorithme par quadtree ou par kd-tree.

Troisième partie

Applications

CHAPITRE 7

DÉTECTION ET CLASSIFICATION DES POINTS SINGULIERS

7.1 Introduction

Dans ce chapitre, nous étudions la détection de points singuliers dans les champs de vecteurs. La détection de structures singulières concerne un grand nombre de domaines de recherche : la localisation de phénomènes météorologiques [Memin 99], [Rougon 00], [Yang 00], la mécanique des fluides expérimentale pour analyser et comprendre les écoulements [Rao 92a], [Shu 93], [Ford 94], l'imagerie médicale pour l'analyse des flux sanguins [Nogawa 97], et l'industrie pour l'analyse de textures de bois ou de semi-conducteurs [Rao 92a].

La section 7.2 introduit le cadre mathématique dans lequel les points singuliers sont définis. La section 7.3 présente les principales méthodes de la littérature, étudie les liens entre ces méthodes ainsi que leurs spécificités et le contexte dans lequel elles ont été développées. Ensuite, dans la section 7.4, nous proposons de détecter et de classifier les singularités par une méthode de portrait de phase. Cette méthode a été introduite par RAO et JAIN en 1992 [Rao 92a]. Notre contribution consiste à utiliser une base orthogonale de polynômes bivariés pour estimer le modèle de mouvement affine dont dépendent la détection et la classification des structures.

À partir de ces travaux, nous proposons en section 7.4.2, trois algorithmes multi-échelles. Le premier repose sur l'utilisation d'un algorithme à fenêtre adaptative (fenêtre de dimensions variables). Le second utilise une fenêtre mobile non adaptative (fenêtre de dimensions fixes). Cette fenêtre est appliquée aux différentes résolutions du champ. Le troisième utilise la structure et l'algorithme du quadtree introduits au chapitre 4. Dans la section 7.4.3, nous présentons un algorithme qui permet de définir une zone autour d'une singularité dans laquelle un modèle polynomial est valide. Enfin, dans la section 7.5, nous présentons des résultats de ces travaux sur des champs de déplacements synthétiques et expérimentaux.

7.2 Caractérisation mathématique des points singuliers

Dans cette section, nous présentons le cadre mathématique des points singuliers d'un champ de déplacements. Le lecteur désirant plus de détails pourra se référer au manuscrit de thèse MARIE-ANGE DENIS-BROSSARD [Denis-Brossard 00].

7.2.1 Définitions

Soit M une variété compacte de l'espace euclidien \mathbb{R}^d .

Définition 5 Un point \vec{x} du champ \mathcal{C} et un point singulier si $\vec{x} \in M$ et $\mathcal{C}(\vec{x}) = 0$.

Définition 6 Un point non singulier est dit régulier.

Définition 7 Un point singulier $\vec{x} \in M$ du champ \mathcal{C} est dit simple si la différentielle du champ en \vec{x} est non singulière (n'admet aucune valeur propre nulle).

L'étude du spectre du Laplacien du champ \mathcal{C} permet de classer les points singuliers en deux catégories :

- les singularités hyperboliques ;
- les singularités non hyperboliques.

Définition 8 Un point singulier $\vec{x} \in M$ du champ \mathcal{C} est dit hyperbolique, si la différentielle du champ en \vec{x} n'admet aucune valeur propre imaginaire pure.

Le théorème de GROBMAN-HARTMAN [Anosov 88], [Palis 82b] permet de déterminer le comportement d'un champ \mathcal{C} au voisinage de points singuliers hyperboliques. On note $\mathcal{X}^r(M)$ est l'espace des champs de vecteurs de classe C^r .

Théorème 1 [Théorème de GROBMAN-HARTMAN] Soient $\mathcal{C} \in \mathcal{X}^r(M)$, un champ possédant uniquement des points singuliers hyperboliques, et $\vec{x} \in M$ un tel point singulier. Au voisinage de \vec{x} , le champ \mathcal{C} est topologiquement équivalent (i.e. homéomorphe) à sa partie linéaire.

Un deuxième théorème permet quant à lui de définir le comportement d'un champ au voisinage des points réguliers [Anosov 88].

Théorème 2 [Aspect local au voisinage d'un point régulier] Soient $\mathcal{C} \in \mathcal{X}^r(M)$ un champ de vecteurs et $\vec{x} \in M$ un point régulier de V . Soit \mathcal{C}_c le champ de vecteurs constant sur M défini par $\mathcal{C}_c = (1, 0, \dots, 0)^T$, $\vec{x} \in M$. Il existe un difféomorphisme ϕ sur un voisinage de \vec{x} tel que $\phi \circ \mathcal{C} = \mathcal{C}_c$.

Les théorèmes 1 et 2 nous permettent de mettre en évidence l'importance des points singuliers dans l'étude d'un champ de déplacements. En effet, ces théorèmes nous assurent que la topologie d'un champ de vecteurs régulier est entièrement déterminée par ses points singuliers. De plus, ils affirment qu'au voisinage d'un point singulier $\vec{x} \in M$, un champ régulier \mathcal{C} est localement homéomorphe à un champ affine, correspondant au champ tangent à \mathcal{C} en \vec{x} . Cela signifie que la structure locale d'un champ régulier peut être complètement spécifiée à partir d'une caractérisation topologique exhaustive des champs affines.

En s'appuyant sur des résultats classiques de topologie algébrique [Anosov 88], on peut affirmer qu'au voisinage d'un point singulier, tout champ affine est localement homéomorphe à un représentant canonique parmi un nombre fini de prototypes appelés portraits de phase.

7.2.2 Portraits de phase

Soit $\mathcal{C}(\vec{x})$ où $\vec{x} = (x_1, x_2, \dots, x_d)^T$ un champ de déplacements à d composantes tel que :

$$\mathcal{C}(\vec{x}) = A\vec{x} + b \quad (7.1)$$

où b représente la translation pure et A le tenseur de déformation linéaire associé au champ. Dans le cas où A est inversible, ce champ admet un unique point singulier défini en \vec{x}_c par

$$\vec{x}_c = -A^{-1}b \quad (7.2)$$

Il est important de noter que le point \vec{x}_c peut se trouver en dehors du support Ω .

Le théorème de GROBMAN-HARTMAN assure que la structure autour de \vec{x}_c est complètement déterminée par A . Il existe deux approches algébriques équivalentes permettant d'obtenir une caractérisation exhaustive de cette structure locale : la jordanisation de la matrice A et la décomposition canonique tensorielle de A . Nous présentons ces deux approches.

7.2.2.1 Approche par jordanisation de la matrice A

La forme de Jordan de la matrice A est obtenue en étudiant les racines $(\lambda_i)_{i \in [1..d]} \in \mathbb{C}$ (supposées être des nombres réels et vérifier que $\lambda_1 \geq \dots \geq \lambda_d$) de son polynôme caractéristique \mathcal{P}_A , défini par

$$\mathcal{P}_A(\lambda) = \det(A - \lambda \mathbb{I}_d) = \sum_{0 \leq i \leq d} p_i \lambda^i \quad (7.3)$$

où \mathbb{I}_d est la matrice identité de dimension $d \times d$.

Ces racines sont les valeurs propres de la matrice A et sont reliées aux coefficients $(p_i)_{i \in [1..d]}$ du polynôme \mathcal{P}_A via les polynômes symétriques élémentaires de degré d [Ramis 83]

$$\frac{p_{d-i}}{p_d} = (-1)^i \sum_{1 \leq j_1 < \dots < j_i \leq d} \lambda_{j_1} \dots \lambda_{j_i} \quad (7.4)$$

Les racines λ_i et les coefficients de \mathcal{P}_A sont des invariants algébriques. Il est possible de déterminer le caractère réel ou complexe d'une matrice à partir d'un invariant algébrique de A , le discriminant. Ce discriminant peut se définir à partir de la notion de résultant algébrique.

Définition 9 Soient $\mathcal{P}(X)$ défini en équation (7.5) et $\mathcal{Q}(X)$ défini en équation (7.6) des polynômes à coefficients réels ou complexes de degrés respectifs m et n .

$$\mathcal{P}(X) = \sum_{0 \leq i \leq m} p_i X^i \quad (7.5)$$

$$\mathcal{Q}(X) = \sum_{0 \leq i \leq n} q_i X^i \quad (7.6)$$

On appelle résultant de \mathcal{P} et \mathcal{Q} , l'élément $\mathcal{R}(\mathcal{P}, \mathcal{Q})$ défini par

$$\mathcal{R}(\mathcal{P}, \mathcal{Q}) = \det \begin{bmatrix} p_m & 0 & \dots & 0 & q_n & 0 & \dots & \dots & 0 \\ p_{m-1} & p_m & \ddots & \vdots & q_{n-1} & q_n & \ddots & & \vdots \\ \vdots & p_{m-1} & \ddots & 0 & \vdots & q_{n-1} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & p_m & q_1 & \vdots & & \ddots & 0 \\ p_1 & \vdots & & p_{m-1} & q_0 & q_1 & & & q_n \\ p_0 & p_1 & & \vdots & 0 & q_0 & & & q_{n-1} \\ 0 & p_0 & \ddots & \vdots & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & p_0 & p_1 & \vdots & & \ddots & \ddots & q_1 \\ 0 & \dots & 0 & p_0 & 0 & \dots & \dots & 0 & q_0 \end{bmatrix} \quad (7.7)$$

Soit $\mathcal{P}(X)$ un polynôme de degré $m \geq 2$. On appelle discriminant de \mathcal{P} le résultant de \mathcal{P} et de son polynôme dérivé \mathcal{P}' défini par

$$\Delta(\mathcal{P}) = \mathcal{R}(\mathcal{P}, \mathcal{P}') \quad (7.8)$$

Le discriminant d'une matrice carrée A est le discriminant de son polynôme caractéristique :

$$\Delta(A) = \Delta(\mathcal{P}_A) \quad (7.9)$$

L'étude du signe du discriminant de la matrice A nous permet de déterminer le caractère réel ou complexe des racines de A . En effet, analysons les trois cas possibles :

- $\Delta(A) > 0$: les valeurs propres de A sont réelles et distinctes ;
- $\Delta(A) = 0$: la matrice A admet des valeurs propres réelles, dont 2 au moins sont égales ;
- $\Delta(A) < 0$: la matrice A admet au moins 2 valeurs propres conjuguées.

Nous détaillons la jordanisation de la matrice A dans le cas d'un champ affine de dimension 2. Une description semblable pour le cas d'un champ affine en dimension 3 est présentée dans [Denis-Brossard 00].

En dimension 2, le polynôme caractéristique de A s'exprime sous la forme

$$\mathcal{P}_A = \lambda^2 + P\lambda + Q \quad (7.10)$$

où P est exprimé par la trace de A :

$$P = -\text{tr}(A) = -(a_{11} + a_{22}) \quad (7.11)$$

et Q par le déterminant de A :

$$Q = \det(A) = a_{11}a_{22} - a_{21}a_{12} \quad (7.12)$$

Le discriminant de A est alors défini par

$$\Delta(A) = P^2 - 4Q \quad (7.13)$$

Les racines de \mathcal{P}_A peuvent s'écrire sous la forme analytique suivante

$$\lambda_{1,2} = \frac{1}{2}(tr(A) \pm \sqrt{\Delta(A)}) \quad (7.14)$$

L'énumération exhaustive des formes de Jordan canoniques associées aboutit aux portraits de phase affines $2D$ présentés dans le tableau TAB 7.1.

7.2.2.2 Approche par décomposition tensorielle canonique

Tout tenseur d'ordre 2 peut être décomposé en la somme d'un tenseur symétrique D de trace non nulle, d'un tenseur symétrique A^+ de trace nulle et d'un tenseur antisymétrique A^- :

$$A = D + A^+ + A^- \quad (7.15)$$

avec

$$D = \frac{1}{d}tr(A)\mathbb{I}_d \quad (7.16)$$

$$A^+ = \frac{1}{2}(A + A^T) - D \quad (7.17)$$

$$A^- = \frac{1}{2}(A - A^T) \quad (7.18)$$

Le tenseur D décrit la composante irrotationnelle de dilatation isotrope du tenseur A , A^+ la composante de cisaillement et A^- la composante rotationnelle de vorticité.

Pour simplifier l'écriture des équations suivantes, nous définissons quelques notations. Nous utilisons \mathcal{C}_i pour indiquer la composante du champ \mathcal{C} , par exemple \mathcal{C}_1 représente la composante \mathcal{U} et \mathcal{C}_2 représente la composante \mathcal{V} . Nous introduisons le symbole de Kronecker $\delta_{i,j}$ défini par

$$\delta_{i,j} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \quad (7.19)$$

Nous utilisons aussi le symbole de Levi-Civita $\epsilon_{i,j}$, défini par

$$\epsilon_{i,j} = \det \begin{bmatrix} \delta_{i1} & \delta_{i2} \\ \delta_{j1} & \delta_{j2} \end{bmatrix} \quad (7.20)$$

Dans le cas d'un champ affine en dimension 2, l'équation (7.15) prend la forme

$$A = \frac{1}{2} \left(div(A)\mathcal{I}_d + def(A)S^T \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} S + rot(A) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right) \quad (7.21)$$

où $div(A)$, $def(A)$ et $rot(A)$ sont appelés respectivement divergence, déformation et rotationnel du tenseur A et définis respectivement par les équations (7.22), (7.23) et (7.24) et la matrice S est donnée par l'équation (7.25)

$$div(\mathcal{C}) = \sum_{i=1}^d \frac{\partial \mathcal{C}_i}{\partial \vec{x}_i} = div(A) = tr(A) \quad (7.22)$$

Si $d = 2$ alors $div(A) = a_{11} + a_{22}$.

$$def(\mathcal{C}) = def(A) = \sqrt{\sum_{i,j \in [0..d]^2} \left(\epsilon_{i,j} \frac{\partial \mathcal{C}_i}{\partial \vec{x}_i} \right)^2 + \sum_{i,j \in [0..d]^2} \left((1 - \delta_{i,j}) \frac{\partial \mathcal{C}_i}{\partial \vec{x}_j} \right)^2} \quad (7.23)$$

Si $d = 2$ alors $def(A) = \sqrt{(a_{11} - a_{22})^2 + (a_{12} + a_{21})^2}$.

$$rot(\mathcal{C}) = rot(A) = \sum_{i,j \in [0..d]^2} \epsilon_{i,j} \frac{\partial \mathcal{C}_j}{\partial \vec{x}_i} \quad (7.24)$$

Si $d = 2$ alors $rot(A) = a_{21} - a_{12}$.

$$S = \frac{1}{\sqrt{K}} \begin{bmatrix} a_{12} + a_{21} & def(A) - (a_{11} - a_{22}) \\ def(A) - (a_{11} - a_{22}) & -(a_{12} + a_{21}) \end{bmatrix} \quad (7.25)$$

avec K défini par

$$K = (a_{12} + a_{21})^2 + (def(A) - (a_{11} - a_{22}))^2 \quad (7.26)$$

De nombreux auteurs utilisent l'identité définie en équation (7.21) pour caractériser la structure des champs de vecteurs en dimension 2 [Shu 93], [C.-F. Shu 94], [Maurizot 97].

En dimension 2, la jordanisation de la matrice A et l'approche par décomposition tensorielle sont équivalentes. En effet, il existe des relations fondamentales entre invariants différentiels et algébriques exprimés par les équations 7.27 à (7.29)

$$tr(A) = div(A) \quad (7.27)$$

$$det(A) = \frac{1}{4} [div(A)^2 + rot(A)^2 - def(A)^2] \quad (7.28)$$

$$\Delta(A) = def(A)^2 - rot(A)^2 \quad (7.29)$$

À partir de ces relations, il est possible de reformuler la classification des portraits de phases affines en dimension 2 par rapport aux valeurs relatives des invariants différentiels plutôt qu'avec les signes des invariants algébriques. Ces différents résultats sont rapportés dans le tableau TAB 7.1.

7.2.3 Indice de Poincaré

Nous venons de voir dans les sections précédentes que la topologie d'un champ de vecteurs \mathcal{C} peut être complètement déterminée à partir de ses singularités. Un champ affine n'admet au maximum qu'un seul point singulier qui peut être calculé analytiquement. Dans la pratique, pour un champ quelconque, une telle détermination analytique est impossible. Il est donc nécessaire de disposer d'un critère permettant de mesurer la régularité ou une singularité locale d'un champ. Lorsque le champ est défini sur une variété continue, ce critère est trivial et $\mathcal{C}(\vec{x}) = 0$. Dans le cas où le support du champ est discret, donc observable en un nombre fini de points, cette définition n'est pas opérationnelle. Nous devons alors utiliser une caractérisation plus faible pour quantifier la présence d'une singularité au voisinage du point de mesure. Il est alors possible de s'appuyer sur la notion d'indice du champ.

Définition 10 Soit \mathcal{C} un champ de vecteurs sur une surface orientée $M \subset \mathbb{R}^3$. Étant donné $\vec{x} \in M$, soit $\vec{u} = (u_1, u_2) \rightarrow \vec{x}(\vec{u})$ une paramétrisation de M sur un voisinage ouvert $U \subseteq \mathbb{R}^2$ de \vec{x} . Considérons une courbe de Jordan α tracée sur M , régulière par morceaux, orientée positivement et munie d'une paramétrisation ξ à support compact W . On suppose que $\alpha(W) \subseteq \vec{x}(U)$ et que $M \cap \alpha(W)$ contient au plus un point singulier de \mathcal{C} . En désignant par $\mathcal{C}|_\alpha$ la restriction de \mathcal{C} à $\alpha(W)$, soit $\phi(\xi)$ une détermination différentiable de l'angle entre $\vec{x}_{u_1}(\vec{u})$ et $\mathcal{C}|_\alpha(\xi)$. Le réel noté $\mathcal{I}(\mathcal{C}, \vec{x})$ est défini par

$$\mathcal{I}(\mathcal{C}, \vec{x}) = \frac{1}{2\pi} \oint_\alpha d\phi \quad (7.30)$$

est indépendant de la courbe α et des paramétrisations ξ et \vec{u} . Il est appelé indice du champ \mathcal{C} au point \vec{x} .

Cette définition se généralise en dimension arbitraire [Denis-Brossard 00]. Dans le cas d'un champ de dimension 2, l'angle ϕ peut être déterminé par la fonction arc tangente. Dans ce cas, l'équation (7.30) s'exprime sous la forme

$$\mathcal{I}(\mathcal{C}, \vec{x}) = \frac{1}{2\pi} \oint_\alpha d(\arctan \frac{\mathcal{V}}{\mathcal{U}}) = \frac{1}{2\pi} \oint_\alpha \frac{\mathcal{U}(\vec{x})dx_1 - \mathcal{V}(\vec{x})dx_2}{\mathcal{U}(\vec{x})^2 + \mathcal{V}(\vec{x})^2} \quad (7.31)$$

Ici \mathcal{U} et \mathcal{V} sont la première et la deuxième composantes du champ de vecteurs. Le théorème de GAUSS-BONNET local [Do Carmo 76] implique que l'indice est un invariant topologique local d'un champ de vecteurs. Il constitue alors un descripteur local plus faible que les portraits de phase affines. Soit un champ \mathcal{C} sur un ouvert $U \subset \mathbb{R}^d$ et $J(\mathcal{C}, \vec{x})$ son jacobien en un point $\vec{x} \in U$, on peut démontrer que :

1. si \vec{x} est un point régulier de V alors $\mathcal{I}(\mathcal{C}, \vec{x}) = 0$
2. si \vec{x} est un point singulier non dégénéré de \mathcal{C} , alors

$$\mathcal{I}(\mathcal{C}, \vec{x}) = \begin{cases} 1 & \text{si } J(\mathcal{C}, \vec{x}) > 0, \\ -1 & \text{si } J(\mathcal{C}, \vec{x}) < 0 \end{cases}$$

Cela peut aussi être étendu aux points singuliers dégénérés [Spivak 79]. Dans le tableau TAB 7.1, nous présentons la valeur de l'indice pour les différents portraits de phase d'un champ de vecteurs en dimension 2. Il est important de noter que l'indice permet de séparer les points de selles de tous les autres types de singularités.

L'indice est un descripteur purement local. Le théorème suivant, qui est dérivé du théorème de GAUSS-BONNET global [Do Carmo 76], établit que l'intégrale de l'indice d'un champ de vecteurs sur une variété compacte est fonction de la seule topologie globale de la variété, indépendamment de celle du champ considéré :

Théorème 3 Soit \mathcal{C} un champ de vecteurs sur une variété compacte M , n'admettant que des points singuliers $(\vec{x}_{c_1}, \dots, \vec{x}_{c_n})$ non dégénérés. Alors

$$\sum_{i=1}^n \mathcal{I}(\mathcal{C}, \vec{x}_{c_i}) = \chi(M) \quad (7.32)$$

où $\chi(M)$ est la caractéristique de Euler-Poincaré de la variété M .

En conséquence, il n'existe pas de champ de vecteurs non singulier sur des variétés compactes de caractéristique non nulle. L'indice est communément appelé indice de Poincaré. Il a été utilisé pour caractériser la structure des champs de déplacements en dimension 2 dans [Ford 94], [Cohen 96] et [Nogawa 97].

Les principaux résultats de cette section concernant la définition mathématique d'une singularité sont résumés dans le tableau TAB 7.1. Le formalisme mathématique nous permet d'étudier les méthodes d'extraction des points singuliers d'un champ de déplacements en analysant ces derniers comme des champs de vecteurs au sens mathématique du terme.

7.3 État de l'art sur la détection de points singuliers

7.3.1 Méthodes fondées sur l'utilisation du portrait de phase

La méthode des portraits de phase pour la détection de structures singulières dans les champs de déplacements ou dans les champs de gradients a été introduite en 1992 par RAO et JAIN [Rao 92a]. La méthode que nous proposons se base sur ces travaux, c'est pourquoi nous la détaillons ci-après.

Leur idée est d'approximer localement un mouvement fluide par une équation différentielle linéaire 2-D.

$$\vec{x}'(t) = \frac{d\vec{x}}{dt} = L\vec{x} \quad (7.33)$$

En s'appuyant sur les propriétés mathématiques que nous avons détaillé dans la section précédente, et sur le théorème de GROBMAN-HARTMAN (cf. théorème 1 et 2) la matrice L est représentée localement par la matrice A d'un modèle affine. Ils proposent d'estimer ce modèle affine du mouvement par minimisation de la déviation angulaire par une méthode des moindres carrés non linéaires dans un schéma de LEVENBERG-MARQUARDT. À partir de ce modèle, si elle existe, ils déterminent la position d'une singularité. Pour classifier les singularités, ils utilisent la forme de Jordan de A tel que nous l'avons présenté précédemment. Ces valeurs propres permettent de déterminer le portrait de phase de l'équation différentielle linéaire comme indiqué sur la figure FIG 7.1.

À la suite de ces travaux, plusieurs auteurs ont proposé des méthodes utilisant l'analyse des portraits de phase. Dans [Shu 93], les auteurs ont proposé de changer l'étape d'estimation du mouvement affine. Ils utilisent alors un estimateur des moindres carrés pondérés et ajoutent une contrainte sur la norme des paramètres tel que :

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \text{ et } \sqrt{a_{11}^2 + a_{12}^2 + a_{21}^2 + a_{22}^2} = 1 \quad (7.34)$$

De plus, ils n'utilisent plus les valeurs propres de la matrice de Jordan du mouvement affine pour discriminer les différentes structures. Ils utilisent les propriétés du champ de déplacements telles que le rotationnel, la divergence et la déformation, qui, rappelons le, sont équivalentes dans le cas d'un champ de vecteurs en dimension 2. Leur méthode est testée sur des images de mouvements fluides et pour de la détection de défauts dans les semi-conducteurs. Ces travaux sont étendus dans [C.-F. Shu 94] de manière à rendre la détection plus robuste au bruit. Ils présentent plusieurs résultats sur des champs synthétiques bruités et sur des textures orientées réelles.

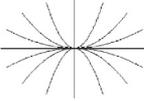
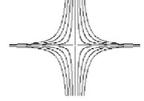
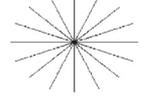
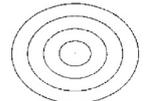
Portrait de phase (Indice)	Racines de \mathcal{P}_A	Forme de Jordan	Invariants algébriques	Invariants différentiels
noeud  $(\mathcal{I} = 1)$	réelles	$\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ $\lambda_1 \lambda_2 > 0$	$\Delta(A) > 0$ $\det(A) > 0$	$ def(A) > rot(A) $ $ div(A) > \sqrt{def(A)^2 - rot(A)^2}$
selle  $(\mathcal{I} = -1)$		$\lambda_2 > \lambda_1$	$\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ $\lambda_1 \lambda_2 < 0$	$\Delta(A) > 0$ $\det(A) < 0$
noeud étoilé  $(\mathcal{I} = 1)$	réelles	$\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{pmatrix}$	$\Delta(A) = 0$ $a_{12} - a_{21} = 0$	$ def(A) = rot(A) = 0$ $div(A) \neq 0$
noeud impropre  $(\mathcal{I} = 1)$		$\lambda_1 = \lambda_2$	$\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{pmatrix}$	$\Delta(A) = 0$ $a_{12} - a_{21} \neq 0$
centre  $(\mathcal{I} = 1)$	complexes $\lambda_{1,2} = \alpha \pm i\beta$	$\begin{pmatrix} 0 & -\beta \\ \beta & 0 \end{pmatrix}$	$\Delta(A) < 0$ $tr(A) = 0$	$ def(A) < rot(A) $ $div(A) = 0$
spirale  $(\mathcal{I} = 1)$		$\alpha = \frac{1}{2}tr(A)$ $\beta = \frac{1}{2}\sqrt{-\Delta(A)}$	$\begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}$	$\Delta(A) < 0$ $tr(A) \neq 0$

TABLEAU 7.1 – Classification de différents portraits de phase affine d'un champ de vecteurs $2D$

7.3.2 Estimation du champ de déplacements et détection de structures singulières simultanées

MAURIZOT et al. [Maurizot 95] fondent leurs travaux sur le principe que les modèles linéaires donnent une approximation de qualité du champ de déplacements au voisinage d'une singularité (cf. théorème de GROBMAN-HARTMAN). Ces approximations linéaires correspon-

dent localement à des solutions asymptotiquement exactes des équations de NAVIER-STOKES. Ils proposent alors une méthode permettant de réaliser conjointement la localisation de points singuliers, d'en délimiter le voisinage et d'estimer le modèle paramétrique de mouvement.

Au début, l'image est découpée en blocs carrés de dimension suffisamment grande pour avoir un support de calcul assez grand mais relativement petit pour que le modèle linéaire soit valable. Notons que l'algorithme fait évoluer de manière optimale la taille de la fenêtre. Par conséquent, le choix de la taille du support n'est pas critique. À l'étape d'initialisation, on choisit un bloc. Le modèle linéaire du mouvement est calculé à l'intérieur de ce bloc par une méthode d'estimation robuste multi-résolutions [Odobez 95]. Grâce à l'équation (7.2), on calcule la position \tilde{x} d'un point singulier \vec{x} . S'il n'existe pas, on recommence avec un autre bloc. Lorsqu'il existe, plusieurs cas peuvent se présenter :

- la position \tilde{x}_i se trouve à l'intérieur de la fenêtre d'estimation alors le prochain centre C_{i+1} sera à la position \tilde{x}_i ;
- la position \tilde{x}_i se trouve à l'extérieur de la fenêtre d'estimation, alors C_{i+1} correspondra à l'intersection du segment reliant le centre C_i à C_{i+1} avec le bord de la fenêtre d'estimation.

Il faut alors décider de la modification de la taille T_i de la fenêtre d'estimation. Ils proposent une méthode statistique qui s'appuie sur une fonction de risque prenant en compte le biais et la variance sur l'estimation de \tilde{x}_i .

En fonction de ce risque, pour un centre C_i donné, on calcule pour différentes tailles de fenêtres la position du point singulier. Si la position reste statistiquement proche alors l'augmentation de la taille de la fenêtre permet d'améliorer l'estimation du modèle, sinon cela signifie que la fenêtre est trop grande pour que le modèle linéaire soit valable.

La convergence globale est obtenue lorsque la fenêtre d'estimation n'évolue plus, ni en taille, ni en position. Ils proposent alors de classer les singularités suivant l'arbre des invariants algébriques de la matrice A , présenté sur la figure FIG 7.1. Cela permet d'accélérer la classification des structures. Leur méthode est testée sur des séquences d'images météorologiques synthétiques et réelles.

7.3.3 L'indice de Poincaré

L'indice de Poincaré peut être appliqué à des champs non-denses. Les méthodes utilisées dans [Ford 94] et [Cohen 96] sont basées sur le calcul de l'indice de Poincaré du champ de déplacements.

Cet indice est calculé le long d'une courbe de Jordan fermée α par l'intégrale suivante :

$$I(C, \vec{x}) = \frac{1}{2\pi} \int_{\alpha} \frac{\mathcal{U}(\vec{x})dx_1 - \mathcal{V}(\vec{x})dx_2}{\mathcal{U}(\vec{x})^2 + \mathcal{V}(\vec{x})^2} \quad (7.35)$$

Dans [Cohen 96], la méthode est appliquée à des champs de déplacements calculés au préalable par une méthode de flot optique. Ensuite, l'indice de Poincaré est calculé localement sur tout le champ. Lorsque le résultat de l'indice est $+1$, cela signifie qu'il y a soit un tourbillon, soit un puits, soit une source. Lorsque le résultat est -1 , il s'agit d'une selle.

Cette technique est utilisée dans [Memin 99] conjointement à l'estimation du mouvement. En effet, la détection de singularités fournissant des informations essentielles puisque dans son voisinage le mouvement peut être localement approximé par un modèle linéaire. Ces informations sont réinjectées dans le schéma d'estimation du mouvement.

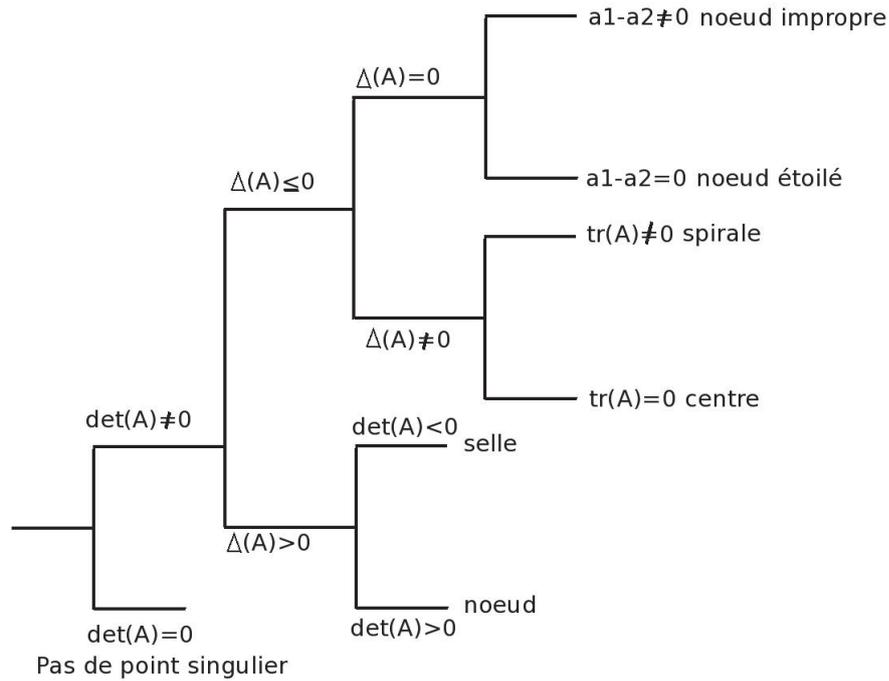


FIGURE 7.1 – Arbre binaire de représentation des invariants algébriques de la matrice A pour la classification des singularités proposé dans [Maurizot 95]

Cette méthode permet de détecter si une singularité existe à l'intérieur d'un chemin fermé, mais ne donne pas sa localisation précise. C'est par applications successives que la localisation est rendue possible. De plus, ce modèle ne permet pas de classifier les structures parmi les tourbillons ($I(\mathcal{C}, \vec{x}) = +1$) et les sources ou les puits ($I(\mathcal{C}, \vec{x}) = -1$).

7.3.4 Les fonctions de potentiels

Les fonctions de potentiels peuvent être utilisées pour détecter les singularités d'un champ de déplacements [Corpetti 03]. Ces fonctions de potentiels sont la fonction de potentiel de vitesse, que nous notons ϕ , et la fonction de courant, que nous notons ψ . Ces fonctions sont liées à la décomposition de Helmholtz-Hodge qui permet de décomposer un champ \mathcal{C} en un champ irrotationnel (\mathcal{C}_{irr}), un champ solénoïdal (\mathcal{C}_{sol}) et un champ laminaire (\mathcal{C}_{lam}).

$$\mathcal{C} = \mathcal{C}_{irr} + \mathcal{C}_{sol} + \mathcal{C}_{lam} \quad (7.36)$$

On peut obtenir les champs irrotationnels et solénoïdaux à partir des équations (7.37) à (7.39), avec $\mathcal{C}_{irr} = (\mathcal{U}_{irr}, \mathcal{V}_{irr})$, $\mathcal{C}_{sol} = (\mathcal{U}_{sol}, \mathcal{V}_{sol})$, $\mathcal{C}_{lam} = (\mathcal{U}_{lam}, \mathcal{V}_{lam})$:

$$\mathcal{C}_{irr}(x_1, x_2) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \nabla h(x_1 - p, x_2 - q) \text{div}(\mathcal{C}(p, q)) dpdq \quad (7.37)$$

$$\mathcal{C}_{sol}(x_1, x_2) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \nabla^\perp h(x_1 - p, x_2 - q) \text{curl}(\mathcal{C}(p, q)) dpdq \quad (7.38)$$

où $\nabla = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2})^T$, $\nabla^\perp = (-\frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_1})^T$ et h est le noyau de Green défini par :

$$h(x_1, x_2) = \frac{1}{2} \ln(x_1^2 + x_2^2) \quad (7.39)$$

Les équations (7.40) et (7.41) permettent de lier les champs irrotationnels et solénoïdaux aux fonctions de potentiels :

$$\mathcal{C}_{irr} = \nabla \phi \quad (7.40)$$

et

$$\mathcal{C}_{sol} = \nabla^\perp \psi \quad (7.41)$$

Ces fonctions de potentiels sont obtenues par l'équation (7.42) et (7.43) :

$$\phi(\vec{x}) = \frac{1}{2} \left(\int_0^{x_1} \mathcal{U}_{irr}(t, x_2) dt + \int_0^{x_2} \mathcal{V}_{irr}(x_1, t) dt + \int_0^{x_1} \mathcal{U}_{irr}(t, 0) dt + \int_0^{x_2} \mathcal{V}_{irr}(0, t) dt \right) \quad (7.42)$$

et

$$\psi(\vec{x}) = \frac{1}{2} \left(\int_0^{x_1} \mathcal{V}_{sol}(t, x_2) dt - \int_0^{x_2} \mathcal{U}_{sol}(x_1, t) dt + \int_0^{x_1} \mathcal{V}_{sol}(t, 0) dt - \int_0^{x_2} \mathcal{U}_{sol}(0, t) dt \right) \quad (7.43)$$

Les extrema locaux de ces fonctions de potentiels indiquent des points où les vitesses \mathcal{C}_{irr} ou \mathcal{C}_{sol} sont nulles. Ces points peuvent être des points singuliers. Le champ peut être représenté par un modèle du type $v = Ax$. Dans le cas irrotationnel, la matrice A_{irr} est donnée par :

$$A_{irr} = \begin{bmatrix} \frac{\partial^2 \phi}{\partial x_1^2} & \frac{\partial^2 \phi}{\partial x_1 x_2} \\ \frac{\partial^2 \phi}{\partial x_1 x_2} & \frac{\partial^2 \phi}{\partial x_2^2} \end{bmatrix} \quad (7.44)$$

Dans le cas solénoïdal, la matrice A_{sol} est donnée par :

$$A_{sol} = \begin{bmatrix} -\frac{\partial^2 \psi}{\partial x_1 x_2} & -\frac{\partial^2 \psi}{\partial x_2^2} \\ \frac{\partial^2 \psi}{\partial x_1^2} & \frac{\partial^2 \psi}{\partial x_1 x_2} \end{bmatrix} \quad (7.45)$$

La matrice A_{irr} est symétrique. Étant associée à un extremum, elle est définie positive (ou négative), ses valeurs propres sont donc non nulles et de même signe. On en déduit à l'aide du tableau TAB 7.1 que les extrema locaux de la fonction ϕ correspondent aux centres de noeuds ou de noeuds étoilés (structures de puits ou de sources en fonction du signe). La trace de la matrice A_{sol} est nulle par construction. On en déduit alors (cf. TAB 7.1) que les extrema locaux de la fonction ψ correspondent aux centres des tourbillons.

Cette méthode permet d'identifier séparément les structures du champ irrotationnel et celles du champ solénoïdal. Ces structures pouvant se superposer, c'est un avantage important de pouvoir traiter les deux composantes séparément. On peut cependant remarquer que cette méthode ne permet pas de détecter les points de selle et les noeuds impropres. Dans [Corpetti 03], cette technique est utilisée pour localiser la position des singularités dans les champs de déplacements. Pour chaque singularité détectée, la structure qui entoure cette position est modélisée par un modèle de RANKINE. Nous décrivons ce modèle dans la section suivante.

7.3.5 Méthodes de détection des tourbillons

7.3.5.1 Le modèle de RANKINE

Le modèle de RANKINE a été défini dans [Nogawa 97] pour modéliser les écoulements incompressibles. Ces champs ne sont pas composés de structures divergentes telles que des sources et des puits. Ce modèle ne peut donc détecter que des tourbillons. Cette détection est fondée sur une modélisation analytique du champ de déplacements. Les auteurs utilisent une fonction complexe pour décrire le mouvement autour du centre du tourbillon dans un plan complexe.

Le tourbillon est décrit par un mouvement linéaire à vorticité constante à l'intérieur d'une zone circulaire centrée sur le point singulier. À l'extérieur de cette zone, le mouvement est décrit par une fonction décroissante inversement au carré de la distance au point singulier. Notons $z = x + iy$ un point du domaine dans le plan complexe \mathbb{C} et $z_k = x_k + iy_k$ le centre d'un tourbillon. Alors le modèle de RANKINE d'un tourbillon peut être exprimé par l'équation (7.46)

$$f_v(z) \triangleq \begin{cases} g_v(z) = \frac{i\beta\|z\|}{z^2} & \text{si } \|z\| \geq r \\ h_v(z) = \frac{i\beta\|z\|}{r^2} & \text{si } \|z\| < r \end{cases} \quad (7.46)$$

où β exprime la force du vortex, r le rayon d'un disque centré en z_k à l'intérieur duquel ce modèle est applicable et $\|z\|$ la distance entre z et z_k .

La détection de tourbillon par le modèle de RANKINE s'appuie sur le théorème des résidus de Cauchy.

Théorème 4 (Théorème des résidus de Cauchy) Soit C un contour fermé sur lequel la fonction $f(z)$ est holomorphe, sauf en un nombre fini de points singuliers au sein de C pour lesquels les résidus sont r_1, r_2, \dots, r_n , on peut alors écrire

$$\oint_{(C)} f(z) dz = 2\pi i (r_1 + r_2 + \dots + r_n). \quad (7.47)$$

La détection de tourbillons est effectuée par le calcul de l'intégrale sur tout le champ. Cela permet de déterminer s'il y a une singularité sur le domaine complet. Ensuite, en reproduisant cela sur des sous-domaines du champ de déplacements par un processus récursif, il est possible de localiser une singularité.

Ce modèle est étendu dans [Corpetti 03] pour caractériser les sources et les puits détectés par les maxima des fonctions de potentiel, que nous avons introduit en section 7.3.4. Dans ce cas, le modèle est défini par l'équation (7.48)

$$f_p(z) \triangleq \begin{cases} g_p(z) = \frac{\alpha\|z\|}{z^2} & \text{si } \|z\| \geq r \\ h_p(z) = \frac{\alpha\|z\|}{r^2} & \text{si } \|z\| < r \end{cases} \quad (7.48)$$

où α est la force de la source.

Le modèle de RANKINE peut donc être utilisé pour localiser les tourbillons [Nogawa 97] ou pour paramétrer les structures qui y sont attachées [Corpetti 03].

7.3.5.2 Le critère λ_2

Le critère λ_2 ne permet de détecter que les structures tourbillonnaires. Ce modèle est dédié à l'analyse des écoulements fluides incompressibles. Plusieurs critères sont couramment utilisés pour caractériser la vorticit . Parmi ces indicateurs, trois quantit s topologiques sont utilis es : le facteur \mathcal{Q} [Chong 90], le crit re λ_2 [Jeong 95] et [Jardin 08], et le crit re d_2 [Vollmers 01]. L'utilisation de ces trois indicateurs est  quivalente dans le cas d'un  coulement incompressible en deux dimensions. Nous pr sentons, ici, le crit re λ_2 tel qu'il est utilis  dans [Jardin 08]. Dans le cas d'un champ de d placements bidimensionnel, la distribution λ_2 s' crit :

$$\lambda_2 = \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} - \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} \quad (7.49)$$

Une valeur de λ_2 n gative est significative d'un minimum de pression qui indique une forte probabilit  de la pr sence d'un tourbillon. De plus le crit re λ_2 permet de s parer le cisaillement de la vorticit .

Dans [Jardin 08], le crit re λ_2 est utilis  pour param triser un mod le de tourbillon de Lamb-Oseen. Ce mod le de tourbillon s'exprime en coordonn e polaire par

$$\begin{aligned} \mathcal{C}_r(r) &= 0 \\ \mathcal{C}_\theta(r) &= \frac{\Gamma}{2\pi r} \left(1 - e^{-r^2/r_0^2}\right). \end{aligned}$$

o  r_0 est le rayon du noyau du tourbillon, Γ est la circulation. La distribution λ_2 associ e   un champ de cette nature est de la forme

$$\lambda_2^{LO}(r) = \frac{\Gamma^2}{4\pi^2 r^2} \left[\frac{1}{r^2} - \left(\frac{1}{r^2} + \frac{2}{r_0^2} \right) e^{-r^2/r_0^2} \right] \left(1 - e^{-r^2/r_0^2}\right) \quad (7.50)$$

La distribution λ_2 d'un tourbillon de Lamb-Oseen est   moyenne nulle et est de carr  int grable. Cette distribution satisfait donc les premi res d finitions d'une fonction d'ondelettes. Dans la pratique, cette distribution est donc directement utilis e pour r aliser un filtrage par ondelettes avec une famille de filtres Ψ_σ caract ris es par un param tre d' chelle σ de la forme :

$$\Psi_\sigma(r) = A \frac{\sigma}{r^2} \left[\frac{\sigma^2}{r^2} - \left(\frac{\sigma^2}{r^2} + 2 \right) e^{-r^2/\sigma^2} \right] \left(1 - e^{-r^2/\sigma^2}\right), \quad (7.51)$$

Cela permet une param trisation directe de la circulation du tourbillon qui est proportionnelle   la racine carr e des coefficients d'ondelettes quand un maximum local de la fonction d'ondelettes est d tect . Les rayons des tourbillons sont alors connus puisqu'ils d pendent directement de l' chelle. Cette m thode est modulaire puisqu'elle peut aussi permettre de prendre en compte plus de param tres tels que l'ellipticit  ou l'orientation des structures observ es. Le crit re λ_2 est un invariant galil en, cela rend donc possible la d tection de structures se d pla ant avec le flux.

Notons que si l'on consid re localement un mod le affine du mouvement tel que

$$\mathcal{C}(\vec{x}) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (7.52)$$

alors on peut constater que

$$\lambda_2 = \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} - \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} = -\det(A) \quad (7.53)$$

Cela nous permet d'établir un lien entre les portraits de phase et l'indicateur λ_2 . Cet indicateur est comparé à notre méthode dans [Kihl 08b] ainsi que dans la section 7.5.

7.3.6 Synthèse

Nous présentons dans le tableau TAB 7.2, une synthèse des différentes approches présentées dans cet état de l'art. Nous précisons pour chacune d'elles, le type de structures qu'elle est susceptible de détecter et de classifier. Nous constatons rapidement que les méthodes par portraits de phase sont celles qui permettent de détecter et de classifier le plus grand nombre de singularités. Ces méthodes sont inhérentes à l'utilisation d'un modèle de mouvement affine, elles sont donc particulièrement bien adaptées à notre modèle polynomial. C'est pourquoi nous proposons une approche de cette famille.

Approche	détection	classification
Portrait de phase nos méthodes , [Rao 92a], [Shu 93] et [C.-F. Shu 94]	6 portraits de phase noeuds, selles, noeuds étoilés, noeuds impropres, centres et spirales	6 portraits de phase noeuds, selles, noeuds étoilés, noeuds impropres, centres et spirales
Indice de Poincaré [Ford 94], [Cohen 96] et [Memin 99]	tourbillons, sources (ou puits) et selles	+1 → tourbillons sources (ou puits) -1 → selles
Fonctions de potentiels [Corpetti 03]	tourbillons et sources	tourbillons et sources
Modèle de RANKINE [Nogawa 97]	tourbillons	tourbillons
Modèle de RANKINE [Corpetti 03]		tourbillons et sources (ou puits)
facteur Q [Chong 90]	tourbillons	tourbillons
critère λ_2 [Jeong 95] et [Jardin 08]	tourbillons	tourbillons
critère d_2 [Vollmers 01]	tourbillons	tourbillons

TABLEAU 7.2 – Tableau de synthèse des différentes approches de détection de singularités

7.4 Polynômes orthogonaux multi-variables pour la détection de points singuliers

Nous souhaitons proposer une méthode permettant de détecter et de classifier les points singuliers indépendamment de l'application. Il s'agit donc d'être capable de détecter tous les types de singularités qui peuvent apparaître dans un champ de déplacements 2D. Nous supposons ici que le champ de déplacements a été obtenu au préalable par une méthode quelconque d'estimation du mouvement. Notre méthode de détection et de classification peut s'adapter à tous types de champs qu'ils soient denses ou non, en utilisant soit des bases denses (cf. chapitre 3 à 4), soit des bases éparées (cf. chapitre 5) en fonction de la nature du champ. La méthode fondée sur l'utilisation du portrait de phase affine est adaptée à notre modèle polynomial.

7.4.1 Approximation du modèle affine

Nous présentons rapidement comment nous obtenons notre modèle de mouvement affine par projection sur une base de polynômes. La présentation des bases ayant fait l'objet de la partie II, nous ne rappelons ici que quelques points clef.

La première étape consiste à générer une base de polynômes denses ou non en fonction de la densité du champ à analyser. Ce choix n'influe pas sur la qualité de la détection, mais uniquement sur la complexité algorithmique de la méthode. En cas d'incertitude sur la nature du champ (dense ou éparé), il est possible d'utiliser les bases éparées. Le fait de générer une base dense par la méthode de génération des bases éparées n'aura pas d'effet sur la qualité de la détection.

La projection d'une composante du champ sur un polynôme $P_{i,j}$ de la base est effectuée par

$$\tilde{u}_{i,j} = \frac{\langle \mathcal{U} | P_{i,j} \rangle}{\langle P_{i,j} | P_{i,j} \rangle} \quad (7.54)$$

Le champ de déplacements est donc approximé par une base de degré D par la formule :

$$\tilde{\mathcal{U}} = \sum_{i=0}^D \sum_{j=0}^{D-i} \tilde{u}_{i,j} P_{i,j} \quad (7.55)$$

La projection de la composante verticale du champ de déplacements \mathcal{V} est effectuée de manière similaire. Ici D vaut toujours 1.

Afin de déterminer le modèle de mouvement affine tel qu'il est utilisé dans [Rao 92b], nous devons transposer nos coefficients dans la base canonique tel que cela est expliqué au **chapitre.5** en **section.5.3**. Nous devons donc transformer les coefficients $\tilde{u}_{i,j}$ et $\tilde{v}_{i,j}$ obtenus par projection en $\gamma_{i,j}$ et $\gamma'_{i,j}$. Cette transformation est calculée par

$$X_P \Gamma = \tilde{C}_P \quad (7.56)$$

où la matrice de passage X_P est, pour une base de degré 1,

$$X_P = \begin{bmatrix} \langle 1 | P_{0,0} \rangle & \langle x_1 | P_{0,0} \rangle & \langle x_2 | P_{0,0} \rangle \\ \langle 1 | P_{1,0} \rangle & \langle x_1 | P_{1,0} \rangle & \langle x_2 | P_{1,0} \rangle \\ \langle 1 | P_{0,1} \rangle & \langle x_1 | P_{0,1} \rangle & \langle x_2 | P_{0,1} \rangle \end{bmatrix} \quad (7.57)$$

La matrice des coefficients dans la base canonique est

$$\Gamma = \begin{bmatrix} \gamma_{0,0} & \gamma'_{0,0} \\ \gamma_{1,0} & \gamma'_{1,0} \\ \gamma_{0,1} & \gamma'_{0,1} \end{bmatrix} \quad (7.58)$$

À partir des coefficients de projection, on peut obtenir \tilde{C}_P tel que

$$\tilde{C}_P = \begin{bmatrix} \tilde{u}_{0,0} \langle P_{0,0} | P_{0,0} \rangle & \tilde{v}_{0,0} \langle P_{0,0} | P_{0,0} \rangle \\ \tilde{u}_{1,0} \langle P_{1,0} | P_{1,0} \rangle & \tilde{v}_{1,0} \langle P_{1,0} | P_{1,0} \rangle \\ \tilde{u}_{0,1} \langle P_{0,1} | P_{0,1} \rangle & \tilde{v}_{0,1} \langle P_{0,1} | P_{0,1} \rangle \end{bmatrix} \quad (7.59)$$

Finalement, les coefficients dans la base canonique sont obtenus par

$$\Gamma = X_P^{-1} \tilde{C}_P \quad (7.60)$$

D'où A et b dans l'expression (3.25), deviennent :

$$A = \begin{bmatrix} \gamma_{1,0} & \gamma_{0,1} \\ \gamma'_{1,0} & \gamma'_{0,1} \end{bmatrix} \text{ et } b = \begin{bmatrix} \gamma_{0,0} \\ \gamma'_{0,0} \end{bmatrix}$$

Notons, qu'à partir de ce modèle, il n'est possible d'extraire qu'une singularité sur le domaine spatial. Afin d'extraire plusieurs singularités dans un champ de déplacements, nous utilisons l'un des trois schémas multi-résolutions présentés dans la section suivante.

7.4.2 Schémas multi-résolutions

L'estimateur présenté ci-dessus est très efficace sur un support spatial donné, tant au niveau de la précision de détection d'une singularité qu'en terme de temps de calcul. Nous pouvons donc l'inclure dans un schéma multi-résolutions afin de détecter plusieurs singularités dans un champ de déplacements complet. Nous souhaitons détecter les singularités à différentes échelles. Nous proposons ici trois algorithmes.

7.4.2.1 Algorithme à fenêtre adaptative

Les singularités que l'on peut détecter dépendent à la fois de la taille de la structure qui leur est attachée et de la taille de la fenêtre dans laquelle le modèle affine du mouvement est calculé. Afin de pouvoir prendre en compte ces deux paramètres, nous proposons de réaliser la détection sur l'ensemble du champ avec une fenêtre dont la dimension varie entre $N \times N$ et $n \times n$ vecteurs.

L'algorithme est initialisé pour une fenêtre de $N \times N$ vecteurs. Cette fenêtre parcourt le champ entier en réalisant une détection pour chaque vecteur du champ. Lorsque tout le champ a été scruté, la taille de la fenêtre est décrétementée de deux vecteurs, puis le processus recommence jusqu'à ce que la fenêtre mobile soit de dimension $n \times n$ vecteurs. Le choix des paramètres N ($N > n$) et n dépend de l'application.

La figure FIG 7.2 illustre cette méthode. Parfois une singularité peut être détectée plusieurs fois à quelques pixels de différence. Nous proposons une stratégie pour trouver la meilleure position. Quelque soit la taille de la fenêtre, lorsqu'une singularité est détectée, la partie du champ

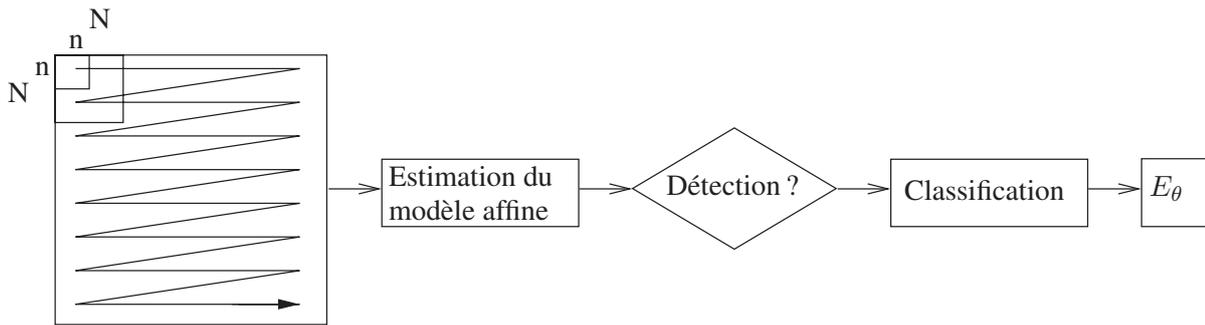


FIGURE 7.2 – Synthèse de l'algorithme à fenêtre adaptative présenté dans [Kihl 08a].

couverte par la fenêtre est reconstruite à partir du modèle affine. Ensuite, l'erreur angulaire est calculée entre cette partie du champ original et sa version reconstruite à partir du modèle affine par la formule :

$$E_{\theta} = \sum_w \frac{1}{2} | \sin(\Theta(\vec{x}) - \tilde{\Theta}(\vec{x})) | \quad (7.61)$$

où $\Theta(x_1, x_2)$ représente l'orientation entre $[-\frac{\pi}{2}; \frac{\pi}{2}]$ d'un vecteur $\mathcal{C}(\vec{x})$ du champ. Finalement, pour déterminer la meilleure position de la singularité, il suffit de chercher dans un voisinage donné de la singularité la position à laquelle est associée l'erreur angulaire minimale.

Bien que le choix des dimensions de la fenêtre d'approximation ainsi que la nécessité de choisir dans un voisinage la meilleure position des singularités soit contraignant, ce schéma peut être utilisé pour détecter les structures se déplaçant avec le flux. En effet, certaines méthodes spécialement dédiées à l'étude des structures tourbillonnaires (cf. section 7.3.5.2), permettent de détecter ce type de structures. Pour cela, ces méthodes utilisent un critère qui est un invariant galiléen. Notre modèle n'en est pas un, mais il est tout de même possible d'obtenir une détection de ces structures. Pour cela, il est alors nécessaire d'ajouter une étape avant l'estimation du modèle affine. Pour chaque position de la fenêtre adaptative, nous retirons à l'ensemble des vecteurs de la fenêtre, la valeur du vecteur central. Ensuite, le schéma multi-résolutions reste le même.

7.4.2.2 Algorithme multi-résolutions à fenêtre mobile

Cet algorithme est très proche du premier. La différence se situe au niveau de la fenêtre d'analyse. Celle-ci est de dimension fixe. La détection des structures aux différentes échelles est effectuée en appliquant la même fenêtre d'analyse sur le champ à différentes résolutions. Ces différentes résolutions sont obtenues par la méthode de décomposition multi-résolutions non-dyadique présentée au chapitre 6 à la section 6.3.3.4. La figure FIG 7.3 illustre cet algorithme. Il est aussi possible d'utiliser cet algorithme comme un invariant galiléen. Cet algorithme est plus coûteux en espace mémoire que le précédent puisqu'il faudra stocker le champ à toutes les résolutions.

7.4.2.3 Algorithme par quadtree

Ici nous utilisons un quadtree afin d'obtenir une partition par grille irrégulière de manière automatique. Cela permet de ne pas se poser la question de la taille de la fenêtre comme cela est le cas avec l'algorithme à fenêtre adaptative.

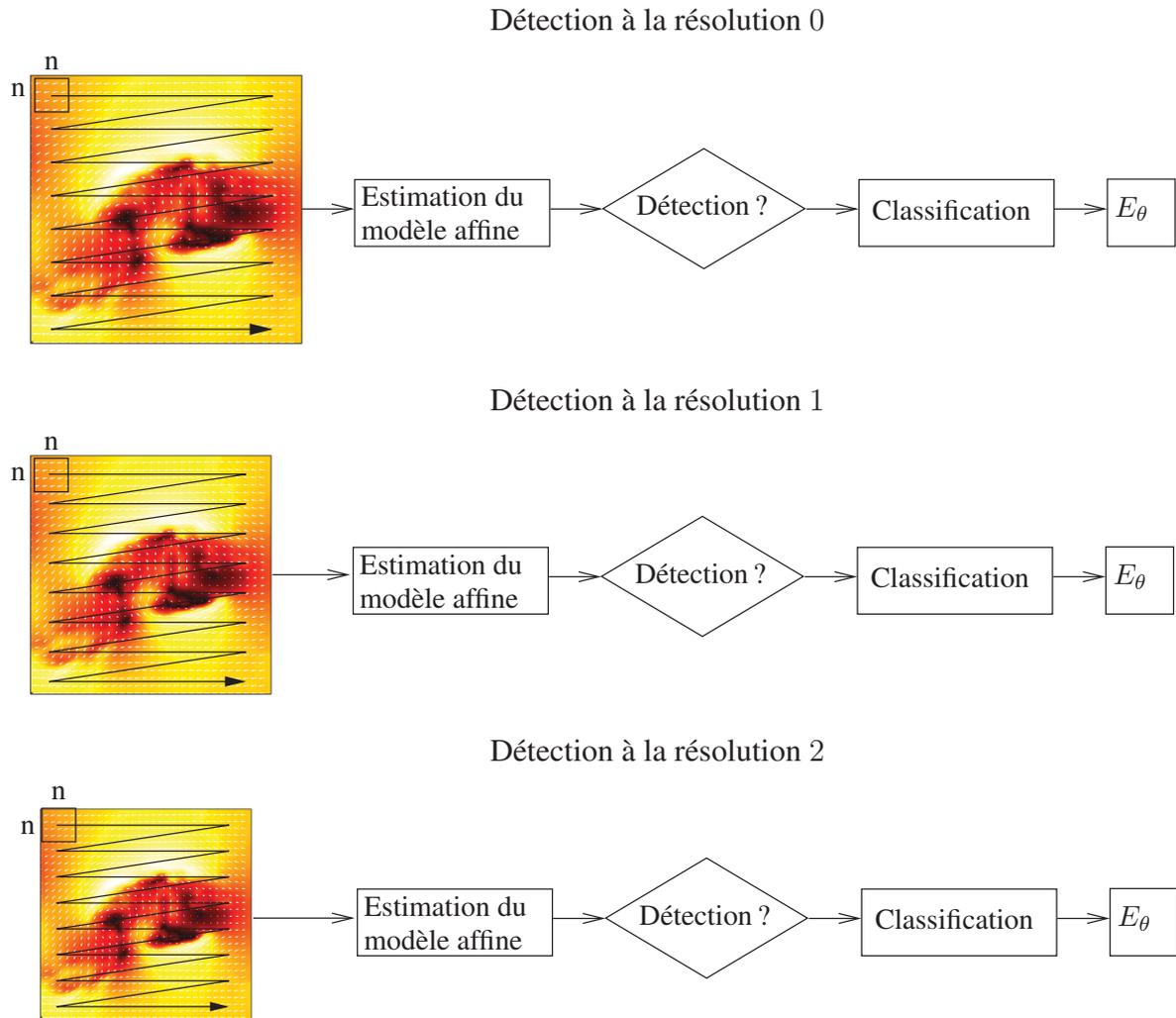


FIGURE 7.3 – Synthèse de l’algorithme multi-résolutions à fenêtre mobile.

L’algorithme est le suivant :

1. Calculer l’approximation du champ avec un quadtree et une base de degré 1 ;
2. Sur chaque feuille du quadtree, réaliser une détection de singularité avec l’équation (3.26) ;
 - si la singularité se trouve à l’intérieur du domaine spatial couvert par cette feuille, alors la singularité est conservée ;
 - sinon la singularité est rejetée ;
3. Pour chaque singularité conservée, on calcule son domaine de linéarité ;
4. Classifier la singularité sur son domaine de linéarité à partir de A .

Il est aussi possible d’effectuer une détection à chaque niveau du quadtree. C’est-à-dire sur chaque noeud du quadtree et non seulement sur les feuilles. Cela permet de détecter plus de structures. Certaines sont des faux positifs, mais la méthode d’estimation de la zone de linéarité que nous proposons ci-dessous permet de supprimer les structures de trop petite taille.

7.4.3 Détermination de la zone de linéarité autour de la singularité

Après avoir détecté une singularité, il est souvent très intéressant de connaître la zone de validité du modèle linéaire qui entoure la singularité.

Chaque singularité est traitée séparément. La méthode se décompose en deux étapes. La première étape consiste à estimer la plus petite fenêtre pour laquelle la singularité est détectée lorsque l'on est centré dessus. Pour cela, nous estimons le modèle affine dans une fenêtre de petite dimension (dans notre cas 3×3 vecteurs) centrée sur la singularité. Si la singularité est détectée dans cette fenêtre, alors la dimension de détection minimale est trouvée. Sinon, la fenêtre est agrandie d'un vecteur de part et d'autre de la singularité et la détection recommence jusqu'à ce que la dimension de détection minimale soit trouvée. Dans notre cas, au delà de 49×49 vecteurs, nous rejetons le point singulier si la dimension de détection minimale n'a pas été trouvée.

En effet, il est important de noter qu'il n'est pas possible d'identifier la zone de linéarité pour tous les points singuliers détectés. Dans certains cas, les singularités détectées par l'un de nos algorithmes, ne le sont plus lorsque l'on se centre la fenêtre d'estimation du modèle sur ces dernières.

La deuxième étape de la méthode d'estimation de la zone de linéarité est la suivante. On initialise une fenêtre aux dimensions de la dimension de détection minimale puis :

1. On estime le modèle affine $Ax + b$;
2. On détecte la position d'une singularité par $\tilde{x} = -A^{-1} b$;
 - si la nouvelle position détectée est à une distance inférieure à K pixels (par exemple $K = 2$) en abscisse et en ordonnée **alors** la fenêtre est agrandie ($n = n + 1$). On retourne à l'étape 1 ;
 - **sinon** la zone de linéarité à un rayon de n pixels autour de la singularité.

Lorsque le domaine de linéarité d'une singularité est estimé, nous effectuons la classification de la structure à partir de la matrice A . Nous utilisons la méthode proposée dans [Maurizot 95] et présentée sur la figure FIG 7.1 pour effectuer la classification à partir des propriétés de A .

Cela permet d'obtenir une estimation de la zone de linéarité. Dans de nombreuses applications, les structures sont supposées circulaires. Avec notre méthode, il est possible de définir des bases polynomiales éparées définies sur un support circulaire. Nous proposons donc de réaliser les différentes estimations sur des bases de cette nature lorsque cela est nécessaire. Les résultats d'estimation de la zone de linéarité présentés dans ce chapitre sont obtenus avec des bases de cette nature.

7.5 Résultats de nos méthodes de détection et de classification de singularités

7.5.1 Comparaison entre la détection à partir de notre estimateur et à partir de l'estimateur de Levenberg-Marquardt

Dans cette section, nous comparons l'estimateur de LEVENBERG-MARQUARDT proposé par [Rao 92a] et notre estimateur reposant sur la projection du champ sur une base de polynômes

bivariable. Pour comparer ces deux estimateurs, nous avons programmé la méthode de RAO et JAIN [Rao 92a] en utilisant la fonction `leasqr` du logiciel octave [Eaton 02].

La méthode de RAO et JAIN n'utilise que l'orientation des vecteurs. Afin de nous placer dans les mêmes conditions, nous normalisons les champs de déplacements en divisant chaque composante d'un vecteur par la norme de ce dernier. Pour notre méthode, le champ normalisé est projeté sur la base de polynômes. Cependant, nous rappelons qu'habituellement, aucune normalisation n'est effectuée avant projection. Nous calculons ensuite l'estimation affine du modèle (A et b) par la méthode de LEVENBERG-MARQUARDT. RAO et JAIN proposent de déterminer la matrice A par minimisation de S , définie à partir des équations (7.62) et (7.63)

$$S = \sum_{x \in W} \mathcal{A}(x) \text{ avec} \quad (7.62)$$

$$\mathcal{A}(x) = \frac{1}{2} R(x) \tilde{R}(x) | \sin(\Theta(x) - \tilde{\Theta}(x)) | \quad (7.63)$$

où W est la fenêtre d'estimation du modèle, R et Θ sont l'amplitude et l'orientation des vecteurs du champ et \tilde{R} et $\tilde{\Theta}$ leur approximation. Afin de parvenir à l'estimation du modèle affine, ils proposent d'utiliser une méthode de minimisation au sens des moindres carrés non-linéaires avec le schéma de LEVENBERG-MARQUARDT. Ils réalisent deux minimisations correspondant à deux initialisations différentes des paramètres de la matrice A . La première minimisation est initialisée avec $A = (1, 1, 0, 0)$. Elle permet d'obtenir le champ de déplacements $W1$ après convergence. La deuxième initialisation est effectuée avec $A = (0, 1, 1, 0)$. Elle permet d'obtenir le second champ $W2$. Le champ de déplacements entre $W1$ et $W2$ qui minimise la déviation angulaire est retenu comme solution.

Sur la figure FIG 7.4, nous illustrons en FIG 7.4(b), FIG 7.4(c) et FIG 7.4(d), plusieurs estimations du mouvement affine présenté en FIG 7.4(a). Le modèle est d'abord estimé en FIG 7.4(b) par la méthode de projection sur une base polynomiale. Ensuite, sur les figures FIG 7.4(c) et FIG 7.4(d), nous présentons les résultats obtenus avec l'estimateur de LEVENBERG-MARQUARDT respectivement pour le champ $W1$ et $W2$. Dans ce cas, la première estimation est rejetée et la seconde est conservée, puisque l'erreur angulaire obtenue sur le champ $W2$ est inférieure à celle de $W1$. On peut noter que l'orientation des vecteurs présentés sur le champ de la figure FIG 7.4(d) est inversée par rapport au champ d'origine. En effet, nous rappelons qu'avec la fonctionnelle choisie par RAO et JAIN, seule l'orientation du champ intervient et non le sens.

Bien que cela soit rare, il peut arriver qu'aucune des deux approximations fournies par la méthode de LEVENBERG-MARQUARDT ne converge vers la bonne solution. Nous présentons sur la figure FIG 7.5 un exemple où ni $W1$ (cf. FIG 7.5(c)), ni $W2$ (cf. FIG 7.5(d)) ne correspondent au modèle du champ affine original (cf. FIG 7.5(a)). Nous présentons aussi en figure FIG 7.5(b), le modèle obtenu par projection sur la base de polynômes. On peut constater que ce dernier modèle est une bonne approximation du champ original. Il permet une bonne localisation de la singularité.

Sur la figure FIG 7.6, nous illustrons la résistance au bruit Gaussien de la méthode de projection. Nous présentons, dans un premier temps, le champ de déplacements affine original sur la figure FIG 7.6(a). Ensuite, ce champ est normalisé, nous le présentons sur la figure FIG 7.6(b). Nous ajoutons un bruit Gaussien centré sur 0 et d'écart type $\sigma = 1$ puis le champ est à nouveau normalisé. On obtient alors le champ présenté sur la figure FIG 7.6(c). Le résultat du modèle obtenu par projection est représenté sur la figure FIG 7.6(d). Les résultats d'approximation par l'estimateur de LEVENBERG-MARQUARDT sont présentés sur les figures FIG 7.6(e)

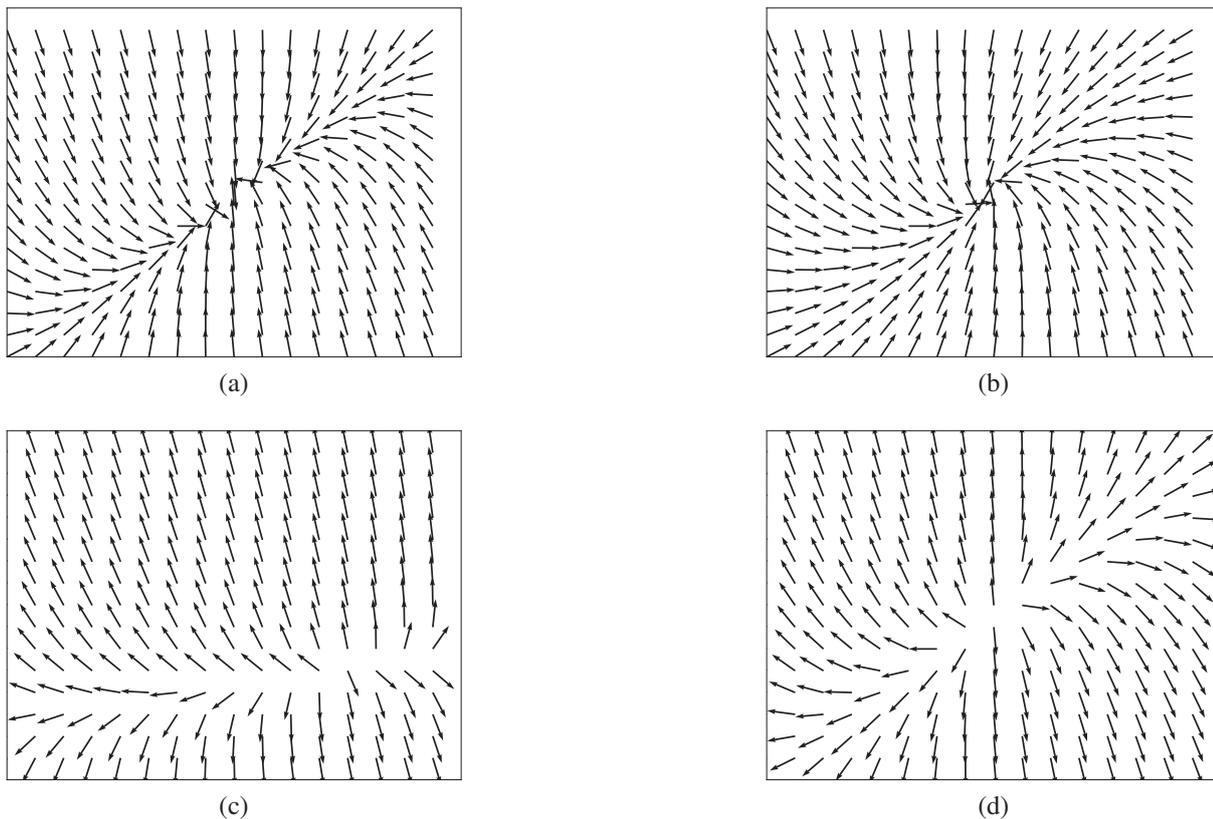


FIGURE 7.4 – Illustration des deux résultats $W1$ et $W2$ obtenus par l'estimateur de LEVENBERG-MARQUARDT ; (a) Champ de vecteur affine normalisé ; (b) Champ correspondant au modèle obtenu par projection du champ (a) sur une base de polynômes bivariés ; (c) Champ $W1$ correspondant à la première estimation du champ (a) par le modèle de LEVENBERG-MARQUARDT (rejeté) ; (d) Champ $W2$ correspondant à la seconde estimation du champ (a) par le modèle de LEVENBERG-MARQUARDT (conservé) ;

et FIG 7.6(f). On constate que le modèle obtenu par projection est bien plus proche du champ original normalisé de la figure FIG 7.6(b) que ceux obtenus avec l'estimateur de LEVENBERG-MARQUARDT. Du point de vue topologique, la meilleure solution proposée par la méthode de LEVENBERG-MARQUARDT est le champ FIG 7.6(f). En effet, celle-ci fait apparaître une singularité de même nature que dans le champ original. Cependant, l'algorithme de RAO et JAIN retient le champ FIG 7.6(e) comme solution puisque ce dernier minimise l'erreur angulaire. On peut donc supposer que notre méthode fournit une meilleure estimation du modèle affine que la méthode de RAO et JAIN.

Nous comparons maintenant statistiquement notre estimateur de mouvement affine à celui de RAO et JAIN [Rao 92a] dans un contexte de détection de singularité. De plus, nous comparons ces deux méthodes de détection à une autre fondée sur l'indice de Poincaré. Pour cela, nous avons calculé 10000 champs de déplacements affines synthétiques de résolution 16×16 vecteurs. Sur chaque champ, nous appliquons les deux estimateurs. À partir des deux modèles obtenus, nous effectuons une détection de singularité dans le champ. Afin d'avoir un plus grand nombre de références sur la détection, nous ajoutons à notre étude, une autre méthode de détection ne s'appuyant pas sur un modèle affine, l'indice de Poincaré [Ford 94]. Les champs étant par construction des champs affines de degré 1, il y a au maximum une seule singularité

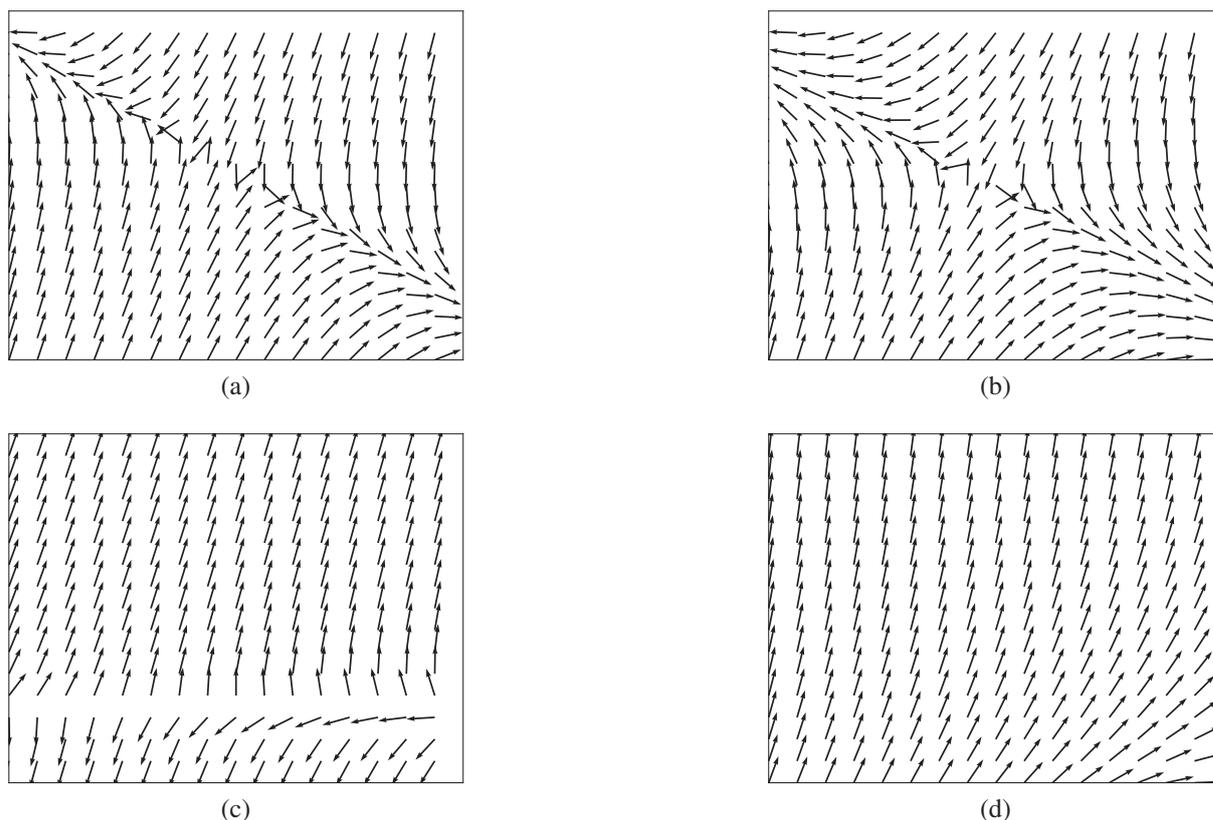


FIGURE 7.5 – Illustration d'un cas où la méthode de LEVENBERG-MARQUARDT ne fournit une bonne approximation ni pour $W1$, ni pour $W2$; (a) Champ de vecteur affine normalisé ; (b) Champ correspondant au modèle obtenu par projection du champ (a) sur une base de polynômes bivariables ; (c) Champ $W1$ correspondant à la première estimation du champ (a) par le modèle de LEVENBERG-MARQUARDT (rejeté) ; (d) Champ $W2$ correspondant à la seconde estimation du champ (a) par le modèle de LEVENBERG-MARQUARDT (conservé) ;

par champ. Chacune des méthodes de détection est alors appliquée de manière globale sur les champs. Les résultats sont présentés dans le tableau TAB 7.3. Pour ces tests, lorsqu'une singularité est détectée à l'intérieur du domaine du champ, nous considérons qu'elle est bien détectée quelque soit la position exacte.

On constate que les deux méthodes s'appuyant sur une estimation du mouvement affine fournissent une meilleure détection que l'indice de Poincaré sur les champs affines. Cependant, la méthode par projection semble fournir plus de faux positifs que la méthode proposée par RAO et JAIN. En revanche, la somme des faux positifs et des faux négatifs des deux méthodes rend la méthode par projection sur une base polynomiale meilleure. Les faux positifs obtenus par notre méthode sont dus à la normalisation du champ. En effet, nous avons normalisé le champ avant projection afin de se comparer avec la méthode de RAO et JAIN. Pour l'estimation par projection, il est préférable que le champ ne soit pas normalisé. Nous avons donc renouvelé l'expérience en fournissant à notre méthode un champ non normalisé. Les résultats sont présentés dans le tableau TAB 7.4. On constate alors que les résultats de la méthode de projection deviennent meilleurs. Il n'y a plus ni de faux positifs, ni de faux négatifs. En effet, nous avons vu au chapitre 3 qu'il est possible d'approximer sans erreur tout champ affine avec une base

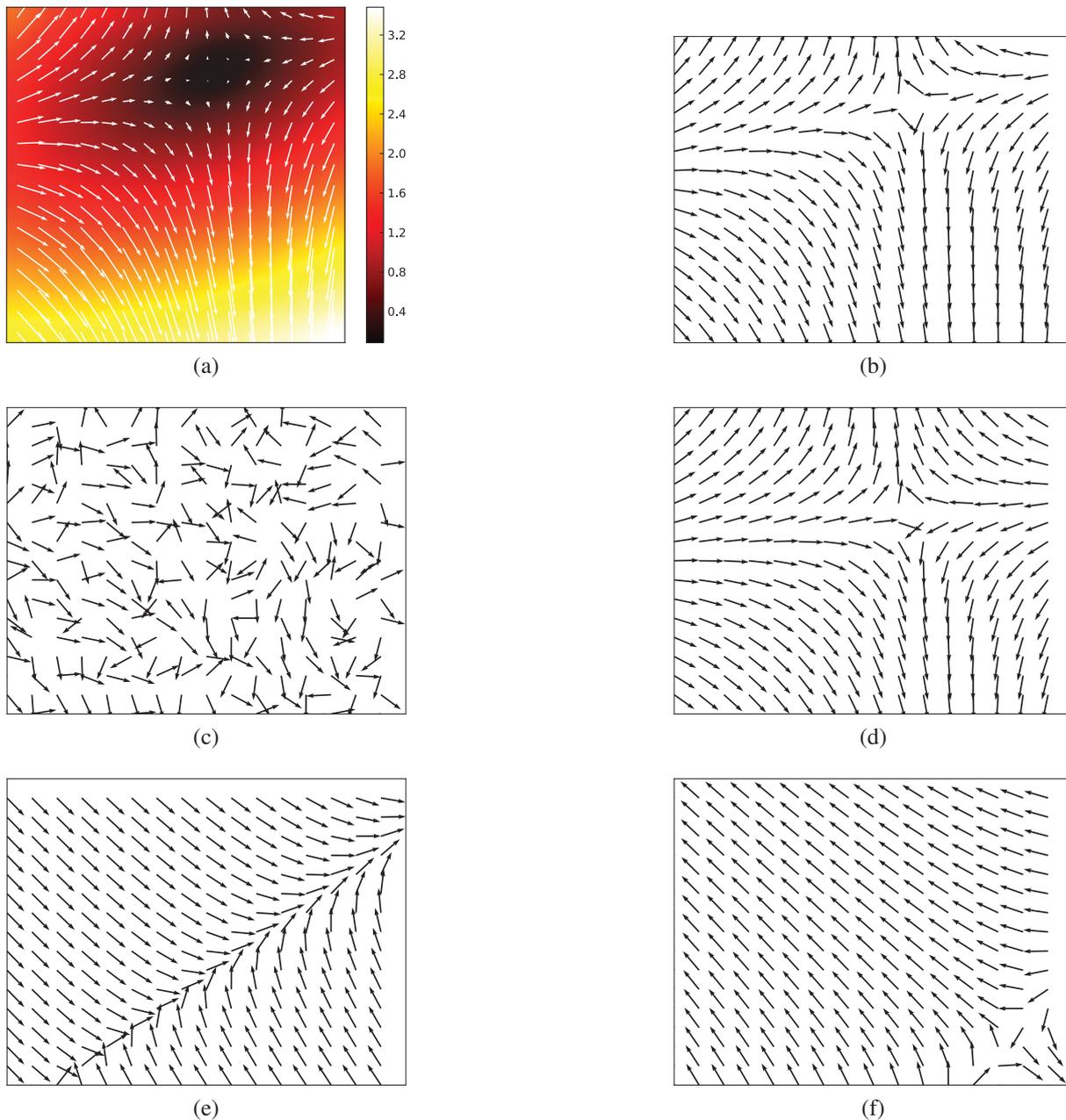


FIGURE 7.6 – Illustration de la résistance au bruit Gaussien de la méthode de projection ; (a) Champ de vecteur affine (non normalisé) ; (b) Champ de vecteur affine normalisé ; (c) Champ de vecteur affine bruité avec un bruit Gaussien centré sur 0 et d'écart type $\sigma = 1$ puis centré ; (d) Modèle obtenu par projection du champ (c) sur une base de polynômes bivariables ; (e) Première estimation du champ (c) par le modèle de LEVENBERG-MARQUARDT ($W1$) (solution retenue par RAO et JAIN) ; (f) Seconde estimation du champ (c) par le modèle de LEVENBERG-MARQUARDT ($W2$) ;

	Levenberg	Projection	Indice de Poincaré
Il y a <i>une singularité</i> dans le champ de déplacements et elle est <i>bien détectée</i> par la méthode	2931	2420	1917
Il y a <i>une singularité</i> dans le champ de déplacements et elle n'est <i>pas détectée</i> par la méthode	479	990	1493
Il n'y a <i>pas de singularité</i> dans le champ de déplacements mais la méthode en <i>détecte une</i>	634	0	229
Il n'y a <i>pas de singularité</i> dans le champ de déplacements et la méthode n'en <i>détecte pas</i>	5956	6590	6361

TABLEAU 7.3 – Résultats quantitatifs de détection de singularités dans des champs de vecteurs normalisés par trois méthodes. La première colonne donne les résultats de détection obtenus à partir de l'estimateur de LEVENBERG-MARQUARDT ; la deuxième colonne donne les résultats obtenus à partir de notre estimateur, par projection sur la base polynomiale (les champs sont normalisés) ; la troisième colonne donne les résultats obtenus par l'indice de Poincaré

de degré 1. Il est donc souhaitable lorsque cela est possible de ne pas réaliser de normalisation avant la projection sur la base polynomiale.

	Levenberg	Projection	Indice de Poincaré
Il y a <i>une singularité</i> dans le champ de déplacements et elle est <i>bien détectée</i> par la méthode	2824	3251	1749
Il y a <i>une singularité</i> dans le champ de déplacements et elle n'est <i>pas détectée</i> par la méthode	427	0	1502
Il n'y a <i>pas de singularité</i> dans le champ de déplacements mais la méthode en <i>détecte une</i>	609	0	193
Il n'y a <i>pas de singularité</i> dans le champ de déplacements et la méthode n'en <i>détecte pas</i>	6140	6749	6556

TABLEAU 7.4 – Résultats quantitatifs de détection de singularités dans un champ de vecteurs affine non normalisé par trois méthodes. La première colonne donne les résultats de détection obtenus à partir de l'estimateur de LEVENBERG-MARQUARDT ; la deuxième colonne donne les résultats obtenus à partir de notre estimateur, par projection sur la base polynomiale (les champs ne sont pas normalisés) ; la troisième colonne donne les résultats obtenus par l'indice de Poincaré

	Levenberg	Projection	Indice de Poincaré
Il y a <i>une singularité</i> dans le champ de déplacements et elle est <i>bien détectée</i> par la méthode	2124	2702	1845
Il y a <i>une singularité</i> dans le champ de déplacements et elle n'est <i>pas détectée</i> par la méthode	1238	660	1517
Il n'y a <i>pas de singularité</i> dans le champ de déplacements mais la méthode en <i>détecte une</i>	2480	204	3294
Il n'y a <i>pas de singularité</i> dans le champ de déplacements et la méthode n'en <i>détecte pas</i>	4158	6434	3344

TABLEAU 7.5 – Résultats statistiques de détection de singularités par trois méthodes sur des champs bruités. La première colonne donne les résultats de détection obtenus à partir de l'estimateur de LEVENBERG-MARQUARDT ; la deuxième colonne donne les résultats obtenus à partir de notre estimateur, par projection sur la base polynomiale (les champs ne sont pas normalisés) ; la troisième colonne donne les résultats obtenus par l'indice de Poincaré

Nous reproduisons cette expérience avec des champs bruités. Nous utilisons 10000 nouveaux champs affines aléatoires. Nous ajoutons à chaque champ un bruit Gaussien de moyenne nulle et d'écart type $\sigma = 0.1$. Chaque champ est ensuite normalisé. Bien que nous ayons vu que cela n'était pas favorable à la méthode de projection, nous continuons à le faire afin de se placer dans les mêmes conditions pour toutes les méthodes. On constate que notre estimateur permet d'obtenir les meilleurs résultats de détection. Notre estimateur a donc une meilleure résistance au bruit Gaussien que les deux autres.

Notre estimateur permet d'obtenir des résultats comparables dans le pire des cas et meilleurs en général. De plus, en présence de bruit Gaussien, notre estimateur est beaucoup plus robuste que les deux autres méthodes auxquelles nous l'avons comparé. L'ensemble de ces tests permet de valider le choix de notre estimateur.

7.5.2 Résultats des algorithmes multi-résolutions sur le champ 100 de la séquence Cylindre-Carré

Nous présentons ici quelques résultats obtenus avec chacun de nos trois algorithmes. Tous sont appliqués sur le champ 100 de la séquence cylindre-carré. Ce champ est présenté sur la figure FIG 7.7.

Nous présentons sur la figure FIG 7.8 le résultat de l'algorithme à fenêtre adaptative avec les paramètres suivants. La plus petite fenêtre est de 3×3 vecteurs et la plus grande de 11×11 vecteurs. On constate que les points singuliers identifiables visuellement semblent être identifiés par l'algorithme. Nous présentons une autre détection sur la figure FIG 7.9 avec la plus petite fenêtre de 7×7 vecteurs et la plus grande de 21×21 vecteurs. Afin de comparer ces résultats (carrés bleus) aux précédents (disques verts), nous les superposons sur la même figure. Plusieurs singularités sont détectées exactement aux mêmes positions que précédemment,

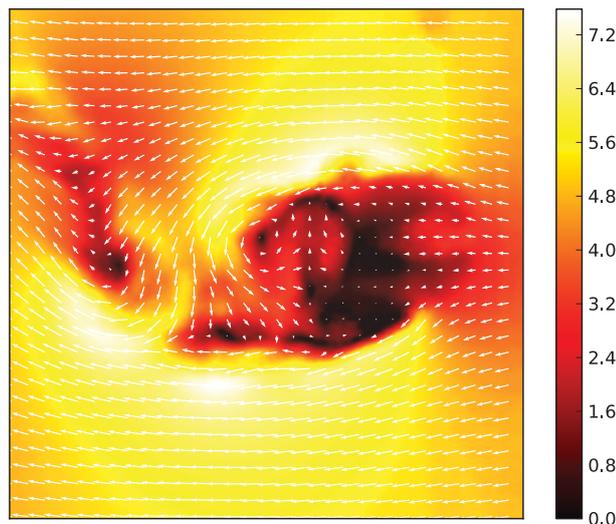


FIGURE 7.7 – Champ 100 de la séquence Cylindre-carré (cf. ANNEXE A) utilisé pour tester nos trois algorithmes de détection de singularités ; le champ est superposé à sa carte de norme.

cependant certaines singularités ne sont pas détectées, et de nouvelles le sont. Comme l'on pouvait s'y attendre, le choix des fenêtres a donc une influence directe et importante sur les résultats obtenus. Par conséquent, une attention toute particulière devra être portée quant à leur choix.

Nous présentons maintenant les résultats de l'algorithme multi-résolutions à fenêtre mobile. Ici, nous utilisons une fenêtre fixe de 5×5 vecteurs. Les singularités sont donc détectées à chaque résolution. La figure FIG 7.10 présente le résultat obtenu avec une décomposition sur 11 résolutions. La décomposition multi-résolutions est effectuée par la méthode multi-résolutions par base de polynômes A-complète non-dyadique. La base de polynômes de niveau 1 utilisée est de 16×16 polynômes et la base de niveau 2 de 15×15 polynômes. Les structures singulières identifiables visuellement semblent toutes être détectées.

Afin de suivre l'évolution de la détection en fonction de la résolution, nous présentons sur la figure FIG 7.11 le résultat de détection à la résolution 0 (disques verts), puis les nouvelles singularités détectées à la résolution 1 (carrés bleus) puis celles détectées à la résolution 2 (triangles rouges).

Nous présentons sur la figure FIG 7.12 le résultat de l'algorithme par quadtree sur 6 niveaux de décomposition. La détection est uniquement effectuée sur les feuilles du quadtree.

Sur la figure FIG 7.13, nous proposons une synthèse des résultats des trois algorithmes sur la même figure. Nous y présentons, avec des disques verts, les points détectés par l'algorithme par fenêtre adaptative, avec des carrés bleus, ceux obtenus avec l'algorithme multi-résolutions à fenêtre mobile, et, avec des triangles rouges, ceux obtenus avec l'algorithme par quadtree. On constate que les résultats des trois algorithmes sont sensiblement les mêmes. Les résultats de l'algorithme à fenêtre adaptative et l'algorithme à fenêtre mobile semblent néanmoins plus proches l'un de l'autre à l'exception de quelques points. L'algorithme par quadtree semble fournir un compromis entre les résultats des deux autres méthodes.

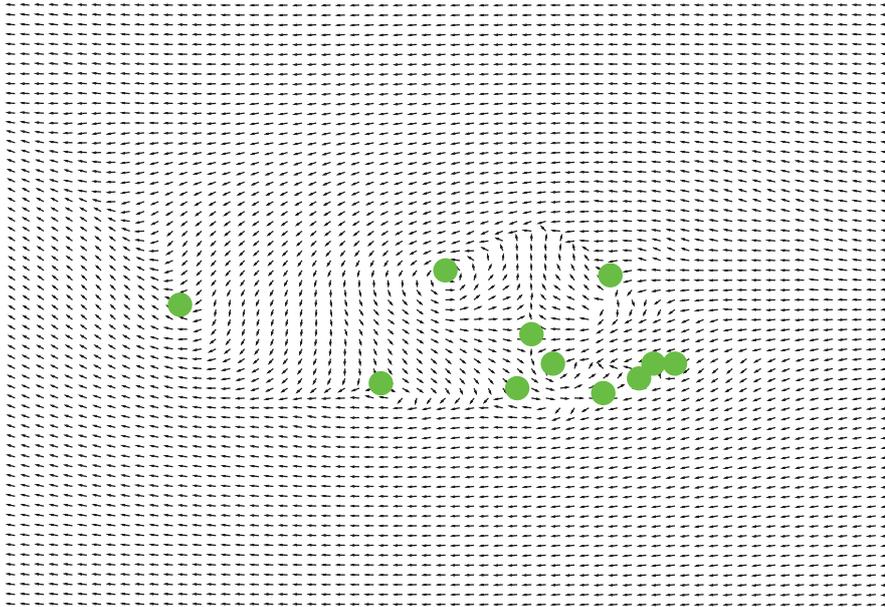


FIGURE 7.8 – Détection de singularités avec l’algorithme à fenêtre adaptative sur le champ 100 de la séquence Cylindre-carré ; pour l’affichage, le champ est normalisé ; fenêtre minimale = 3×3 vecteurs ; fenêtre maximale = 11×11 vecteurs ;

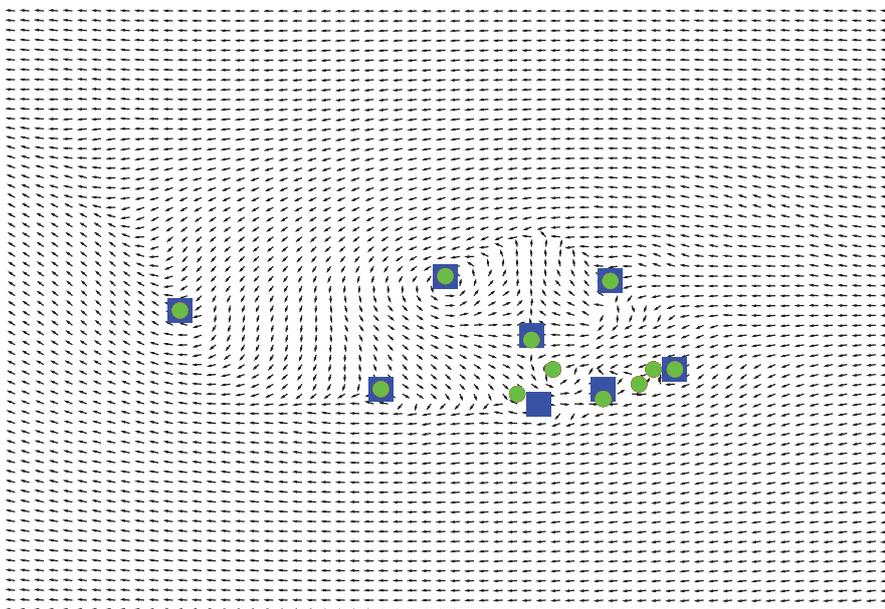


FIGURE 7.9 – Détection de singularités avec l’algorithme à fenêtre adaptative sur le champ 100 de la séquence Cylindre-carré ; pour l’affichage, le champ est normalisé ; les disques verts représentent les singularités détectées pour une fenêtre minimale de 3×3 vecteurs et une fenêtre maximale de 11×11 vecteurs ; les carrés bleus représentent les singularités détectées pour une fenêtre minimale de 7×7 vecteurs et une fenêtre maximale de 21×21 vecteurs ;

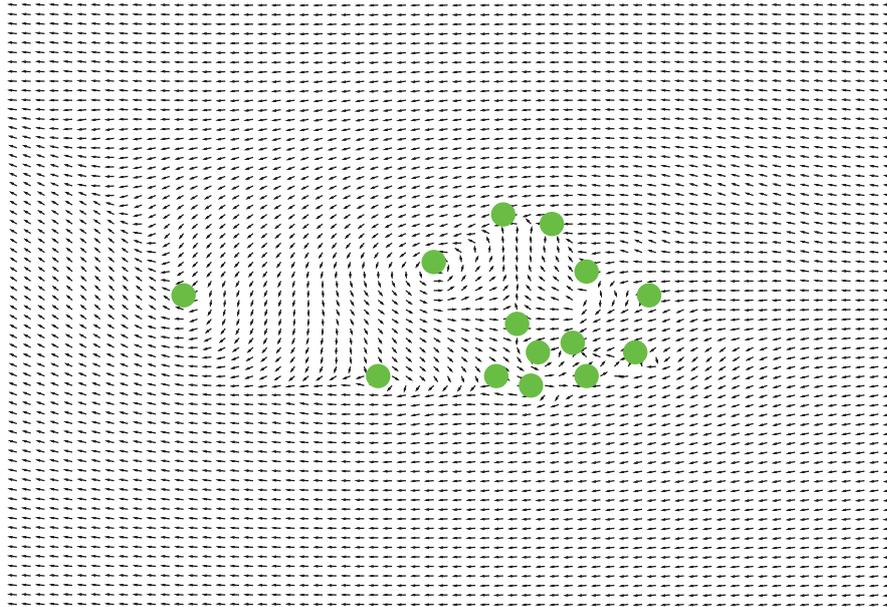


FIGURE 7.10 – Détection de singularités sur le champ 100 de la séquence Cylindre-carré avec l’algorithme multi-résolutions à fenêtre mobile pour 11 niveaux de décomposition ; le champ est normalisé pour l’affichage ; les bases A-complètes de niveau 1 sont de 16×16 polynômes et de 15×15 polynômes pour le niveau 2 ; la fenêtre d’estimation est de 5×5 vecteurs

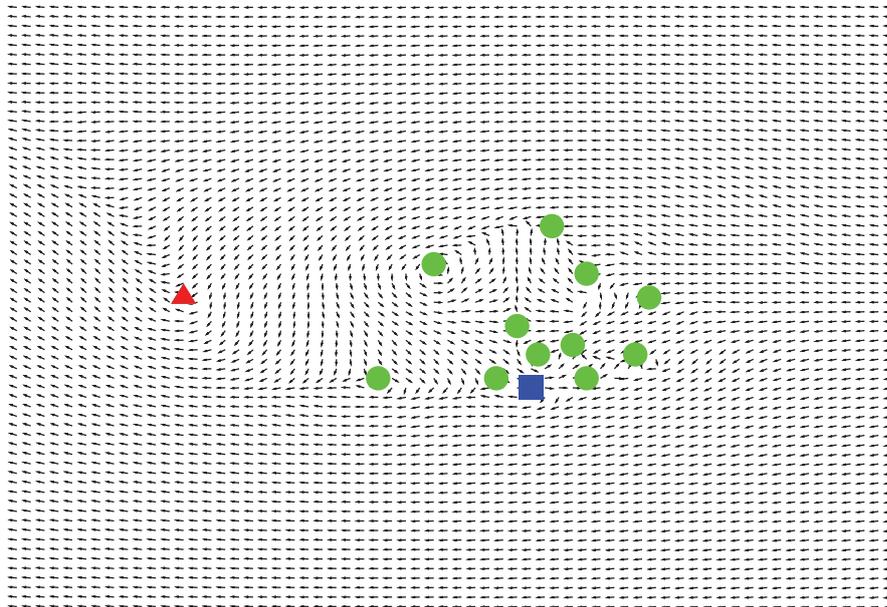


FIGURE 7.11 – Évolution de la détection de singularités sur le champ 100 de la séquence Cylindre-carré avec l’algorithme multi-résolutions à fenêtre mobile sur 3 niveaux de décomposition ; le champ est normalisé pour l’affichage ; les bases A-complètes de niveau 1 sont de 16×16 polynômes et de 15×15 polynômes pour le niveau 2 ; la fenêtre d’estimation est de 5×5 vecteurs ; les singularités détectées à la résolution 0 sont représentées par des disques verts ; les singularités détectées à la résolution 1 sont représentées par des carrés bleus ; les singularités détectées à la résolution 2 sont représentées par des triangles rouges ;

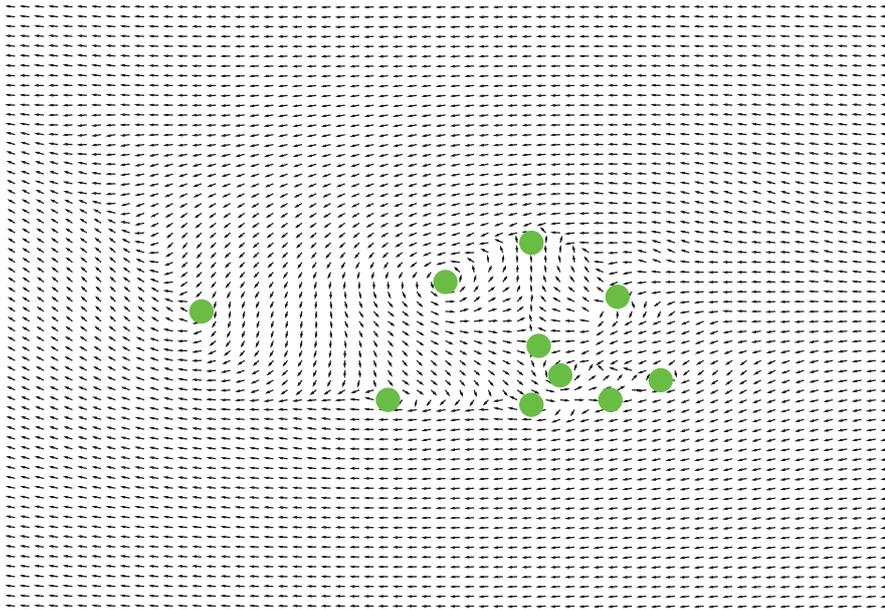


FIGURE 7.12 – Détection de singularités sur le champ 100 de la séquence Cylindre-carré avec l’algorithme par quadtree pour 6 niveaux de décomposition ; le champ est normalisé pour l’affichage ; Les disques verts représentent les singularités détectées

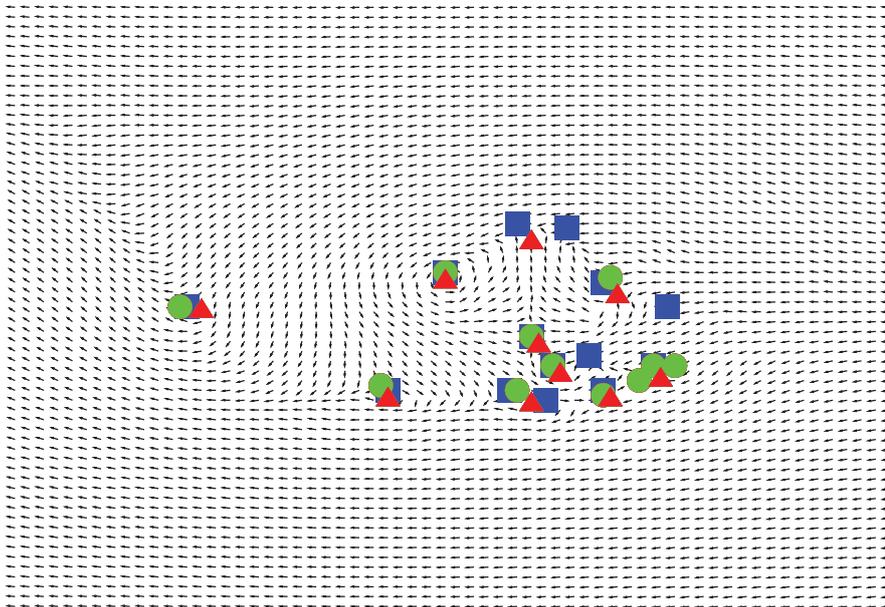


FIGURE 7.13 – Synthèse des algorithmes de détection de singularités sur le champ 100 de la séquence Cylindre-carré ; le champ est normalisé pour l’affichage ; les disques verts représentent les résultats obtenus avec l’algorithme par fenêtre adaptative pour une fenêtre minimum de 3×3 vecteurs et une fenêtre maximum de 9×9 vecteurs ; les carrés bleus représentent les résultats obtenus avec l’algorithme multi-résolutions à fenêtre mobile sur 7 résolutions pour une fenêtre de 5×5 vecteurs ; les triangles rouges représentent les résultats obtenus avec l’algorithme par quadtree sur 6 niveaux.

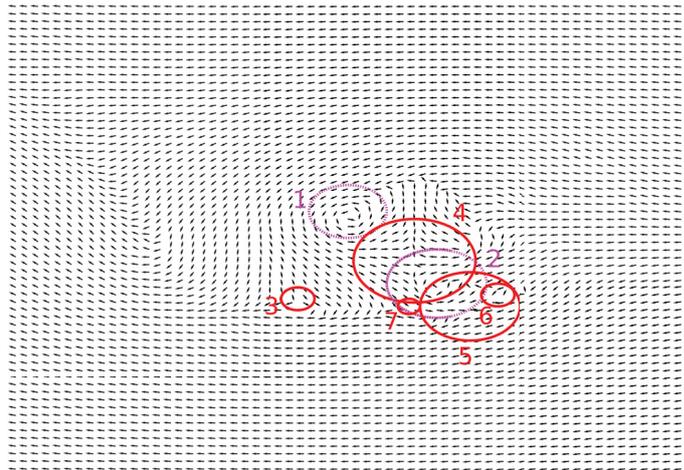
7.5.3 Résultats de l'estimation de la zone de linéarité

Nous présentons sur la figure FIG 7.14 les résultats d'estimation de la zone de linéarité. Sur la figure FIG 7.14(a), nous présentons les zones de linéarité obtenues à partir des points singuliers détectés par l'algorithme à fenêtre adaptative pour une fenêtre minimale de 3×3 vecteurs et une fenêtre maximale de 9×9 vecteurs. Ensuite, sur la figure FIG 7.14(b), nous présentons les zones de linéarité obtenues à partir des points singuliers détectés par l'algorithme multi-résolutions à fenêtre mobile pour une fenêtre de 5×5 vecteurs, la décomposition est effectuée sur 7 niveaux avec une base de niveau 1 de 16×16 polynômes et une base de niveau de 2 de 15×15 polynômes. Enfin, sur la figure FIG 7.14(c), nous présentons les zones de linéarité obtenues à partir des points singuliers détectés avec l'algorithme par quadtree pour une décomposition par quadtree sur 6 niveaux. Pour l'ensemble des résultats de nos trois algorithmes de détection, il existe des points singuliers pour lesquels l'algorithme d'estimation de la zone de linéarité ne trouve aucune zone. De plus, nous pouvons constater qu'en raison des différences de localisation (de quelques pixels), la zone de linéarité estimée peut varier de manière importante. C'est le cas de la structure numérotée 1. La différence de localisation peut aussi avoir une influence sur la classification de la structure, comme c'est le cas pour la structure numérotée 3. En effet, sur les figures FIG 7.14(a) et FIG 7.14(c), elle est classifiée comme un point de selle et sur la figure FIG 7.14(b), elle est classifiée comme une spirale. Cela signifie certainement que deux structures de nature différente sont très proches. Cependant, les résultats obtenus à partir de ces différents algorithmes sont très proches. Les zones de linéarité des structures les plus importantes sont donc estimées et classifiées par notre algorithme.

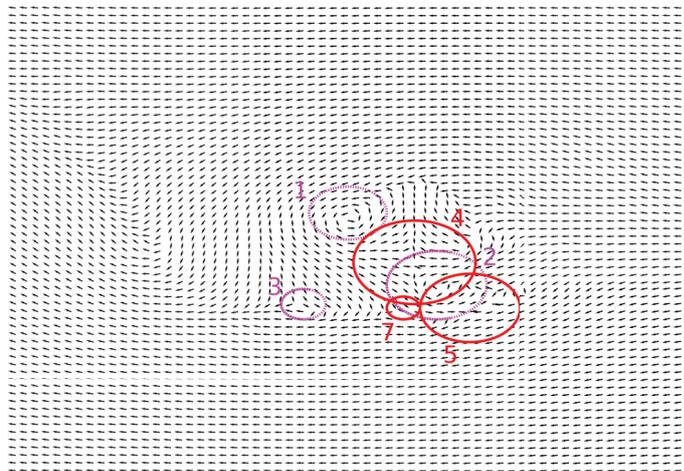
Nous comparons maintenant nos algorithmes de détection de singularités et d'estimation de la zone de linéarité avec deux méthodes issues de la littérature. Dans un premier temps, nous comparons nos techniques avec la méthode proposée dans [Corpetti 03]. Cette méthode est basée sur la détection des singularités par les extrema des fonctions de potentiels et sur une paramétrisation des structures par les modèles de RANKINE. Les résultats sont calculés sur un champ synthétique extrait de [Corpetti 03]. Ensuite, notre méthode est comparée à une méthode spécifique à l'étude de séquences de mécanique des fluides expérimentale. Cette technique présentée dans [Jardin 08], [Kihl 08b] est basée sur l'utilisation du critère λ_2 . Cette méthode ne permet de détecter que les tourbillons mais le critère λ_2 est un invariant Galiléen capable de détecter ces structures même noyées dans le mouvement global. Cela nous permet de comparer notre algorithme à fenêtre mobile en se plaçant dans un repère Galiléen.

7.5.4 Comparaison avec la détection par les fonctions de potentiels

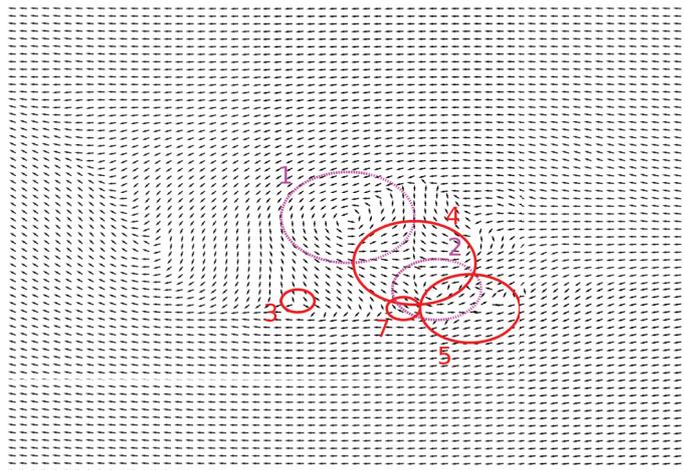
Nous comparons nos techniques avec la méthode proposée dans [Corpetti 03]. Cette méthode est basée sur la détection des singularités par les extrema des fonctions de potentiels (cf. section 7.3.4) et sur une paramétrisation des structures par les modèles de RANKINE (cf. 7.3.5.1). Dans [Corpetti 03], les auteurs proposent un champ synthétique avec les paramètres précis permettant de le calculer. Nous utilisons donc ce champ afin de nous confronter à leur méthode. Dans ce cas précis, tous nos algorithmes fournissent des résultats identiques. Nous comparons donc notre algorithme de détection par quadtree (cf. section 7.4.2.3) avec la méthode proposée dans [Corpetti 03]. Nous remercions T. CORPETTI de nous avoir fourni son algorithme pour effectuer des comparaisons.



(a)



(b)



(c)

FIGURE 7.14 – Résultats de l’algorithme d’estimation de la zone de linéarité ; (a) Résultat à partir des points singuliers détectés par l’algorithme à fenêtre adaptative pour une fenêtre minimale de 3×3 vecteurs et une fenêtre maximale de 9×9 vecteurs ; (b) Résultat à partir des points singuliers détectés par l’algorithme multi-résolutions à fenêtre mobile pour une fenêtre de 5×5 vecteurs, la décomposition est effectuée sur 7 niveaux avec une base de niveau 1 de 16×16 polynômes et une base de niveau de 2 de 15×15 polynômes ; (c) Résultats à partir des points singuliers détectés avec l’algorithme par quadtree pour une décomposition par quadtree sur 6 niveaux.

Structure	Position	Rayon
Source	(210,60)	20
Puits	(100,180)	30
Tourbillon	(200,350)	15
Tourbillon	(210,60)	50
Tourbillon	(50,50)	20
Tourbillon	(100,180)	25

TABLEAU 7.6 – Paramètres pour obtenir le champ de déplacements (396×276 vecteurs) présenté sur la figure FIG 7.15

Le champ synthétique utilisé pour ces tests est obtenu à partir des paramètres donnés dans le tableau TAB 7.5.4. Les dimensions de ce champ sont de 396×276 vecteurs. Il est composé d'un puits, d'une source et de quatre tourbillons. Il est à noter que le puits et la source sont tous les deux superposés à un tourbillon. Ce champ est représenté sur la figure FIG 7.15.

Nous appliquons donc l'algorithme de détection par quadtree présenté en section 7.4.2.3 sur le champ synthétique. Ensuite, pour toutes les singularités détectées, nous estimons la zone de linéarité qui lui est associée avec l'algorithme proposé en section 7.4.3. Nous obtenons les résultats présentés sur la figure FIG 7.16. On peut constater que tous les tourbillons ont été détectés, mais que les sources et les puits ne le sont pas. Les deux singularités où sont superposées une source (ou puits) et un tourbillon ont été détectées comme des spirales. Nous détectons aussi deux singularités de type points de selle qui ne sont pas détectées à partir des fonctions de potentiels.

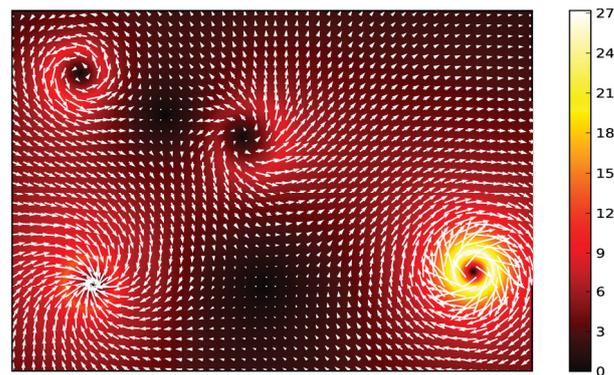
Cependant, lorsque deux structures sont superposées, l'estimation du domaine de linéarité n'est pas valide pour les deux singularités. Nous proposons donc une variante à notre algorithme. L'algorithme d'estimation de la zone de linéarité est effectué en deux passes. La première permet d'estimer le domaine de linéarité de la partie purement rotationnelle du champ. Cela permet d'estimer le domaine de linéarité d'une singularité lorsque celle-ci est associée à un tourbillon. La deuxième passe permet d'estimer le domaine de linéarité de la partie purement divergente du champ, c'est-à-dire les structures de type puits ou sources.

Pour la partie purement rotationnelle du champ, l'estimation est effectuée ainsi :

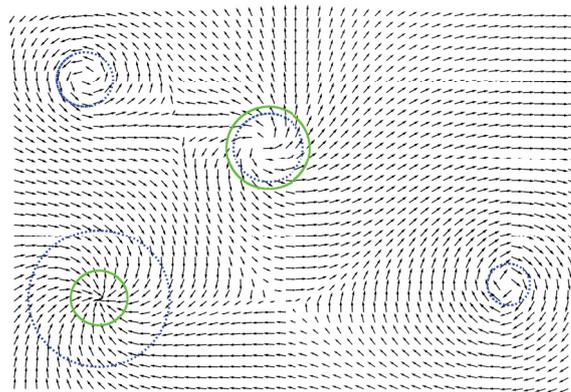
À partir de la position pos d'une singularité :

On initialise $rot_{buff} = 0$ et $continu = FALSE$, puis

1. On projette le champ sur une base éparsée définie sur un support spatial circulaire de rayon n pixels ;
2. On transforme ces coefficients dans la base canonique afin d'obtenir les coefficients sous la forme $Ax + b$;
3. On estime la position $p\tilde{o}s$ de la singularité ;
4. On calcule $\Delta(A)$ et $|rot(A)| = |a_{1,2} - a_{1,1}|$;
 - **si** ($\Delta(A) < 0$ **ET** $|rot(A)| > rot_{buff}$)
 - $continu = TRUE$
 - $rot_{buff} = |rot(A)|$
 - **sinon**
 - $continu = FALSE$

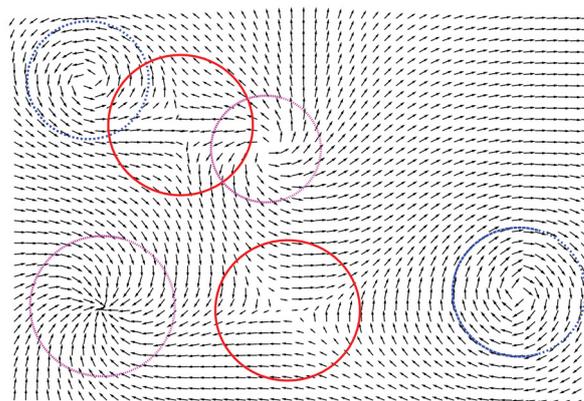


(a)



(b)

FIGURE 7.15 – Champ de déplacements synthétique extrait de [Corpetti 03] créé à partir des paramètres présentés dans le tableau TAB 7.5.4 ; (a) Champ de déplacements superposé à sa carte de norme en couleur ; (b) Champ de déplacements normalisé pour une meilleure lecture ; Les cercles verts (traits continus) superposés au champ représentent les sources et les puits ; Les cercles bleus (traits discontinus) superposés au champ représentent les tourbillons.



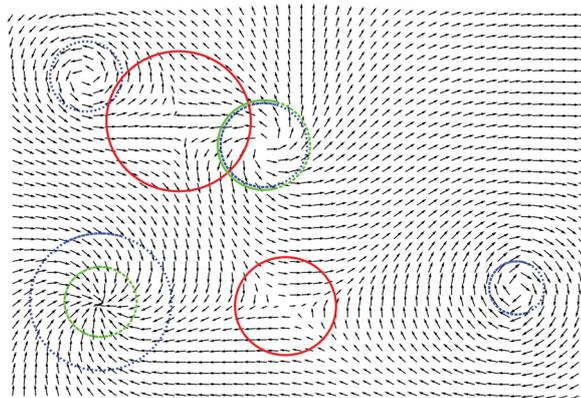
(a)

FIGURE 7.16 – Résultats de l'algorithme de quadtree sur le champ de déplacements synthétique **FIGURE.7.15** ; Les cercles rouges (traits continus) superposés au champ représentent les points de selles ; Les cercles bleus (traits discontinus) superposés au champ représentent les centres ; Les cercles violets (pointillés) superposés au champ représentent les spirales ;

5. *si* la distance entre pos et \tilde{pos} est inférieure à un seuil $dist_{max}$, retourner à l'étape 1
6. *sinon* le domaine de linéarité est un cercle de rayon $n - 1$ et le rotationnel associé est rot_{buff}

Sur le même principe, on construit l'algorithme dans le cas de la détection des structures purement divergentes.

Nous présentons en figure FIG 7.17, les résultats obtenus après les deux passes sur le champ synthétique. Ici $n = 2$ à l'initialisation et le seuil $dist_{max} = 3$. On constate que cette méthode permet une détection, une localisation et un paramétrage de la zone de linéarité d'une singularité. Sur les tableaux TAB 7.5.4, TAB 7.8 et TAB 7.9, nous comparons nos résultats à ceux obtenus dans [Corpetti 03] pour ce même champ. Ces tableaux permettent de comparer respectivement les positions obtenues (TAB 7.5.4), le rayon de la structure associée à une singularité détectée (TAB 7.8) et la force de la structure (TAB 7.9). On constate que toutes les structures sont bien détectées et que les résultats sont assez proches de ceux obtenus dans [Corpetti 03]. Nos résultats sont globalement un peu moins précis mais il est important de noter que les structures du champ sont générées à partir du modèle de RANKINE, qui est aussi utilisé pour paramétrer les structures dans [Corpetti 03]. De plus, notre méthode est rapide, puisqu'elle ne nécessite pas de décomposition en composante solénoïdale et irrotationnelle du champ, ni d'une étape de paramétrisation des structures (modèle de RANKINE). Sur le tableau TAB 7.9, nous présentons une estimation de la force de la structure calculée par le rotationnel ou par la divergence du modèle affine à l'intérieur du domaine de linéarité de la structure. Ces quantités ne sont pas directement comparables à celles obtenues dans [Corpetti 03]. Cependant, elles permettent d'obtenir des informations sur la force des structures les unes par rapport aux autres, ainsi que le sens de la rotation ou s'il s'agit d'un puits ou d'une source. Ces quantités sont donc riches en information.



(a)

FIGURE 7.17 – Résultats de l'algorithme de quadtree sur le champ de déplacements synthétique (cf. figure FIG 7.15 avec l'algorithme modifié ; les cercles rouges (traits continus) superposés au champ représentent les points de selles ; les cercles bleus (traits discontinus) superposés au champ représentent les centres ; les cercles verts (traits continus) superposés au champ représentent les sources et les puits ;

Nous proposons sur la figure FIG 7.18, un zoom sur la structure détectée à la position (97, 172) qui est la position pour laquelle l'erreur de détection est la plus grande. On constate que la position détectée semble cohérente avec la position que l'on perçoit visuellement.

structure	positions réelles	positions détectées dans [Corpetti 03]	positions détectées avec notre méthode
Source	(210,60)	(211,60)	(209,61)
Puits	(100,180)	(100,180)	(97,174)
Tourbillon	(200,350)	(201,350)	(199,349)
Tourbillon	(210,60)	(211,60)	(209,61)
Tourbillon	(50,50)	(50,49)	(48,51)
Tourbillon	(100,180)	(101,180)	(97,174)

TABLEAU 7.7 – Comparaison des positions des singularités obtenues par la méthode proposée dans [Corpetti 03] et par notre méthode

structure	positions réelles	rayon réel	rayon estimé dans [Corpetti 03]	rayon estimé avec notre méthode
Source	(210,60)	20	21	23
Puits	(100,180)	30	30	30
Tourbillon	(200,350)	15	15	17
Tourbillon	(210,60)	50	39	47
Tourbillon	(50,50)	20	20	23
Tourbillon	(100,180)	25	26	28

TABLEAU 7.8 – Comparaison de l'estimation des rayons des structures associées aux singularités détectées par la méthode proposée dans [Corpetti 03] et par notre méthode

En fait, notre méthode permet de détecter précisément les structures apparentes. En effet, le champ test est le champ résultant de la somme de six structures. Les positions des structures de ce champ peuvent donc être différentes des positions originales, à cause de l'effet des champs les uns sur les autres. De plus, l'écart entre nos résultats et ceux obtenus dans [Corpetti 03] peut s'expliquer en raison de la nature du champ. On peut toutefois noter que les structures présentes dans le champ sont générées à partir du modèle de RANKINE, lequel est aussi utilisé dans [Corpetti 03] pour paramétriser les structures.

Nous pouvons donc conclure que notre méthode permet d'obtenir une estimation rapide et relativement précise de la position de singularité. De plus, elle fournit une estimation du

structure	positions réelles	force réelle	force estimée dans [Corpetti 03]	indication de la force par notre méthode
Source	(210,60)	250	249.6	25.8
Puits	(100,180)	-150	-151.5	-9.5
Tourbillon	(200,350)	-400	-399.1	-27.8
Tourbillon	(210,60)	-250	-247.9	-5
Tourbillon	(50,50)	200	205.3	10.6
Tourbillon	(100,180)	150	149.8	5.8

TABLEAU 7.9 – Comparaison de l'estimation de la force des structures associées aux singularités détectées par la méthode proposée dans [Corpetti 03] et par notre méthode

domaine de linéarité des structures détectées. Bien que moins précise que la méthode présentée dans [Corpetti 03], notre méthode permet de détecter les points de selles qui ne peuvent pas être détectés avec les fonctions de potentiels.

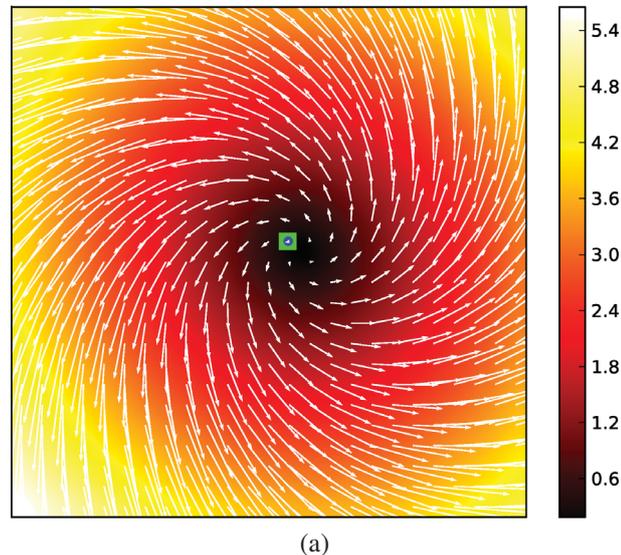


FIGURE 7.18 – Zoom sur la structure détectée à la position (97, 172) ; Le carré bleu superposé au champ représente le centre ; Le carré vert superposé au champ représente le puits ;

7.5.5 Comparaison avec le critère λ_2

Nous comparons ici notre méthode de détection utilisant l'algorithme à fenêtre adaptative et une méthode basée sur le critère λ_2 [Jardin 08], [Kihl 08b]. Cette dernière ne permet de détecter que les structures tourbillonnaires. Nous comparons donc uniquement la détection de structures tourbillonnaires. La détection de ces structures dans une séquence d'écoulement fluide nécessite soit d'utiliser un indicateur qui soit un invariant galiléen comme dans la première méthode, soit de se positionner dans un repère mobile se déplaçant avec la structure observée. Nous utilisons donc notre algorithme de détection à fenêtre mobile auquel nous apportons une modification afin d'utiliser un repère mobile pour l'estimation de la structure. Avant chaque estimation du modèle affine, la valeur du vecteur central de la fenêtre est retirée à l'ensemble des vecteurs de cette fenêtre. Cette modification intervient juste avant l'étape d'estimation du mouvement affine sur la figure FIG 7.2.

Cette étude est effectuée sur une séquence de 1022 champs de déplacements expérimentaux obtenus par PIV dans le sillage d'un cylindre de diamètre 60 mm plongé dans un écoulement d'air à 28 m/s, soit un nombre de Reynolds de 10.5×10^5 . Les séries de 1023 images proviennent d'une caméra Phantom V5 (8 bit, 1024×1024 px) cadencée à 1 kHz. L'illumination est réalisée par un laser Pegasus (532 nm 10 mJ). Les champs de déplacements recouvrent une zone d'étendue $3D \times 3D$ dans le sillage immédiat du cylindre.

Cette expérience est représentée sur le schéma de la figure FIG 7.19. Un traitement PIV itératif en dimension deux et fournissant un champ à deux composantes avec déformation de fenêtres est réalisé par le logiciel Lavisoin Davis 7.2. Les champs de vecteurs obtenus ont une taille de 128×128 mailles, pour une fenêtre d'interrogation finale de 32×32 px avec recouvrement de 75%.

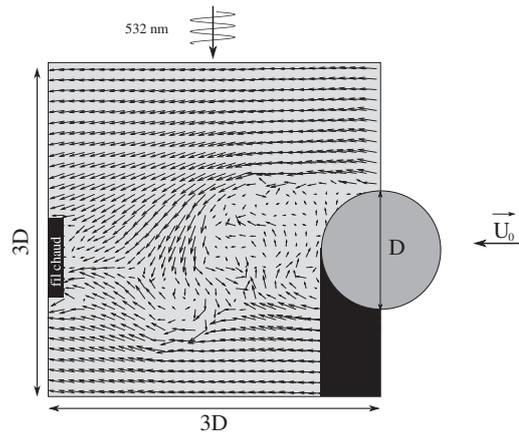


FIGURE 7.19 – Dispositif expérimental et aperçu du champ de déplacements correspondant.

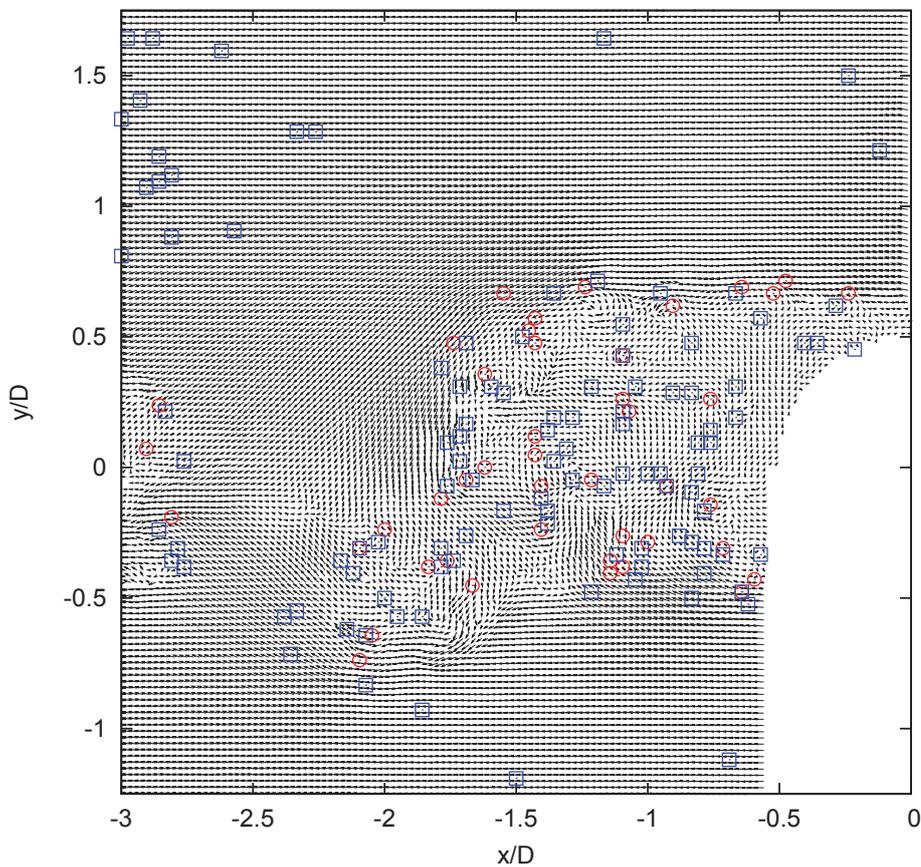


FIGURE 7.20 – Comparaison des méthodes : centres détectés par l'algorithme à fenêtre adaptative (\square) et tourbillons détectés par le critère λ_2 (\circ).

La figure FIG 7.20 représente les structures détectées par chacune des méthodes sur un champ de la séquence. Bien que comparables, le nombre et la position des structures observées diffèrent selon la méthode de détection employée, en particulier dans les zones hors sillage. Nous avons donc réalisé l'étude suivante. Lorsqu'une singularité est détectée par l'algorithme à fenêtre adaptative, alors localement, dans le voisinage de la singularité, le champ est reconstruit

à partir du modèle affine. Une nouvelle détection par la méthode fondée sur le critère λ_2 est appliquée à ce nouveau champ. Les résultats de cette étude sont alors présentés sur la figure FIG 7.21. Les différences de comportement des deux méthodes sont alors atténuées.

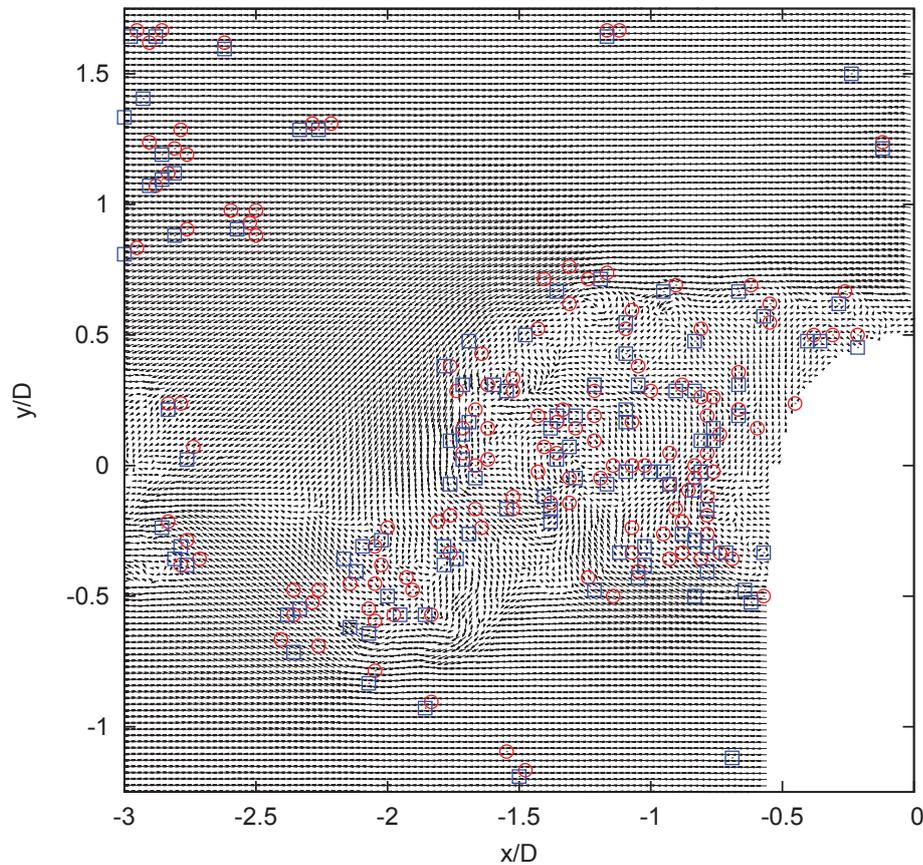


FIGURE 7.21 – Comparaison des méthodes sur un champ localement affine : centres détectés par l’algorithme à fenêtre adaptative (\square) et tourbillons détectés par le critère λ_2 (\circ) ; Le champ présenté ici est obtenu à partir du champ de la figure FIG 7.20 ; Pour chaque singularité repérée par un carré sur cette figure, le champ est reconstruit localement avec un modèle affine.

Nous présentons sur la figure FIG 7.22 un zoom sur deux structures, l’une détectée par l’algorithme à fenêtre adaptative et l’autre par la méthode fondée sur le critère λ_2 . On constate que la singularité détectée par l’algorithme à fenêtre adaptative est simple à identifier visuellement, tandis que l’autre l’est plus difficilement. Cependant, n’ayant aucune vérité de terrain, il est difficile de vérifier précisément si localement l’une ou l’autre des méthodes propose les meilleurs résultats.

Nous effectuons donc une étude statistique sur les 1022 champs de déplacements de la séquence. Nous calculons le cumul des positions détectées en tous points du champ au cours de la séquence. Nous représentons, sur la figure FIG 7.23(a), le cumul des positions détectées par l’algorithme à fenêtre adaptative et sur la figure FIG 7.23(b), par la méthode fondée sur l’utilisation du critère λ_2 . Ces figures confirment que notre méthode détecte beaucoup plus de structures, comme nous l’avons constaté sur la figure FIG 7.20. Il n’est toujours pas possible de déterminer s’il s’agit de bonne détection ou de faux positifs. Cependant, on constate que dans le sillage, les deux méthodes semblent faire apparaître la même zone d’intérêt. On peut donc en conclure que statistiquement, les deux méthodes fournissent des résultats proches.

7.6 Conclusion

Dans ce chapitre, nous avons proposé trois algorithmes multi-résolutions pour la détection des points singuliers dans un champ de déplacements. Ces trois algorithmes utilisent localement une estimation du modèle affine par projection sur une base de polynômes orthogonaux. Nous avons aussi proposé une méthode d'estimation de la zone de linéarité autour d'une singularité.

À partir des comparaisons que nous avons effectuées, nous pouvons en conclure que nos algorithmes fournissent des résultats comparables ou meilleurs à ceux obtenus avec les méthodes classiques de la littérature. De plus, notre méthode s'appuyant sur un formalisme mathématique générique permet d'envisager des applications à la détection de structures en dimension supérieure. De plus, nos algorithmes multi-résolutions s'appuient sur un estimateur du mouvement affine robuste au bruit gaussien et très rapide. Nos algorithmes peuvent être appliqués à des champs denses ou à des champs épars.

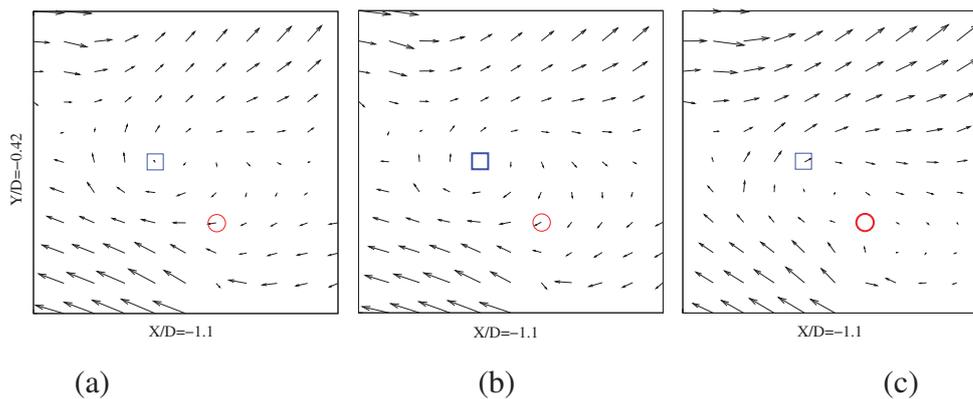
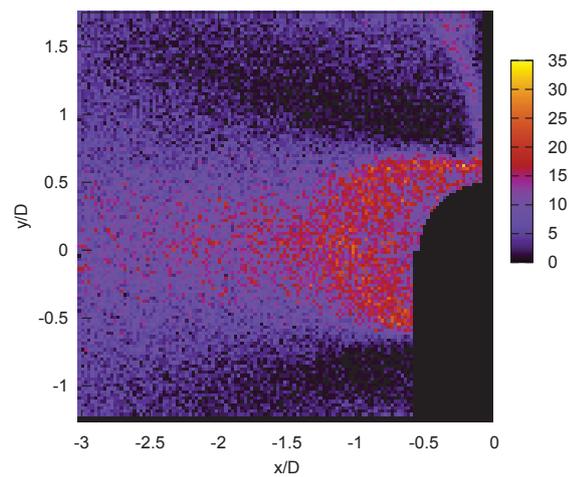
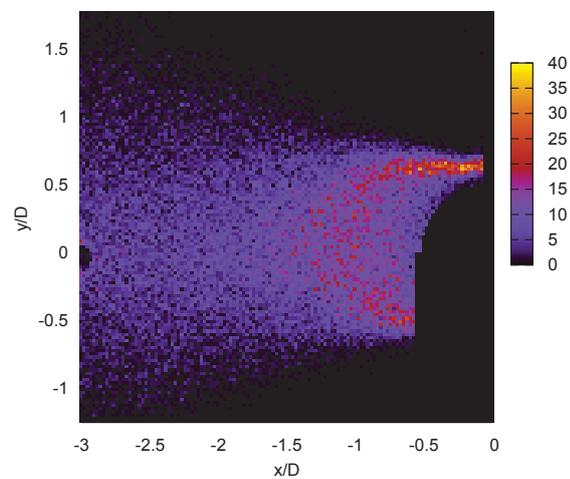


FIGURE 7.22 – Exemple de structures détectées et aperçu des champs de déplacements correspondant dans la fenêtre mobile attachée à chaque structure. (a) fenêtre de référence ; (b) fenêtre attachée à la structure détectée par l’algorithme à fenêtre adaptative (□) c’est-à-dire que nous avons retiré à l’ensemble des vecteurs de la fenêtre, la valeur du vecteur qui se trouve sur la position repérée par le carré bleu ; (c) fenêtre attachée à la structure détectée par la méthode fondée sur l’utilisation du critère λ_2 (○) c’est-à-dire que nous avons retiré à l’ensemble des vecteurs de la fenêtre, la valeur du vecteur qui se trouve sur la position repérée par le cercle rouge ;



(a)



(b)

FIGURE 7.23 – Cumul des tourbillons détectés en tout point du champ sur l'ensemble de la séquence ; (a) Algorithme à fenêtre adaptative ; (b) Méthode fondée sur l'utilisation du critère λ_2 ;

CHAPITRE 8

RECONNAISSANCE DU MOUVEMENT DANS LES SÉQUENCES VIDÉO

8.1 Introduction

Dans ce chapitre, nous présentons une méthode originale de détection et de reconnaissance de mouvements humains à partir de l'information dynamique contenue dans un champ de déplacements. Notre méthode repose sur l'utilisation du modèle polynomial étudié dans les chapitres précédents et d'une mesure de similarité définie à partir de ce modèle. La qualité d'une méthode de reconnaissance repose sur l'utilisation d'un descripteur cohérent et d'un classifieur de qualité. Ici nous portons uniquement notre attention sur l'étude des descripteurs de mouvement.

Dans un premier temps, nous décrivons les principales approches utilisées dans le cadre de la reconnaissance de mouvements humains. Ensuite, nous présentons notre descripteur ainsi qu'une mesure de similarité associée. Enfin, nous présentons deux applications. La première concerne la détection et la reconnaissance de cinq mouvements de boxe, effectués par un utilisateur se trouvant devant une caméra. Nous sommes alors dans un contexte proche de celui de la commande d'un jeu vidéo. La seconde concerne la classification des vidéos de la base de WEIZMANN [Blank 05].

8.2 État de l'art

Les applications mettant en oeuvre la reconnaissance de mouvements humains sont nombreuses. Les principaux domaines d'applications sont l'annotation de vidéos, la vidéo surveillance assistée par ordinateur, l'interaction homme machine, etc. Le mouvement humain peut être analysé à différents niveaux d'abstraction. Suivant la hiérarchie proposée dans [Moeslund 06], le mouvement peut être analysé en tant que primitives d'actions (bouger le bras), en tant qu'actions (donner un coup de poing) ou encore en tant qu'activités (boxer). Dans

notre cas, nous nous intéressons aux actions. Un état de l'art très complet a été réalisé par POPPE [Poppe 10]. Dans cet état de l'art, seules les méthodes permettant de traiter une grande variété d'actions et mettant en oeuvre l'étude du corps humain dans sa globalité sont étudiées. Nous faisons nous aussi ce choix. Nous ne présenterons donc pas les méthodes spécialisées à un type d'actions données, telles que l'étude de la marche humaine afin d'extraire des caractéristiques biométriques [Cuzzolin 06], [Elgammal 04], [Wang 07a], ou la reconnaissance de gestes [Mitra 07], [Erol 07]. POPPE présente à la fois les descripteurs d'action (modélisation) et les méthodes de classification. Nous faisons le choix, dans ce chapitre, de nous intéresser plus particulièrement aux descripteurs de l'action.

Dans cette section, nous présentons un état de l'art des méthodes de reconnaissance de mouvements humains. Dans une première partie, nous présentons quelques approches reposant sur des détecteurs de points, car celles-ci sont souvent utilisées en amont d'autres méthodes. Nous présentons ensuite quelques uns des modèles les plus répandus utilisant les silhouettes puisque ces modèles sont parfois utilisés en complément des méthodes s'appuyant sur les champs de déplacements. Enfin, nous présentons plus en détails les méthodes qui utilisent les champs de déplacements ou les champs de gradients d'une image.

8.2.1 Détecteur de points

De nombreuses méthodes ne reposent ni sur l'utilisation d'un descripteur de silhouette, ni sur un descripteur de mouvement, mais sur l'utilisation de détecteur de points. Nous ne souhaitons pas dresser un état de l'art de cette famille de descripteurs, mais ceux-ci sont parfois utilisés en amont de descripteurs de champs de déplacements. Par exemple, LAPTEV et al. [Laptev 08] proposent d'utiliser un descripteur spatio-temporel pour chaque point détecté par l'algorithme présenté dans [LAPTEV 05]. Ils définissent alors autour de chaque point un volume spatio-temporel décomposé avec une grille 3D dont chaque cellule est modélisée par un histogramme d'orientation du gradient (HOG) [Lu 06b] et par un histogramme d'orientation du flot optique (HOF) [Li 07] Ces deux descripteurs sont détaillés dans la section 8.2.3 par la suite.

Nous présentons donc quelques références où les détecteurs de points sont utilisés pour reconnaître des actions humaines. LAPTEV et LINDENBERG [Laptev 03] étendent le détecteur de coins de HARRIS [Harris 88] en 3D. OIKONOMOPOULOS et al. [Oikonomopoulos 05] utilisent une extension 3D du détecteur proposé dans [Kadir 03]. RAPANTZIKOS et al. [Rapantzikos 07] appliquent un détecteur s'appuyant sur une décomposition en ondelettes discrètes. SUN et al. [Sun 09] utilisent le descripteur SIFT [Lowe 04] et SCOVANNER et AL. [Scovanner 07] proposent une extension de ce descripteur en 3D. WILLEMS et al. [Willems 08] étendent le détecteur de points SURF [Bay 06] en 3D.

8.2.2 Méthodes reposant sur un modèle d'apparence de mouvement

Ces méthodes s'appuient sur un modèle de mouvement au sens de l'évolution d'une image au cours du temps. Nous traitons séparément les cas où cette évolution est estimée par le flot optique. Nous évoquons dans un premier temps, les travaux qui utilisent une représentation permettant de suivre l'évolution d'une silhouette au cours de l'action, appelée modèle d'apparence de mouvement. Ces descripteurs sont très utilisés. La majorité de ces descripteurs sont calculés à partir de la différence entre images (DOF pour Difference Of Frame) donnée par :

$$D(x_1, x_2, t) = \begin{cases} 1 & \text{si } |I(x_1, x_2, t) - I(x_1, x_2, t - 1)| > \Gamma \\ 0 & \text{sinon} \end{cases} \quad (8.1)$$

où Γ est un seuil.

À partir de cette DOF, un descripteur représentant une image de l'énergie du mouvement (MEI pour Motion Energy Image) est proposé par BOBICK et DAVIS [Bobick 96]. Le MEI est obtenu par :

$$MEI_N(x_1, x_2, t) = \bigcup_{i=0}^{N-1} D(x_1, x_2, t - i) \quad (8.2)$$

Ils ajoutent ensuite un autre descripteur s'appuyant sur la DOF. Ce descripteur représente "l'histoire" du mouvement (MHI pour Motion History Image) [Davis 97]. Il est obtenu par :

$$MHI_N(x_1, x_2, t) = \max\left\{\bigcup_{i=0}^{N-1} D(x_1, x_2, t - i) \times (N - i)\right\} \quad (8.3)$$

Les deux descripteurs sont utilisés comme deux composantes d'une image vectorielle. Chaque composante est décrite avec sept moments de HU [Hu 62]. Afin de reconnaître un mouvement, la distance de Mahalanobis est calculée sur les moments de HU du MEI et MHI de ce mouvement et ceux de tous les mouvements de la base de test. KELLOKUMPU et al. [Kellokumpu 08b] utilisent des attributs de caractérisation de textures pour modéliser les images MHI et MEI et reconnaître les actions humaines. Pour cela, ils utilisent comme modèle des histogrammes du LBP (Local Binary Patterns) [Ojala 02]. Le LBP est défini pour un bloc de l'image à partir du pixel central du bloc. Ensuite, on regarde pour tous les pixels de ce bloc s'ils sont supérieurs (ou égaux) ou strictement inférieurs à ce pixel central. Leur valeur dans le LBP sera alors respectivement 1 ou 0. En lisant les pixels du bloc dans un ordre donné, on peut alors créer un vecteur contenant ces valeurs binaires. Ce vecteur est appelé LBP. Cette méthode est étendue dans [Kellokumpu 08a]. Ils proposent de modéliser les actions humaines avec un descripteur utilisé pour décrire les textures dynamiques le LBP-TOP (Local Binary Patterns from Tree Orthogonal Planes qui est une version spatio-temporelle du LBP qui consiste à calculer le LBP sur chacun des trois plans (Ox_1x_2 , Ox_1t , Ox_2t) d'une séquence d'image avec des voisinages différents selon le plan. Cette représentation, LBP-TOP, a été utilisée avec succès pour la reconnaissance d'expression faciale [Zhao 07]. Le volume spatio-temporel représentant une action est décomposé en plans x_1x_3 et x_2x_3 . Sur chaque plan, le LBP est calculé, directement à partir des images de la vidéo, puis modélisé par un histogramme. La concaténation de ces histogrammes forme le LBP-TOP. Afin d'obtenir une représentation plus locale, le volume spatio-temporel est décomposé en sous-volumes. Par ailleurs, WEINLAND et al. [Weinland 05], [Weinland 06] étendent le descripteur MHI à partir de représentations 3D des silhouettes. Ces représentations sont obtenues sur des séquences vidéos réalisées à l'aide de 5 caméras calibrées. Ce descripteur est nommé MHV (pour Motion History Volume).

Dans un autre article, WEINLAND et al. [Weinland 07] utilisent pour seul descripteur les silhouettes 3D des différentes postures d'une action. Ces silhouettes sont obtenues par soustraction du fond. Il n'utilisent donc plus le MHV. Les actions sont alors reconnues à partir d'un classifieur de type modèle de Markov Cachés (Hidden Markov Model ou HMM) paramétré pour fournir la meilleure vue en 2D de la posture 3D la plus proche de celle analysée à un instant donné. Ces travaux sont étendus [Weinland 08], afin de faire correspondre les silhouettes de l'ensemble d'apprentissage aux contours de la silhouette lorsque la suppression de fond ne permet pas d'obtenir une silhouette satisfaisante en utilisant la distance de Chanfrein.

WANG et SUTER [Wang 07b], proposent d'utiliser deux descripteurs simultanément. Le premier est l'AME (Average Motion Energy) et le second le MMS (Mean Motion Shape). L'AME est un descripteur proche du MHI, défini par :

$$AME(x_1, x_2) = \frac{1}{\tau} \sum_{t=1}^{\tau} D(x_1, x_2, t) \quad (8.4)$$

Le MMS est défini à partir du contour de la silhouette. On se place dans un repère dont le centre est le centroïde de la silhouette. On peut alors définir un vecteur composé des k coordonnées complexes z_i des k points du contour $z = [z_1, z_2, \dots, z_k]^T$. Pour une action donnée, un ensemble de descripteurs de ce type est calculé à partir de chaque silhouette. Afin de rendre ce descripteur plus robuste au changement d'échelle ainsi qu'à de petites rotations, ils utilisent une méthode d'analyse procustéenne¹ [Boyd 01].

WANG et al. [Wang 07c] utilisent une \mathcal{R} -transform [Tabbone 06] sur les silhouettes obtenues par soustraction de fond. Cette transformée est basée sur la transformée de Radon. Elle permet d'obtenir un descripteur invariant aux changements d'échelle, et la rotation ne produit qu'un déphasage du signal de sortie. Cette méthode est étendue [Souvenir 08] en prenant en compte le temps comme troisième composante pour le calcul de la \mathcal{R} -transform. De plus, une métrique plus adaptée à ce descripteur est utilisée. Cette méthode permet de reconnaître simultanément le mouvement et la position de la prise de vue.

FUJIYOSHI et al. [Fujiyoshi], [Fujiyoshi 04] proposent un nouveau modèle appelé squelette en étoile ("star skeleton"). Ce modèle est obtenu à partir des contours d'une silhouette. Dans un premier temps, on calcule le centre de gravité des points du contour puis la distance entre ce centre et chaque point du contour. En suivant le contour dans un sens donné, on crée une fonction discrète dont chaque point est associé à la distance entre un point du contour et la position du centre de gravité. On obtient alors une fonction périodique ayant comme période le nombre de points du contour. Cette fonction est lissée dans le domaine de Fourier. Ensuite, dans le domaine temporel, on estime les maxima locaux de cette fonction. On obtient alors les positions sur le contour des extrémités des différentes branches de ce squelette en étoile. En général, ces points représentent les membres et la tête. Une action humaine est alors définie comme une suite de squelettes en étoile. Cette méthode est étendue [Chen 06] afin d'ajuster automatiquement à 5 le nombre de branches de l'étoile, grâce à un filtre passe bas particulier.

Dans leurs travaux, GORELICK et al. [Blank 05] et [Gorelick 07] proposent d'étudier les silhouettes comme des volumes spatio-temporels. Pour cela, ils étendent en 3 dimensions les travaux de Gorelick et al. [Gorelick 06] sur la représentation et la classification de forme (en 2 dimensions) à partir des équations de Poisson. Ils extraient ainsi 7 composantes caractéristiques spatio-temporelles. La première est la saillance spatio-temporelle. Ensuite, 6 composantes sont calculées à partir des valeurs propres du Hessien 3×3 des solutions locales des équations de Poisson. Pour représenter les caractéristiques globales, ils calculent les moments pondérés en 3 dimensions des silhouettes avec comme fonction de pondération l'une des 7 composantes. La concaténation des moments pondérés de chaque composante permet de former le descripteur d'une action.

ZHANG et al. [Zhang 08] proposent une adaptation d'un descripteur de forme appelé shape context [Belongie 00] en Motion context pour décrire le mouvement. La zone d'intérêt est dé-

1. L'analyse procustéenne est une méthode statistique de comparaison de forme. Le principe est de transformer une forme avec des transformations affines, jusqu'à ce qu'elle soit le plus proche possible d'une référence donnée. Ainsi, les différences qui subsistent sont considérées comme discriminantes.

tectée en calculant pour chaque pixel son écart type temporel au cours d'un nombre donné d'images (par exemple $(2 \times n) + 1$ images). On obtient alors une image où chaque pixel représente la variation de l'intensité lumineuse locale entre l'instant $t - n$ et l'instant $t + n$. La zone d'intérêt ainsi détectée est délimitée par un rectangle dans lequel on définit un nouveau repère en coordonnées polaires. L'origine O de ce repère est le centre de gravité du rectangle. Ensuite, plusieurs détecteurs de points sont appliqués sur la "silhouette". Grâce à des méthodes de partitionnement, un certain nombre de ces points sont conservés et transformés en mots visuels de mouvement. On forme ainsi un dictionnaire Motion Word Dictionary. Chacun de ces mots visuels de mouvement est ensuite représenté dans un système de coordonnées en trois dimensions. Le premier axe est représenté par l'angle formé entre l'origine O et un des points, le second axe par la distance entre O et un point, enfin le dernier axe est représenté par l'indice du mot visuel dans le dictionnaire. Cela forme le modèle pour un groupe d'images données, appelé contexte de Mouvement (Motion Context).

8.2.3 Méthodes reposant sur un descripteur de champ de déplacements

Nous présentons maintenant les approches qui utilisent l'évolution d'une image au cours du temps, par exemple, à partir du flot optique calculé pour un groupe d'images.

Les premiers travaux dans ce sens ont été proposés par POLONA et NELSON [Polana 94] pour reconnaître des activités relatives à des mouvements répétitifs tels que la marche ou la course. Ils calculent, dans un premier temps, un champ de déplacements entre les différentes images successives (à l'instant $t - 1$ et t) par une méthode de flot optique. Ce champ de déplacements est décomposé spatialement avec une grille. Dans chaque cellule de la grille, ils accumulent l'amplitude du mouvement au cours du temps, pour une durée donnée. Comme il traitent des mouvements répétitifs, ils supposent que les mouvements analysés sont périodiques. Finalement, ils utilisent un algorithme des k plus proches voisins pour reconnaître les activités. Cette méthode permet uniquement de détecter des activités périodiques, mais elle a l'avantage de fonctionner en temps réel.

EFROS et al. [Efros 03] utilisent l'algorithme de flot optique de LUCAS et KANADE pour calculer le flot optique entre les images. Le champ de déplacements alors obtenu est décomposé en composantes horizontale et verticale. Ces deux composantes sont à nouveau décomposées par une technique de redressement simple alternance, c'est-à-dire :

$$\mathcal{U}^+(\vec{x}) = \begin{cases} \mathcal{U}(\vec{x}) & \text{si } \mathcal{U}(\vec{x}) > 0 \\ 0 & \text{sinon} \end{cases}, \mathcal{U}^-(\vec{x}) = \begin{cases} \mathcal{U}(\vec{x}) & \text{si } \mathcal{U}(\vec{x}) < 0 \\ 0 & \text{sinon} \end{cases} \quad (8.5)$$

De la même manière, à partir de \mathcal{V} , on obtient \mathcal{V}^+ et \mathcal{V}^- . Les composantes sont ensuite floutées avec une gaussienne. L'image vectorielle à quatre composantes constitue le descripteur de mouvement. Des descripteurs ainsi obtenus sont comparés par une mesure de corrélation normalisée. À partir de cette mesure, une matrice de similarité est construite et la classification est effectuée en fonction de la matrice obtenant le meilleur score.

La méthode d'EFROS et al. [Efros 03] est aussi utilisée dans [Ahad 08] pour étendre les descripteurs MEI et MHI [Davis 97] (cf. section 8.2.3). Les 4 composantes du flot optique remplacent les silhouettes binaires dans le calcul du MEI et du MHI, renommés ici DMHI (Directional Motion History Image) et DMEI (Directional Motion energy Image). Finalement, le descripteur est obtenu à l'aide des moments de HU calculés sur le DMEI et le DMHI comme cela a été fait par BOBICK et DAVIS [Davis 97]. Ces nouveaux modèles permettent de résoudre

le problème d'occlusion du mouvement d'une action par une autre action de la même personne. Par exemple, dans le cas où une personne s'assoit puis se relève, le mouvement de l'action *se relever* "efface" l'action de *s'asseoir* dans modèle de type MEI et MHI. Finalement, le DMHI est comparé au MHI sur une base de tests de 10 mouvements d'aérobic effectués par 8 personnes ayant des propriétés physiques différentes. D'après leurs résultats, les performances de ce modèle sont bien meilleures que celles du MHI sur cette base de tests.

Les travaux de EFROS et al. [Efros 03] sont encore étendus par DANAFAR et al. [Danafar 07]. Leur méthode nécessite une estimation de la silhouette qui est effectuée en deux étapes. La première étape consiste à utiliser le détecteur de coins de Harris pour rechercher la zone où se trouve la silhouette. Une boîte englobante est alors calculée autour de cette zone. Ensuite, à l'intérieur de cette boîte, le fond et la silhouette sont séparés à l'aide de l'algorithme de partitionnement de données K-means. La silhouette est alors décomposée en 3 parties horizontales. La première partie est de un cinquième de la hauteur de la silhouette et correspond à la tête et aux épaules. La deuxième et la troisième parties sont de deux cinquièmes, d'abord pour le corps et les bras puis pour les jambes. Le flot optique correspondant à chaque partie de la silhouette est modélisé par la méthode proposée par EFROS et al. [Efros 03]. Cela permet d'obtenir des informations plus locales.

IKIZLER et al. [Ikizler 08] utilisent conjointement l'information du flot optique et de la silhouette. Le modèle de mouvement utilisé est celui proposé par EFROS et al. [Efros 03]. Cependant, ici, la silhouette est décomposée avec une grille régulière et le modèle est calculé dans chaque cellule. On obtient ainsi un vecteur par cellule. Ces vecteurs sont ensuite concaténés pour former le descripteur de mouvements. La silhouette est détectée par une méthode d'estimation de la probabilité des contours [Martin 04] reposant sur l'utilisation du détecteur de contours de Canny. Deux types de lignes (courtes ou longues) sont extraites des contours en utilisant la transformée de Hough. Ces lignes sont ensuite modélisées avec un histogramme représentant l'orientation de cette ligne. Les orientations sont quantifiées avec un pas de 15° . Cette modélisation est effectuée dans chaque cellule d'une grille régulière décomposant la silhouette. La concaténation de tous les histogrammes de la grille constitue un vecteur permettant de modéliser la silhouette. Les dimensions de ce vecteur sont réduites par une méthode mesurant l'entropie des vecteurs afin de ne sélectionner que les éléments les plus informatifs de ce vecteur. Un troisième modèle est utilisé afin d'obtenir le descripteur de l'action. Ce dernier est modélisé par une gaussienne représentant la vitesse moyenne des champs de la séquence. La classification d'une action est alors obtenue en deux étapes, la première étape consiste à extraire un sous-ensemble de solutions à partir du troisième modèle (gaussienne) par un test probabiliste avec un seuillage. Ensuite, un classifieur de type Séparateurs à Vaste Marge (SVM) est utilisé pour trouver la meilleure solution dans ce sous-ensemble à partir des descripteurs de silhouette et de mouvement.

FATHI et al. [Fathi 08] utilisent un classifieur ayant deux niveaux de reconnaissance. Leur algorithme analyse des volumes spatio-temporels définis autour d'une silhouette humaine et centrés sur celle-ci pour 5 images temporellement. Au premier niveau, ils utilisent le descripteur de EFROS et al. [Efros 03] auquel ils ajoutent une cinquième composante. Cette dernière est calculée à partir de la norme L^2 des quatre autres. Au second niveau de reconnaissance, des descripteurs plus locaux sont créés à partir de combinaisons pondérées de portions du descripteur du premier niveau.

ALI et SHAH [Ali 10] calculent les composantes cinématiques du champ de déplacements, telles que la divergence (cf. équation (8.6)), la vorticité (cf. équation (8.7)), le champ de défor-

mation symétrique (cf. équation (8.8)) et le champ de déformation antisymétrique (cf. équation (8.9)). Les composantes de ces deux champs sont utilisées comme des composantes du modèle.

$$\operatorname{div}(x_1, x_2) = \frac{\partial \mathcal{U}(x_1, x_2)}{\partial x_1} + \frac{\partial \mathcal{V}(x_1, x_2)}{\partial x_2} \quad (8.6)$$

$$\operatorname{vort}(x_1, x_2) = \frac{\partial \mathcal{V}(x_1, x_2)}{\partial x_1} - \frac{\partial \mathcal{U}(x_1, x_2)}{\partial x_2} \quad (8.7)$$

$$\mathcal{C}_{sym} = \begin{cases} \mathcal{U}_{sym}(x_1, x_2) = \mathcal{U}(x_1, x_2) + \mathcal{U}(x_1, x_2)^s \\ \mathcal{V}_{sym}(x_1, x_2) = \mathcal{V}(x_1, x_2) + \mathcal{V}(x_1, x_2)^s \end{cases} \quad (8.8)$$

où $\mathcal{U}(x_1, x_2)^s$ et $\mathcal{V}(x_1, x_2)^s$ sont obtenues par symétrie des matrices représentant les composantes \mathcal{U} et \mathcal{V} à un instant donné par rapport à un axe donné. Lorsque les images sont carrées, cette symétrie est obtenue par transposition, sinon l'axe de symétrie peut devenir l'axe central selon x_1 ou selon x_2 .

$$\mathcal{C}_{antisym} = \begin{cases} \mathcal{U}_{antisym}(x_1, x_2) = \mathcal{U}(x_1, x_2) - \mathcal{U}(x_1, x_2)^s \\ \mathcal{V}_{antisym}(x_1, x_2) = \mathcal{V}(x_1, x_2) - \mathcal{V}(x_1, x_2)^s \end{cases} \quad (8.9)$$

En plus de ces 6 composantes, ils ajoutent deux nouvelles quantités cinématiques calculées à partir du gradient du champ de déplacements calculé localement en tous points du champ de déplacements par le Jacobien suivant :

$$\nabla \mathcal{C}(x_1, x_2, x_3) = \begin{pmatrix} \frac{\partial \mathcal{U}(x_1, x_2)}{\partial x_1} & \frac{\partial \mathcal{U}(x_1, x_2)}{\partial x_2} \\ \frac{\partial \mathcal{V}(x_1, x_2)}{\partial x_1} & \frac{\partial \mathcal{V}(x_1, x_2)}{\partial x_2} \end{pmatrix} \quad (8.10)$$

À partir de ce Jacobien, ils calculent le second invariant Q (cf. équation (8.11)) et le troisième invariant R (cf. équation (8.12)).

$$Q(x_1, x_2) = \frac{1}{2}(P^2 - \operatorname{trace}(\nabla \mathcal{C}(x)^2)) \quad (8.11)$$

avec P le premier invariant défini par $P = -\operatorname{trace}(\nabla \mathcal{C}(x))$ et trace l'opérateur calculant la trace d'une matrice.

$$R(x_1, x_2) = -\det(\nabla \mathcal{C}(x)) \quad (8.12)$$

avec \det l'opérateur calculant le déterminant d'une matrice.

Ils ajoutent encore 2 quantités cinématiques basées sur le tenseur de déformation S , le second et le troisième invariant. Enfin, la dernière quantité du modèle est le troisième invariant du tenseur de rotation O . Ces tenseurs sont obtenus à partir du tenseur $\nabla \mathcal{C}$ par les équations (8.13) et (8.14) :

$$S = \frac{1}{2}(\nabla \mathcal{C}(x) + \nabla \mathcal{C}(x)^T) \quad (8.13)$$

$$O = \frac{1}{2}(\nabla \mathcal{C}(x) - \nabla \mathcal{C}(x)^T) \quad (8.14)$$

où T est l'opérateur de transposition de matrice.

Pour chacune des ces composantes, les modes cinématiques de l'action sont calculés par une analyse en composantes principales spatio-temporelles de type snapshot [SIROVICH 87].

C'est-à-dire que chaque composante spatio-temporelle en trois dimensions est résumée à un certain nombre de modes en deux dimensions. Ces modes représentent le modèle utilisé pour la classification. Grâce à une base d'apprentissage, plusieurs représentations d'une même action sont modélisées avec ce descripteur puis stockées.

LU et LITTLE [Lu 06b] proposent d'effectuer simultanément le suivi des personnes et la reconnaissance d'actions sur des vidéos de sports. Pour cela, ils utilisent des histogrammes de gradients orientés (HOG pour Histograms of Oriented Gradient). L'image est d'abord subdivisée avec une grille régulière, puis le HOG est calculé sur chaque cellule de cette grille. Ensuite, le suivi et la reconnaissance d'actions sont effectués à l'aide d'un HMM hybride. La méthode est testée sur la base de vidéos de football utilisée dans [Efros 03] puis sur la base de vidéos de Hockey sur glace utilisée dans [Okuma 04]. Ils étendent ces travaux dans [Lu 06a] en réduisant le volume de leur modèle HOG en utilisant une analyse en composantes principales. Les histogrammes de gradients orientés sont étendus en 3D dans [Kläser 08].

LI propose dans [Li 07] une méthode reposant sur des histogrammes, appliqués cette fois à l'orientation des champs de déplacements plutôt qu'au champ de gradients. Aucune technique d'extraction de silhouette n'est utilisée, mais cependant, un seuillage sur l'amplitude des vecteurs est appliqué afin d'isoler le mouvement de premier plan. Ensuite, un histogramme est calculé sur chaque champ. L'histogramme est calculé pour 90 orientations. Chaque représentant d'une orientation donnée est pondérée par l'amplitude du vecteur. L'amplitude et l'orientation sont calculées respectivement par les équations (8.15) et (8.16) :

$$\rho(x) = \sqrt{\mathcal{U}(x)^2 + \mathcal{V}(x)} \quad (8.15)$$

$$\theta(x) = \arctan\left(\frac{\mathcal{V}(x)}{\mathcal{U}(x)}\right) \quad (8.16)$$

Finalement les histogrammes sont normalisés de sorte que la somme de ces éléments soit égale à 1. Chaque action de l'ensemble d'apprentissage est décomposée en une suite composée d'autant d'histogrammes que la séquence compte de champs. La dimension des histogrammes est réduite de 90 à 5 par une analyse en composantes principales. Un descripteur similaire, composé de séries temporelles d'histogrammes d'orientation du flot optique, est proposé par CHAUDRY et al. [Chaudhry 09]. La dimension des histogrammes n'est pas réduite. Ces séries temporelles ne peuvent pas être étudiées par un système dynamique linéaire car les histogrammes ne sont pas dans un espace euclidien. Les auteurs proposent alors une généralisation au cas non linéaire des noyaux de Binet-Cauchy [Vishwanathan 07], afin de fournir une mesure pour comparer deux histogrammes.

TRAN et al. [Tran 08] utilisent le flot optique et une représentation binaire de la silhouette pour former le modèle. Pour un champ de déplacements donné, le mouvement est modélisé de la manière suivante. Chacune des composantes du flot optique est filtrée avec un filtre médian. Ensuite la composante horizontale, la composante verticale ainsi que l'image binaire de la silhouette sont subdivisées avec une grille de 2×2 cellules sans recouvrement. Chaque cellule est à nouveau décomposée spatialement, à partir du centre de la cellule, en 18 parties radiales couvrant chacune 20 degrés. Sur chacune de ces parties radiales, on calcule l'intégrale des trois composantes initiales (\mathcal{U} , \mathcal{V} et l'image binaire). On obtient ainsi pour chaque composante, 4 histogrammes composés de 18 classes. La concaténation des ces histogrammes fournit un vecteur à 216 dimensions qui constitue le modèle de mouvement d'un champ donné. À partir de ce modèle, on forme le descripteur de mouvement appelé contexte du mouvement. Il est composé du modèle de l'image courante auquel s'ajoute les 10 premiers vecteurs de l'ACP des modèles des

5 images précédentes, les 50 premiers vecteurs de l'ACP des modèles des 5 images courantes et enfin les 10 premiers vecteurs de l'ACP des modèles des 5 images suivantes.

YANG et al. [Yang 07] modélisent le flot optique avec une combinaison du descripteur SIFT (Scale-Invariant Feature Transform) [Lowe 04] et du descripteur LBP [Ojala 02]. Ils obtiennent ainsi un descripteur très robuste aux changements d'illumination et de pose pour un coût calculatoire relativement faible. Ensuite, la reconnaissance d'une action est effectuée par un algorithme de détection de la plus longue sous-séquence commune tel que cela est souvent utilisé pour la reconnaissance de phrase dans un texte. Cela permet de prendre en compte l'ordre temporel d'une action.

JIANG et MARTIN [Jiang 08] proposent un descripteur spatio-temporel permettant de combiner un descripteur de mouvement et un descripteur de forme (silhouette). Pour cela, ils adaptent une méthode de visualisation de champ de déplacements [Interrante 98]. Leur méthode consiste à extraire des flux de lignes par un algorithme de type ICM (Iterative Conditional Modes) [Besag 86]. Ici, l'ICM est utilisé pour réaliser une estimation locale du mouvement semblable à une méthode d'estimation du mouvement par corrélation, mais en permettant à la fois de minimiser le coût de la correspondance entre images et de maximiser la cohérence du mouvement dans un voisinage. L'ICM est appliqué sur le résultat d'une détection de contours de Canny. Les relations de voisinage sont définies par une triangulation de Delaunay des points de contours conservés. On obtient alors un champ épars, qui est interpolé au sein de chaque cellule du diagramme de Delaunay pour devenir dense. Ces champs de déplacements 2D sont alors concaténés pour former un descripteur volume spatio-temporel. La reconnaissance d'un mouvement revient à un problème de minimisation non linéaire et non-convexe, ce qui rend le problème très coûteux en temps de calcul. Ils proposent une méthode pour résoudre ce problème. Pour plus de détails concernant cette minimisation, consulter [Jiang 08].

KE et al. [Ke 05] proposent de décomposer une vidéo en volumes spatio-temporels. La classification est réalisée avec des descripteurs volumétriques calculés à partir de l'information du flot optique. La composante horizontale et la composante verticale du champ sont traitées séparément. Les auteurs étendent en 3D une méthode proposée par VIOLA et al. [Viola 01] pour effectuer une détection de visages dans des images. La vidéo est d'abord décomposée en une séquence de champs de déplacements. Pour leurs tests, ils utilisent des volumes composés de 64×64 vecteurs spatialement et pour 40 champs temporellement. Sur un volume donné, quatre mesures sont calculées. La première consiste à calculer l'intégrale sur le volume global. Ensuite, le volume est subdivisé en deux volumes égaux selon x_1 , x_2 et selon l'axe temporel. Pour chacune de ces trois subdivisions, on calcule la différence entre les intégrales des deux sous-volumes. On recalcule ces quatre mesures sur chacun des sous-volumes tant que cela est possible jusqu'à ce que les volumes atteignent la taille minimum de 4×4 vecteurs spatialement et pour 4 champs selon l'axe temporel. Pour la classification, ils utilisent une cascade de classifieurs, comme cela est proposé dans [Wu 04]. Les classifieurs traitent les composantes de façon marginale. Cette approche est étendue dans [Ogata 06]. Cependant, le champ de déplacements est décomposé en quatre composantes (\mathcal{U}^+ , \mathcal{U}^- , \mathcal{V}^+ , \mathcal{V}^-) suivies d'un lissage Gaussien tel que cela est proposé dans [Efros 03]. De plus, le classifieur utilisé dans [Ke 05] est amélioré afin de prendre en compte les différentes composantes (maintenant 4) du champ de façon vectorielle. Cela réduit le temps de calcul de la phase d'entraînement du classifieur.

SHECHTMAN et IRANI [Shechtman 05], proposent de comparer deux champs de déplacements sans avoir à les calculer. Pour cela, ils utilisent le formalisme des approches par tenseur de structure. Une vidéo est étudiée comme un volume spatio-temporel. Les champs sont mod-

élisés localement, pour un sous volume spatio-temporel V_i par une matrice (3×3) M_i définie par :

$$M_i = G_i^T G_i \quad \text{avec} \quad G_i = \begin{pmatrix} \partial_{x_1} V_i(x(1)) & \partial_{x_2} V_i(x(1)) & \partial_{x_3} V_i(x(1)) \\ \vdots & \vdots & \vdots \\ \partial_{x_1} V_i(x(i)) & \partial_{x_2} V_i(x(i)) & \partial_{x_3} V_i(x(i)) \\ \vdots & \vdots & \vdots \\ \partial_{x_1} V_i(x(n)) & \partial_{x_2} V_i(x(n)) & \partial_{x_3} V_i(x(n)) \end{pmatrix} \quad (8.17)$$

où $x(1)$ à $x(n)$ représentent tous les triplets (x_1, x_2, x_3) du sous-volume V_i . L'étude du rang de la matrice M permet de définir si le mouvement est cohérent sur le sous-volume étudié. En effet, si le rang est inférieur à 3 alors le mouvement est cohérent sur l'ensemble du sous volume, dans le cas contraire, le volume spatio-temporel est composé de plusieurs mouvements. SHECHTMAN et IRANI utilisent ce principe pour déterminer une mesure de ressemblance entre deux sous-volumes V_1 et V_2 . Pour cela, ils concatènent les matrices G_1 et G_2 associées respectivement aux sous-volumes V_1 et V_2 pour former la matrice G_{12} suivante :

$$G_{12} = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} \quad (8.18)$$

Ils peuvent alors calculer la matrice M_{12} suivante :

$$M_{12} = G_{12}^T G_{12} \quad (8.19)$$

À partir de l'étude des valeurs propres de M_{12} , ils déduisent une mesure de ressemblance entre deux sous-volumes. Ce modèle est utilisé dans [Ke 07b] pour décrire les zones faiblement texturées. En effet, ils constatent que les méthodes fondées sur un descripteur de silhouette sont plus robustes que les méthodes de reconnaissance s'appuyant sur un descripteur du flot optique dans les zones faiblement texturées, et inversement. Les zones faiblement ou fortement texturées sont extraites à partir du résultat du détecteur de Harris seuillé. Les zones fortement texturées sont modélisées par une pyramide de filtrages de type mean shift [Dementhon 06]. Ces travaux sont étendus dans [Ke 07a].

JHUANG et al. [Jhuang 07] proposent une méthode reposant sur une modélisation du processus de mouvement dans le cortex visuel². Le cortex visuel est organisé en deux voies différentes : une voie ventrale et une voie dorsale. On considère que la voie ventrale traite les informations de forme et de couleur et que la voie dorsale analyse des informations de mouvement. Ces deux voies ont une organisation hiérarchique très similaire. Dans un premier temps, trois modèles locaux sont calculés en chaque pixel de chaque image. Ils sont appelés unité $S1$. Ces unités sont calculées sur trois couches. La première couche modélise le gradient spatio-temporel [Zelnik-Manor 01]. La seconde modélise le flot optique (calculé avec l'algorithme de LUCAS et KANADE [Lucas 81b]) en fonction de directions ($0^\circ, 90^\circ, 180^\circ$ et 270°) et de vitesses privilégiées (3 ou 6 pixels) [Giese 03]. La troisième est modélisée par filtrages spatio-temporels avec des filtres obtenus par des dérivés troisième de Gaussienne 3D [Simoncelli 98]. Ensuite, une étape ($C1$) de détection des maxima locaux de chaque couche est effectuée avec une grille de 8×8 pixel. Cette détection est effectuée tous les 4 pixels. L'étape suivante ($S2$) consiste à retrouver dans les images issues de $C1$, 500 blocs $C1$ de $n \times n$ pixels ($n = 4, 8, 12$ et 16)

2. C'est la partie du cerveau qui est chargée de traiter les informations visuelles. Il se situe dans le lobe occipital [Sánchez 05].

8.3. Notre descripteur de mouvement fondé sur la projection sur une base de polynômes

extraits aléatoirement de la vidéo d'apprentissage. L'étape $C2$ consiste à retrouver les maxima locaux des pixels de $S2$, on obtient alors un scalaire. En empilant les réponses $S2$ d'une image pour toutes les couches, on obtient un vecteur. Afin de prendre en compte l'ordre temporel d'une action, les vecteurs successifs d'une action sont concaténés pour former une matrice ($S3$). Cette étape est suivie d'une nouvelle détection de maxima locaux ($C3$). La classification d'une vidéo est alors effectuée avec un classifieur de type SVM. Ces travaux sont étendus dans [Schindler 08] en prenant en compte simultanément la forme et le mouvement. ESCOBAR et al. [Escobar 09] utilisent aussi un modèle basé sur la perception du mouvement dans le système visuel humain. Pour cela, ils modélisent les fonctions de deux sous-régions fonctionnelles du cortex visuel humain, appelées zone V1 et zone MT. La zone V1 analyse le mouvement local, la zone MT le mouvement global. La zone V1 est modélisée en filtrant localement les images avec des filtres de Gabor spatio-temporels. Il y a donc un descripteur V1 par pixel. La zone MT modélise le mouvement à partir de descripteurs de la zone V1 pour former un descripteur plus global. Les descripteurs de V1 dans un voisinage circulaire donné sont utilisés comme entrées d'un réseau de neurones fournissant en sortie un descripteur MT. Le rayon de ce voisinage circulaire varie en fonction de sa position dans l'image. Plus le voisinage est éloigné du centre de l'image, plus le rayon est grand.

Bien que son domaine d'applications soit différent des approches précédentes, nous présentons le descripteur proposé par HOEY et LITTLE [Hoey 00], [Hoey 02]. Ils utilisent les polynômes de Zernike pour reconnaître les émotions sur des vidéos de visages, cette approche est très proche de celle que nous proposons dans la section suivante. Ici les fonctions polynomiales sont obtenues par projection dans une base de polynômes de Zernike (cf. section 2.3.4). Il s'agit de polynômes complexes définis sur le cercle unité qui forment une base orthogonale complète. En projetant un champ de déplacements sur cette base de polynôme, on obtient une approximation du champ de déplacements. Les auteurs estiment qu'une base composée de 11 polynômes de Zernike permet de modéliser les cinq expressions du visage étudiées. Notre méthode est plus générale car la famille de polynômes (réels) dépend du choix d'une fonction de pondération pour le produit scalaire choisi. De plus, le domaine de définitions de la base n n'est pas limité au cercle unité, il peut couvrir tout le champ de déplacements.

8.3 Notre descripteur de mouvement fondé sur la projection sur une base de polynômes

Nous avons présenté dans la section précédente plusieurs descripteurs utilisés en reconnaissance de mouvements humains. Nous avons donc vu que certaines approches utilisent l'information des champs de déplacements. Malgré le nombre important de méthodes, nous constatons qu'il existe peu de descripteurs s'appuyant directement sur un champ de déplacements. Les descripteurs utilisés sont souvent les mêmes, avec quelques adaptations. De nombreuses méthodes s'appuient sur le descripteur défini par EFROS et al. [Efros 03] et, par exemple, ajoutent une cinquième composante ou utilisent ce modèle localement dans une grille, ou encore autour de points particuliers. Les méthodes qui utilisent un même descripteur se différencient principalement par le classifieur utilisé. La seule approche qui utilise un descripteur obtenu par projection sur une famille de fonctions est la méthode de HOEY et LITTLE [Hoey 00],[Hoey 02], qui utilise les polynômes de Zernike. Cependant, à notre connaissance, ce descripteur n'a pas été utilisé pour la reconnaissance d'action humaine mais plus précisément pour la reconnaissance d'expressions sur des visages. Quelques autres approches utilisent une analyse en composantes

principales, mais ces décompositions interviennent souvent en aval d'un autre descripteur. De plus, nous savons qu'il est très difficile d'étudier les coefficients obtenus par analyse en composantes principales dans un cadre général. En effet, les coefficients n'ont de sens que pour la séquence à partir de laquelle ils ont été calculés.

Nous proposons ici un nouveau descripteur de mouvement qui utilise le formalisme des modélisations par série de fonctions polynomiales. Dans ce chapitre, nous modélisons le mouvement estimé dans des séquences vidéos représentant des actions humaines.

La première étape consiste à générer une base de polynômes orthogonaux (cf. chapitre 3). La projection d'une composante du champ sur un polynôme $P_{i,j}$ de la base est effectuée par

$$\tilde{u}_{i,j} = \frac{\langle \mathcal{U} | P_{i,j} \rangle_{\mathcal{E}}}{\langle P_{i,j} | P_{i,j} \rangle_{\mathcal{E}}} \quad (8.20)$$

Le champ de déplacements est donc approximé par une base de degré D par la formule :

$$\tilde{\mathcal{U}} = \sum_{i=0}^D \sum_{j=0}^{D-i} \tilde{u}_{i,j} P_{i,j} \quad (8.21)$$

La projection de la composante verticale du champ de déplacements \mathcal{V} est effectuée de manière similaire.

À partir de ces coefficients, nous construisons une empreinte selon la définition donnée par l'équation (8.22). Cette empreinte est une matrice composée de tous les coefficients polynomiaux de la composante horizontale du champ de déplacements ordonnés dans l'ordre lexicographique sur la première colonne et de tous les coefficients polynomiaux de la composante verticale du champ de déplacements ordonnés dans l'ordre lexicographique sur la seconde colonne. Il est à noter, que dans ce chapitre, nous utiliserons toujours des bases denses de polynômes de Legendre.

$$E = \begin{pmatrix} \tilde{u}_{0,0}(\mathcal{U}_n) & \tilde{v}_{0,0}(\mathcal{V}_n) \\ \vdots & \vdots \\ \tilde{u}_{i,j}(\mathcal{U}_n) & \tilde{v}_{i,j}(\mathcal{V}_n) \\ \vdots & \vdots \\ \tilde{u}_{0,n}(\mathcal{U}_n) & \tilde{v}_{0,n}(\mathcal{V}_n) \end{pmatrix} \quad (8.22)$$

Nous définissons la notion de quantité de mouvement \mathcal{Q} d'un champ de déplacements \mathcal{C} comme la valeur de la norme L_2 de l'empreinte de ce champ.

$$\mathcal{Q}(\mathcal{C}) = \|E\|_2 \quad (8.23)$$

Le rayon spectral β_+ d'une matrice à n valeurs propres est défini dans le cas d'une matrice E par

$$\beta_+(E) = \max_{i=1 \text{ à } n} \{|\lambda_i|\} \quad (8.24)$$

où $\max\{|\lambda_i|\}$ est la plus grande valeur propre en valeur absolue de la matrice E . On peut alors calculer la norme L^2 de la matrice E par

$$\mathcal{Q}(\mathcal{C}) = \sqrt{\beta_+(E^T E)} \quad (8.25)$$

8.3. Notre descripteur de mouvement fondé sur la projection sur une base de polynômes

avec T l'opérateur de transposition de matrice.

De même, nous pouvons définir la distance entre deux champs \mathcal{C}_1 et \mathcal{C}_2 comme

$$D(\mathcal{C}_1, \mathcal{C}_2) = \mathcal{Q}(\mathcal{C}_1 - \mathcal{C}_2) = \mathcal{Q}(E_1 - E_2) = \sqrt{\beta_+((E_1 - E_2)^T(E_1 - E_2))} \quad (8.26)$$

Cette mesure de distance prend en compte la quantité de mouvement propre à chaque champ. On peut souhaiter faire abstraction de cette quantité de mouvement afin d'obtenir une mesure de distance relative. Pour cela, on pose $\epsilon_r = \frac{E_1}{\mathcal{Q}(\mathcal{C}_1)} - \frac{E_2}{\mathcal{Q}(\mathcal{C}_2)}$

Nous proposons désormais une mesure de similarité entre les empreintes de mouvement. Initialement, le tenseur \mathcal{T} est défini à partir de deux empreintes E_1 et E_2 par

$$\mathcal{T}^{E_1, E_2} = E_1^T E_2 \quad (8.27)$$

On peut alors considérer la quantité suivante comme un produit scalaire

$$\beta_+(\mathcal{T}^{E_1, E_2}) \quad (8.28)$$

En normalisant par rapport à la quantité de mouvement de chaque champ, on obtient la mesure de la similarité entre les champs représentés par les deux empreintes E_1 et E_2 par l'équation (8.29) où E_1 et E_2 ne doivent pas être toutes les deux nulles.

$$S(E_1, E_2) = \frac{\beta_+(\mathcal{T}^{E_1, E_2})}{\sqrt{\beta_+(\mathcal{T}^{E_1, E_1})\beta_+(\mathcal{T}^{E_2, E_2})}} \quad (8.29)$$

Cette mesure de similarité vaut 0 si les empreintes sont orthogonales, 1 si elles sont identiques et -1 si elles sont opposées. La mesure est donc comprise dans $[-1, 1]$.

Sur la figure FIG 8.1, nous illustrons une mesure de similarité pour deux champs de déplacements. Ces champs sont extraits par flot optique avec l'algorithme de LUCAS et KANADE pour un voisinage de 5×5 pixels. Les images sont issues de vidéos de la base de WEIZMANN [Blank 05]. Sur les images FIG 8.1(a) et FIG 8.1(b), nous présentons deux images de deux personnes faisant un pas. Nous représentons ensuite sur les figures FIG 8.1(c) et FIG 8.1(d), les champs de déplacements entre ces deux images. Le champ de déplacements n'est calculé que sur une zone d'intérêt : la personne en mouvement. Ces champs de déplacements sont modélisés par projection sur une base dense de polynômes de Legendre bivariés de degré 7. Les reconstructions issues de cette modélisation sont présentées sur les figures FIG 8.1(e) et FIG 8.1(f). Ici la mesure de similarité entre ces deux champs est de 0.96. Les deux champs de déplacements avant modélisation semblent qualitativement assez proches, le résultat obtenu est alors assez évident. Le mouvement représenté par ces champs est un mouvement de marche, il est donc assez naturel que deux acteurs reproduisent assez fidèlement le même mouvement.

Nous comparons maintenant deux autres champs. Les mêmes utilisateurs effectuent différents mouvements dans la base de WEIZMANN. Nous choisissons donc le mouvement "jack". Il s'agit de sauter en écartant les pieds et les bras, puis de sauter à nouveau en resserrant les pieds et en joignant les mains au dessus de la tête (bras tendus). Ce mouvement est moins naturel et demande plus de coordination, on peut donc s'attendre à ce qu'il soit plus difficile à reproduire par deux acteurs différents. Nous présentons ce mouvement sur la figure FIG 8.2. Nous présentons sur les figures FIG 8.2(a) et FIG 8.2(b) les couples d'images. Il s'agit des mêmes personnes que sur les figures FIG 8.1(a) et FIG 8.1(b), mais pour un autre mouvement. Ensuite, nous présentons les champs de déplacements en FIG 8.2(c) et FIG 8.2(d). Ici encore, le mouvement est calculé dans la zone d'intérêt uniquement. Ces mouvements sont modélisés

avec une base de degré 7. Nous représentons en FIG 8.2(e) et FIG 8.2(f), les reconstructions issues de ces modélisations. Ici la mesure de similarité entre ces deux champs est de 0.80. Cette mesure reste cohérente avec ce que l'on peut constater qualitativement.

Enfin, nous présentons dans le tableau TAB 8.1 les résultats des mesures de similarités croisées entre ces quatre champs. À partir de ce tableau, nous constatons que lorsque les champs sont qualitativement proches, la mesure de similarité est importante et qu'elle est faible dans le cas contraire. En effet, les mesures de similarité entre les mouvements "walk" et "jack" ne dépassent jamais 0.16, ce qui est une mesure de similarité assez faible. Nous proposons deux applications qui utilisent cette mesure de similarité. La première est un algorithme de reconnaissance de mouvements humains dans les vidéos. La deuxième un algorithme de classification de vidéos d'actions humaines.

	<u>Daria_walk</u>	<u>Denis_walk</u>	<u>Daria_jack</u>	<u>Denis_jack</u>
<u>Daria_walk</u>	1	0.96	0.16	0
<u>Denis_walk</u>	0.96	1	0	0
<u>Daria_jack</u>	0.16	0	1	0.80
<u>Denis_jack</u>	0	0	0.8	1

TABLEAU 8.1 – Tableau récapitulatif des calculs de similarité entre les champs ; Daria_walk cf. figure FIG 8.1(c) ; Denis_walk cf. figure FIG 8.1(d) ; Daria_jack cf. figure FIG 8.2(c) ; Denis_jack cf. figure FIG 8.2(d) ;

8.4 Algorithme de reconnaissance de mouvements humains

8.4.1 Présentation de l'algorithme

Ici nous traitons de la reconnaissance de mouvements humains du point de vue de la vision par ordinateur. Nous cherchons à reconnaître un mouvement de référence dans une séquence vidéo. Dans cette application, nous supposons avoir à notre disposition une vidéo représentant un mouvement de référence. Ce mouvement de référence est modélisé par la méthode suivante. Nous donnons la méthodologie générale, puis ensuite, nous détaillons chaque étape. *Pour chaque couple d'images :*

1. Extraire la zone d'intérêt ;
2. Estimer le champ de déplacements ;
3. Calculer l'empreinte du champ de déplacements ;

À ce stade, nous disposons d'une séquence d'empreintes. Celle-ci est alors divisée en n groupes (G_1, G_2, \dots, G_n), n doit être choisi en fonction de l'application et des mouvements à reconnaître. Nous illustrons cette étape d'extraction des empreintes à partir de la séquence d'apprentissage sur la figure FIG 8.3.

Pour estimer le mouvement entre les images, nous utilisons l'algorithme de flot optique de LUCAS et KANADE [Lucas 81b]. Nous avons choisi d'utiliser cet algorithme parce qu'il fournit des résultats de bonne qualité tout en étant relativement rapide. De plus, cet algorithme a été utilisé dans [Le Besnerais 05] afin d'obtenir une méthode d'estimation du flot optique en

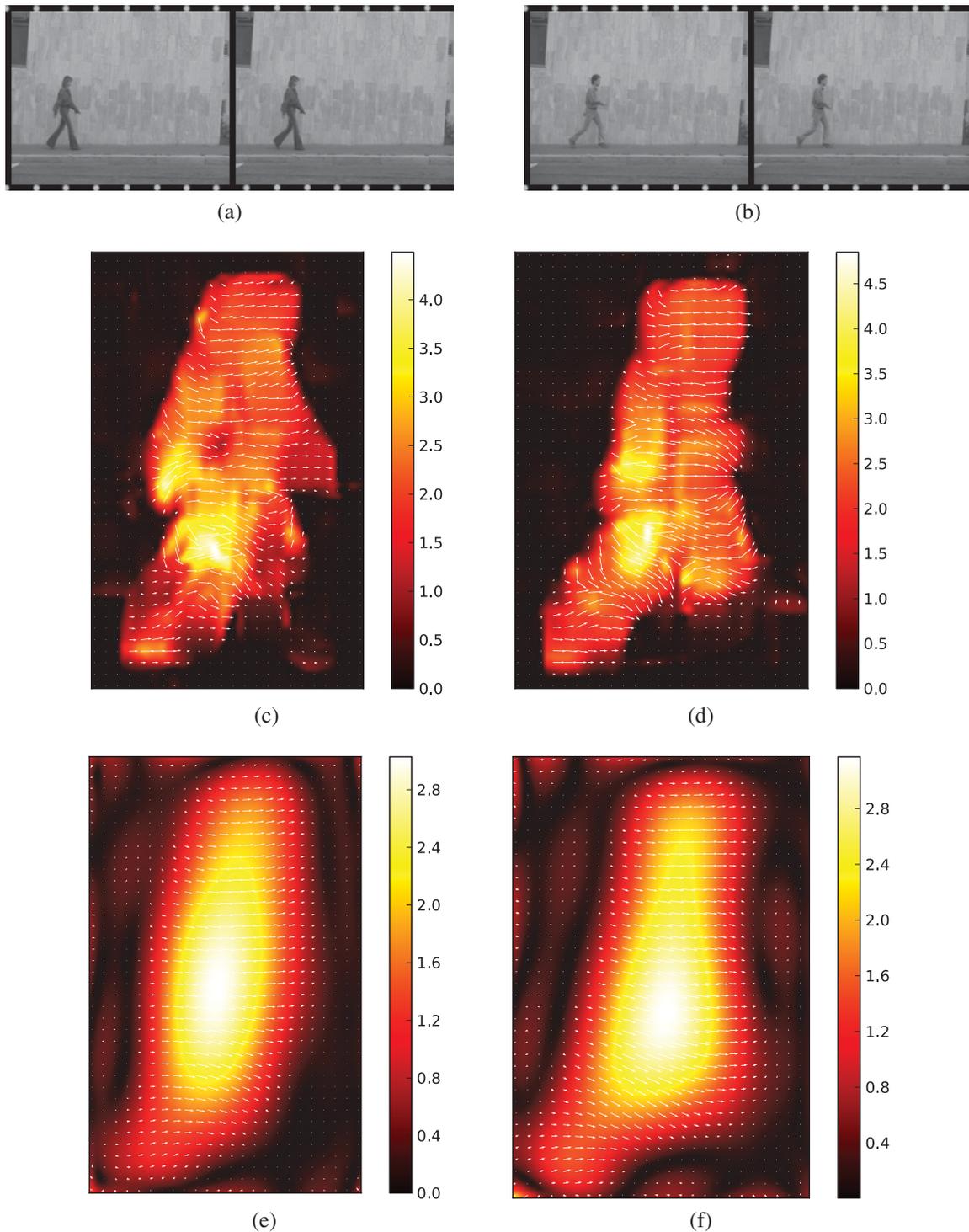


FIGURE 8.1 – Walk : Mesure de similarité entre le champ (c) et le champ (d) = 0.96 (avec une base de degré 7) ; (a) Images 18 et 19 de la séquence Daria_walk issue de la base de WEIZMANN ; (b) Images 16 et 17 de la séquence Denis_walk issue de la base de WEIZMANN ; (c) et (d) champ de déplacements calculé par flot optique (LUCAS et KANADE avec un voisinage de 5×5 pixels) sur la zone d'intérêt respectivement de (a) et (b) ; (e) et (f) représentation de la modélisation des champs (c) et (d) par projection sur une base de polynômes de degré 7.

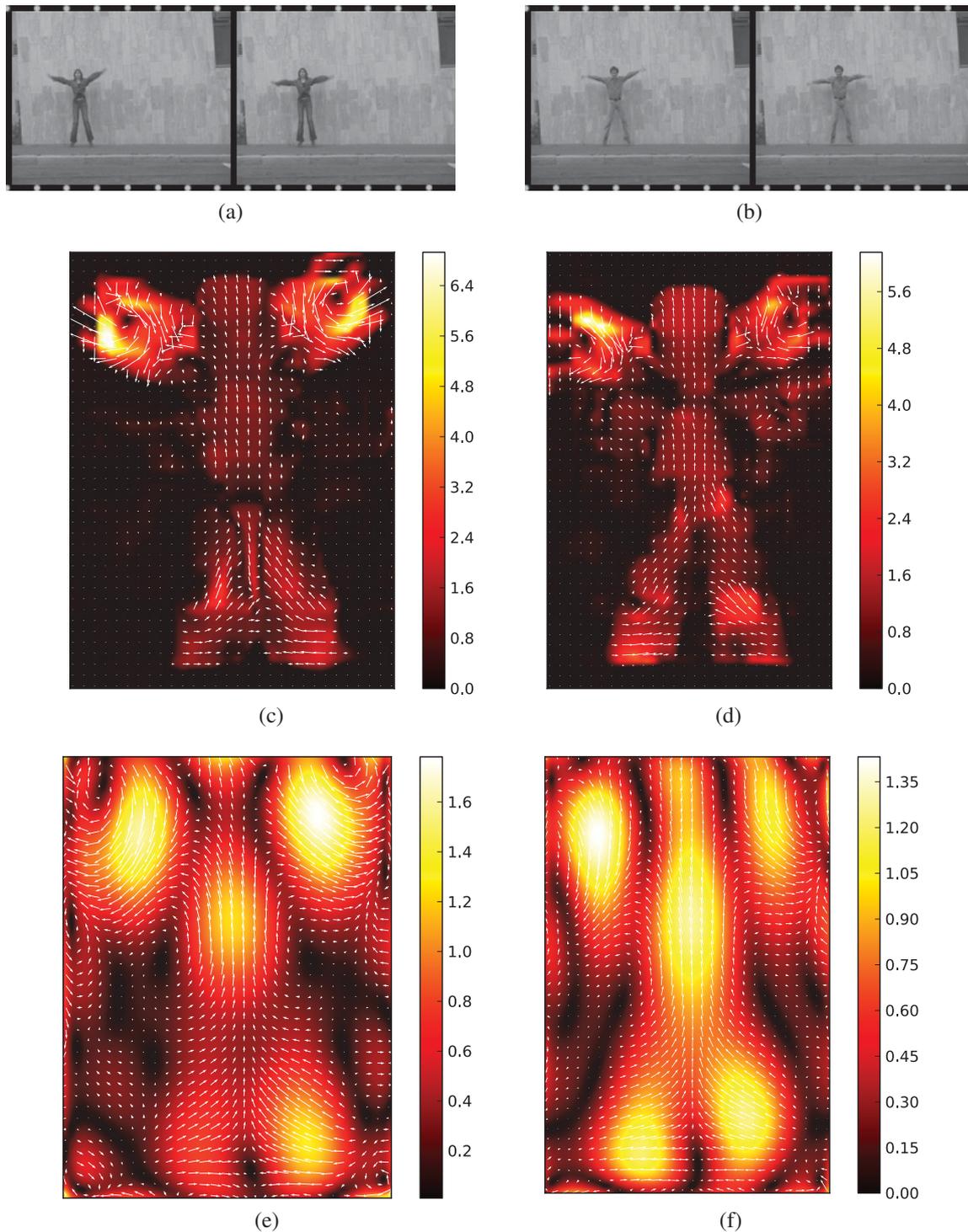


FIGURE 8.2 – Jack : Mesure de similarité entre le champ (c) et le champ (d) = 0.80 (avec une base de degré 7) ; (a) Images 51 et 52 de la séquence Daria_jack issue de la base de WEIZMANN ; (b) Images 39 et 40 de la séquence Denis_jack issue de la base de WEIZMANN ; (c) et (d) champ de déplacements calculé par flot optique (LUCAS et KANADE avec un voisinage de 5×5 pixels) sur la zone d'intérêt respectivement de (a) et (b) ; (e) et (f) représentation de la modélisation des champs (c) et (d) par projection sur une base de polynômes de degré 7.

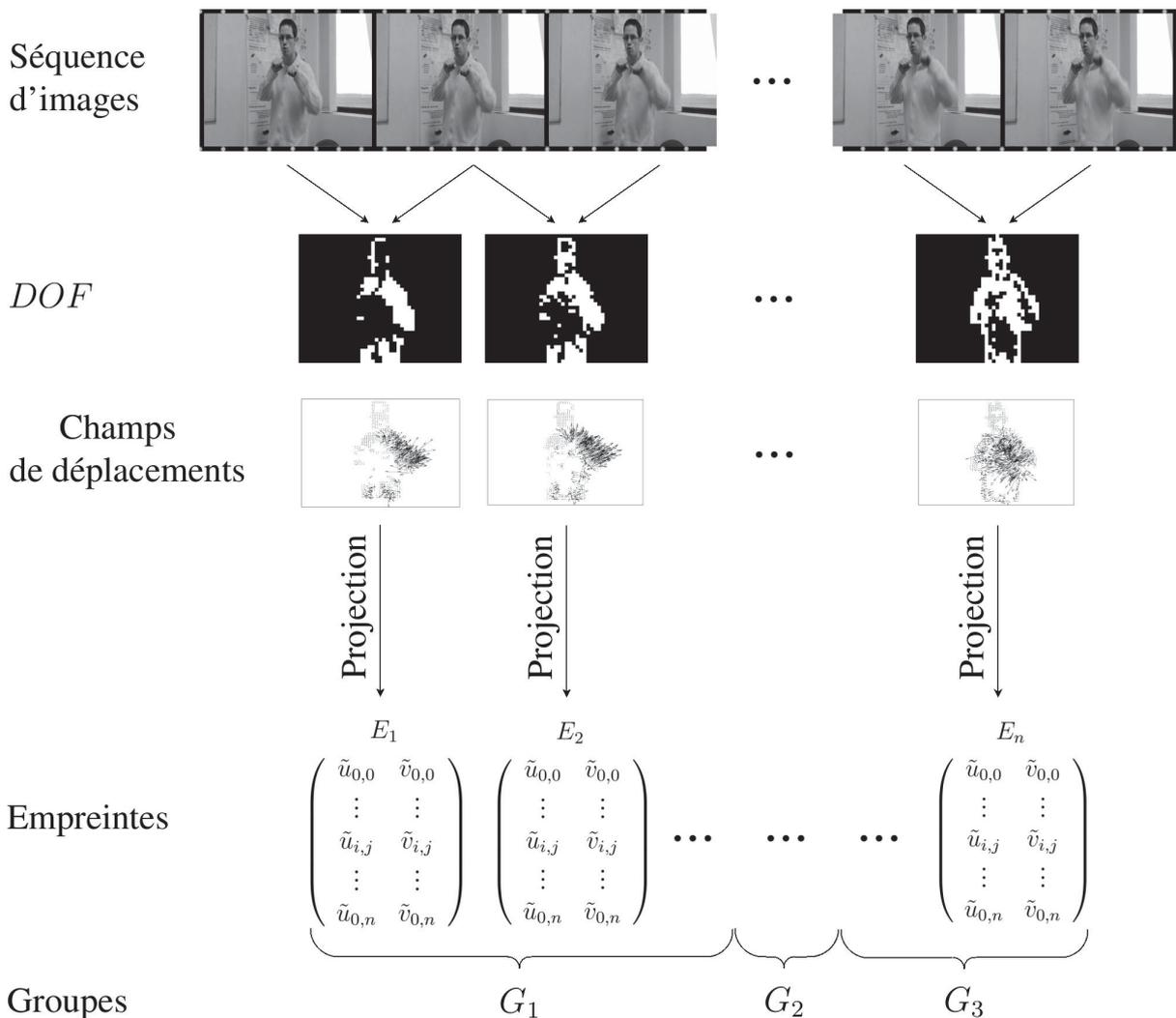


FIGURE 8.3 – Illustration de l'étape d'extraction des empreintes à partir de la séquence d'apprentissage d'un mouvement type

temps réel grâce à des techniques de programmation mettant à profit les processeurs des cartes graphiques. Cela est important puisque nous verrons en perspective qu'une implémentation en temps réel de la méthode est envisageable.

L'étape d'extraction de la zone d'intérêt a plusieurs objectifs : calculer un masque pour ne conserver que les mouvements les plus importants ou isoler les silhouettes. Pour cela, il est possible d'utiliser la différence entre image (DOF) définie en équation (8.1).

Ensuite, pour chaque champ, l'empreinte du mouvement est calculée par projection du champ de déplacements sur une base de polynômes bivariés. Dans le cas de la reconnaissance de mouvements de boxe, un masque est appliqué au champ de déplacements. Dans ce cas le champ est épars, il peut donc être intéressant d'utiliser une base de polynômes éparse (cf. chapitre 5). Cela permet de ne prendre en compte que l'information du mouvement dans les zones où ce dernier est défini. Cependant, dans le cas de la classification des différentes vidéos de la base de WEIZMANN, nous utilisons des bases de polynômes denses. En effet, dans ce cas là, l'étape d'extraction de la zone d'intérêt permet de réduire le domaine de définition du champ

à une zone autour de la silhouette. En utilisant des bases de polynômes denses, nous modélisons la zone en mouvement de la silhouette et le mouvement lui-même dans un seul modèle.

À partir d'un mouvement de référence, l'objectif de l'application est de retrouver ce mouvement dans une vidéo. L'algorithme utilisé pour reconnaître un mouvement de référence dans une vidéo est défini comme suit :

Pour chaque couple d'images de la séquence vidéo à analyser :

1. *Estimer le champ de déplacements ;*
2. *Extraire la zone d'intérêt ;*
3. *Calculer l'empreinte du champ de déplacements ;*
4. *Calculer la similarité entre cette empreinte et toutes les empreintes du mouvement de référence ;*
5. *Rechercher l'indice du groupe qui obtient la plus grande similarité puis :*
 - *si ce meilleur score est inférieur à un seuil, à définir empiriquement, alors on pousse dans la pile S l'indice 0, car cela signifie qu'il y a trop d'incertitude sur la mesure de similarité ;*
 - *si l'indice du meilleur score est équivalent à l'indice se trouvant déjà en haut de la pile alors ne rien faire ;*
 - *sinon pousser cet indice dans la pile S .*
6. *Si la suite 1, 2, ..., n (n le nombre de groupes) se trouve dans le haut de la pile alors le mouvement de référence est reconnu ;*

La figure FIG 8.4 illustre cet algorithme. La simplicité de cet algorithme nous permet de mesurer directement l'impact de notre descripteur de mouvement. En effet, les algorithmes de reconnaissance les plus efficaces de la littérature ont recours à des algorithmes beaucoup plus élaborés tels que les modèles de Markov cachés (HMM) ou les séparateurs à vaste marge. Cependant, il est tout à fait possible d'utiliser ce descripteur de mouvement au sein de la majorité des algorithmes de la littérature.

8.4.2 Résultats sur une application de type commande de jeu vidéo

Notre application s'inspire de la reconnaissance de mouvement tel que cela peut être utilisé dans le domaine des interfaces utilisateur avancées comme par exemple dans les jeux vidéos.

Nous cherchons à reconnaître les mouvements effectués par un utilisateur devant une caméra. Pour chaque mouvement, nous avons une seule référence de ce mouvement. L'utilisateur peut effectuer les 5 mouvements de boxe de référence suivants :

- esquivier un coup en bougeant à droite (R) ;
- esquivier un coup en bougeant à gauche (L) ;
- esquivier en se baissant (D) ;
- frapper avec un direct du bras arrière (P) ;
- frapper avec un direct du bras avant (J) ;

Ces mouvements sont présentés sur la figure FIG 8.5.

L'utilisateur est filmé pendant qu'il effectue un exemple de chaque mouvement. Cela permet d'obtenir une référence de chaque mouvement. Notons que dans cette application, nous n'avons qu'une référence de chaque mouvement, mais nous avons bien une référence par utilisateur. Chaque mouvement de référence est décomposé en 4 groupes d'empreintes.

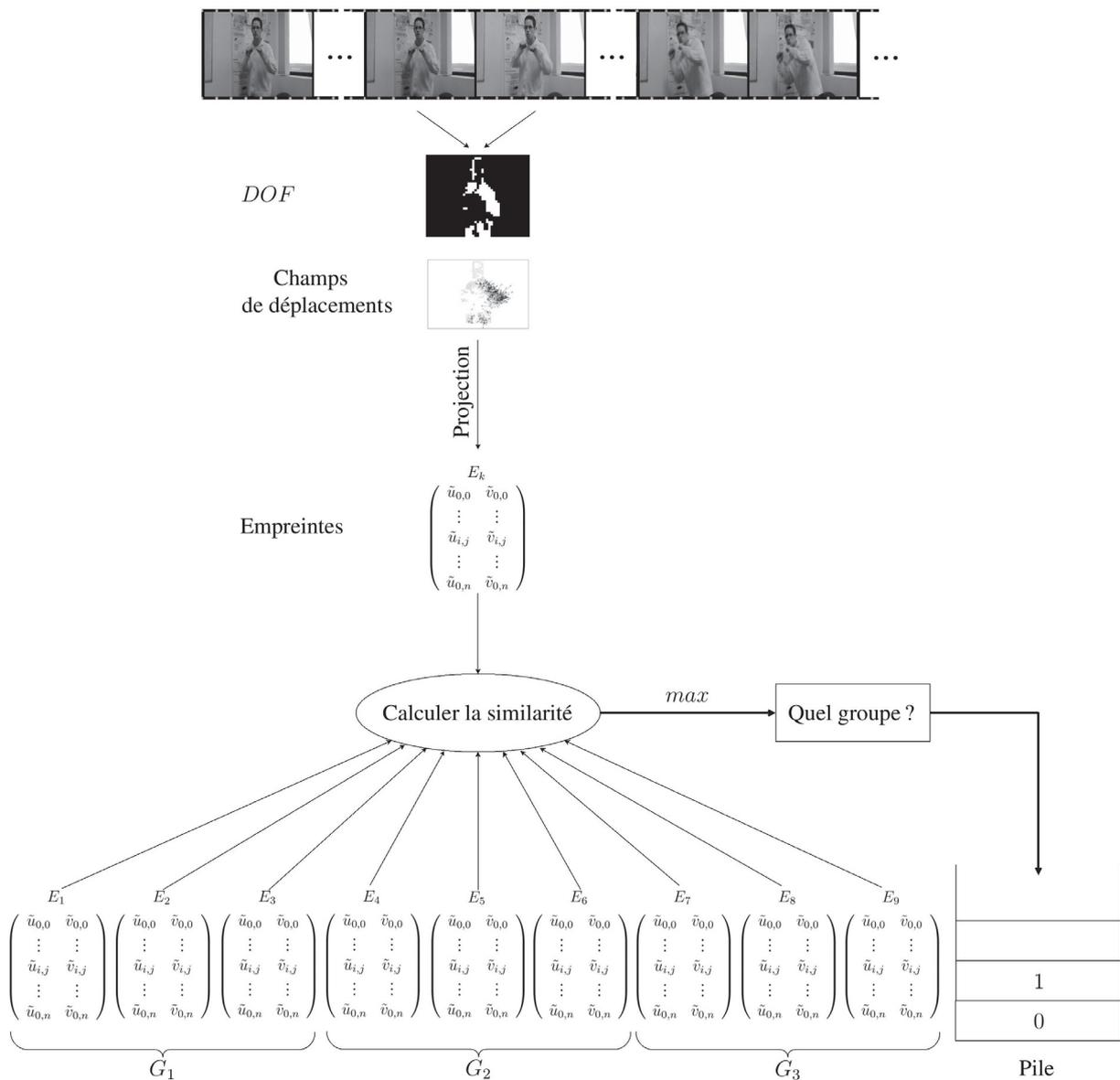


FIGURE 8.4 – Illustration de l’algorithme de détection et de reconnaissance de mouvement humain

Nous obtenons donc 5 vidéos de mouvements de référence. Afin de localiser précisément le début et la fin du mouvement de référence, cette étape a été effectuée manuellement. Ces vidéos sont alors transformées en séquences de champs de déplacements. Nous utilisons l’algorithme de flot optique de LUCAS et KANADE avec une fenêtre de voisinage de 5×5 pixels sur des images en niveaux de gris. Pour l’extraction de la zone d’intérêt par *DOF*, Γ est fixé à 5 niveaux de gris. Ensuite ces champs sont transformés en séquences d’empreintes par projection sur une base de polynômes de degré 3. Finalement, nous tentons de retrouver les mouvements de référence dans une séquence vidéo. Les images et les champs sont de 180×144 pixels. Dans l’algorithme de reconnaissance de mouvement, le seuil concernant le meilleur score de similarité avant de pousser un indice dans la pile est fixé à 0.8.

Sur la figure FIG 8.6, nous présentons quelques résultats de reconnaissance de mouvement pour trois utilisateurs. Chacune des séquences a une durée de 60 secondes. Pour chaque utilis-



(a) Esquiver un coup en bougeant à droite (*R*) (Premier utilisateur sur la figure FIG 8.6a)



(b) Esquiver un coup en bougeant à gauche (*L*) (Second utilisateur sur la figure FIG 8.6b)



(c) Esquiver en se baissant (*D*) (Premier utilisateur sur la figure FIG 8.6a)



(d) Frapper avec un direct du bras arrière (*P*) (Troisième utilisateur sur la figure FIG 8.6c)



(e) Frapper avec un direct du bras avant (Premier utilisateur sur la figure FIG 8.6a)

FIGURE 8.5 – Les 5 mouvements de référence utilisés pour nos tests

teur, nous présentons sur une ligne la détection des mouvements de la séquence réalisée par un expert, et en dessous sur une autre ligne les résultats obtenus par notre algorithme.

Pour le premier utilisateur, la séquence est composée de 37 mouvements et 32 sont correctement détectés et reconnus. Il n'y a aucun faux positifs et la localisation temporelle des mouvements est très proche de celle de l'expert.

Pour le second utilisateur, il y a 34 mouvements d'après l'expert, 31 sont correctement reconnus par notre algorithme. La localisation temporelle des mouvements reste bonne mais il y a 7 faux positifs dans la détection.

Pour le troisième utilisateur, la séquence compte 40 mouvements, notre algorithme en détecte correctement 35 et il y a 2 faux positifs.

Quelque soit l'utilisateur, les mouvements ne sont pas toujours effectués avec la même vitesse d'exécution. La figure FIG 8.6 permet de vérifier que la majorité des mouvements est détectée par notre algorithme, il semblerait donc que notre algorithme soit robuste à la vitesse d'exécution des mouvements.

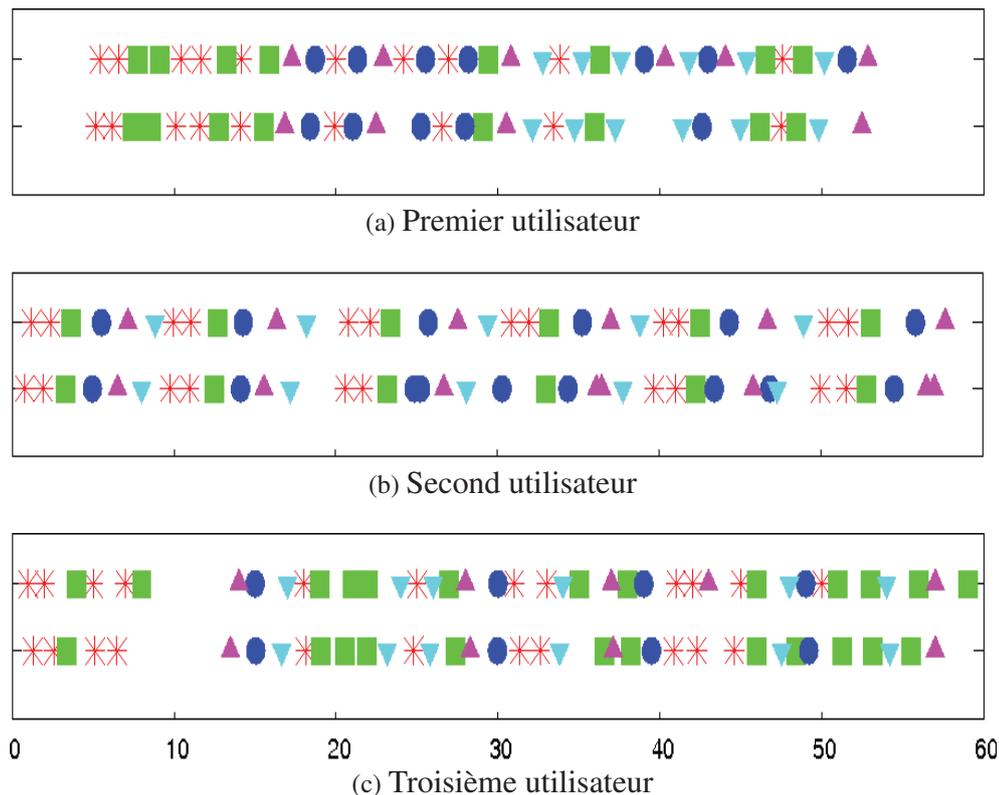


FIGURE 8.6 – Reconnaissance de mouvement de 5 actions avec une base de polynômes de degré 3. Pour chaque utilisateur, nous présentons sur la ligne du haut la séquence attendue et sur la ligne du bas les résultats obtenus avec notre algorithme de détection. * frapper avec un direct du bras avant ; □ frapper avec un direct du bras arrière ; ○ esquiver un coup en bougeant à gauche ; △ esquiver un coup en bougeant à droite ; ▽ se baisser.

Si les champs de déplacements sont calculés à l'avance, la reconnaissance de mouvement peut être calculée en 45 secondes pour une séquence de 60 secondes avec une base éparse de degré 3 telle que celle utilisée dans cette application. Nous avons constaté que dans la littérature, il existe des implantations informatiques sur carte graphique (GPU) de l'algorithme de flot optique de LUCAS et KANADE [Le Besnerais 05] permettant de calculer tous les champs de déplacements de nos vidéos en moins de 10 secondes. Cela nous permet d'envisager une application en temps réel de notre méthode.

8.5 Application à la classification d'une base de vidéos

Dans cette section, nous proposons une méthode pour classifier les actions humaines. Pour cela, nous classifions les vidéos de la base de WEIZMANN [Blank 05]. Dans un premier temps, nous présentons cette base. Ensuite, nous détaillons l'algorithme de classification et enfin nous exposons nos résultats.

8.5.1 Présentation de la base de WEIZMANN

Dans ce chapitre, nous utilisons la base de WEIZMANN issue des travaux de GORELICK et al. [Gorelick 07], composée de 10 mouvements types. En effet, initialement [Blank 05], la base était composée seulement de 9 mouvements. Nous avons choisi cette base car elle est relativement simple à étudier et que de nombreux autres auteurs l'ont utilisée pour valider leurs résultats. Cette base est simple puisque le fond de la vidéo est relativement constant sur toutes les vidéos. De plus, il n'y a pas de mouvement de caméra, ni de changement d'illumination pendant les vidéos. La base est constituée des 10 mouvements suivants :

- **Walk** : marcher ;
- **run** : courir ;
- **jump** : sauter à pieds joints ;
- **side** : se déplacer en pas chassés ;
- **bend** : se baisser ;
- **wave1** : agiter une main au dessus de la tête ;
- **wave2** : agiter les deux mains au dessus de la tête ;
- **pjump** : sauter sur place ;
- **skip** : sauter à cloche pied (cloche pied) (ce mouvement n'est pas dans la version [Blank 05]) ;
- **jack** : sauter sur place en écartant les pieds et les bras puis en les serrant de manière alternative (pjump+wave2).

Chacun de ces mouvements est effectué par 9 personnes. Un extrait de chaque mouvement est présenté sur les figures FIG 8.7 et FIG 8.8. Pour présenter l'ensemble des acteurs de la base de vidéo, nous présentons chaque mouvement avec un acteur différent.

Les vidéos ont une durée différente les unes des autres. Dans une vidéo donnée, l'acteur effectue plusieurs fois le même mouvement type, à l'exception du mouvement **bend** qui n'est effectué qu'une fois par vidéo. Sur ces vidéos, nous avons extrait manuellement une seule exécution du mouvement de référence, par exemple un pas pour une vidéo de la classe **walk**, ou un saut pour une vidéo de la classe **jump**. Le tableau TAB 8.2 précise, pour chaque classe d'actions et pour un acteur donné, le début et la fin de l'action que nous pouvons sélectionner pour notre ensemble d'apprentissage. Afin de classifier les vidéos, nous utilisons soit 1, soit 2, soit 3 références (1, 2 ou 3 colonne du tableau TAB 8.2) de chaque mouvement pour composer notre ensemble d'apprentissage.

8.5.2 Présentation de l'algorithme global de classification

Avant de classifier une vidéo, il est nécessaire de modéliser les mouvements de référence (cf. TAB 8.2) qui constituent notre ensemble d'apprentissage. L'objectif de cette étape est de représenter un extrait vidéo par un tableau d'empreintes. Nous proposons donc l'algorithme suivant :

Pour chaque couple d'images, de la vidéo d'un mouvement de référence, l'algorithme doit :

1. *calculer la zone d'intérêt ;*
2. *dans cette zone d'intérêt, estimer le mouvement ;*
3. *calculer l'empreinte polynomiale du champ de déplacements ;*

Nous illustrons cet algorithme sur la figure FIG 8.9 et nous détaillons chaque étape ci-après.

La zone d'intérêt est calculée avec la méthode utilisée dans [Zhang 08]. Cette étape est calculée pour chaque couple d'image en prenant en compte les 4 images précédentes ainsi que



(a) Images 2 à 14 par pas de 2 de la séquence walk



(b) Images 10 à 22 par pas de 2 de la séquence run



(c) Images 1 à 15 par pas de 2 de la séquence jump



(d) Images 1 à 13 par pas de 2 de la séquence side



(e) Images 20 à 38 par pas de 3 de la séquence bend

FIGURE 8.7 – Cinq premiers mouvements de la base de WEIZMANN

Classes \ Acteurs	Acteurs								
	1	2	3	4	5	6	7	8	9
Walk	12-35	10-33	6-31	1-24	6-32	8-32	6-31	1-23	7-30
Run	5-34	14-32	7-39	7-36	11-42	1-28	5-33	1-30	5-34
Jump	5-26	5-26	13-37	12-35	5-28	11-33	5-24	14-38	9-31
Side	1-25	4-27	1-25	13-38	6-32	6-30	13-38	12-36	13-38
Bend	1-68	1-67	12-59	4-55	17-68	5-60	4-62	1-56	5-54
Wave1	20-52	12-50	12-50	9-45	10-48	8-45	10-43	11-51	12-49
Wave2	4-36	22-51	10-43	10-48	18-53	12-50	14-48	13-50	25-59
Pjump	13-35	10-34	18-41	6-33	8-29	11-33	19-44	11-38	13-39
Skip	7-27	7-27	11-33	11-35	9-30	3-25	9-32	6-28	11-33
Jack	28-62	26-50	25-51	22-54	10-42	13-50	12-47	14-47	12-44

TABLEAU 8.2 – Indice de début et de fin des actions pour chaque classe d'action et pour chaque acteur ; ces indices sont sélectionnés manuellement.



(a) Images 1 à 25 par pas de 4 de la séquence wave1



(b) Images 6 à 24 par pas de 3 de la séquence wave2



(c) Images 6 à 18 par pas de 2 de la séquence pjump



(d) Images 5 à 17 par pas de 2 de la séquence skip



(e) Images 2 à 20 par pas de 3 de la séquence jack

FIGURE 8.8 – Cinq derniers mouvements de la base de WEIZMANN

les 5 images suivantes. Une image est créée à partir de l'écart-type temporel des pixels de ces 11 images, puis binarisée en fonction d'un seuil. Nous fixons ce dernier à 25 niveaux de gris pour des images dont la dynamique est comprise dans $[0, 255]$. Tous les pixels dont la valeur est supérieure à ce seuil appartiennent à la zone d'intérêt (zone où le mouvement est non nul). Cette étape est utilisée pour isoler les silhouettes. En effectuant cette étape avant l'estimation du mouvement, il est ainsi possible de gagner du temps de calcul lors de l'estimation du mouvement. Nous définissons alors une boîte autour de la silhouette et le champ de déplacements n'est calculé que dans cette zone. Cette méthode permet aussi d'isoler les parties du corps en mouvement sans obtenir de zones parasites dues aux changements d'illumination par exemple.

L'empreinte polynomiale est calculée en projetant le champ de déplacements sur une base polynomiale dense de degré donné. Le modèle d'un mouvement est donc composé d'un vecteur (tableau) d'empreintes de longueur variable (puisque tous les mouvements ne sont pas de même durée). Dans le cas où nous utilisons une seule référence par mouvement type, l'ensemble d'apprentissage est une liste de 10 vecteurs puisqu'il y a 10 mouvements types.

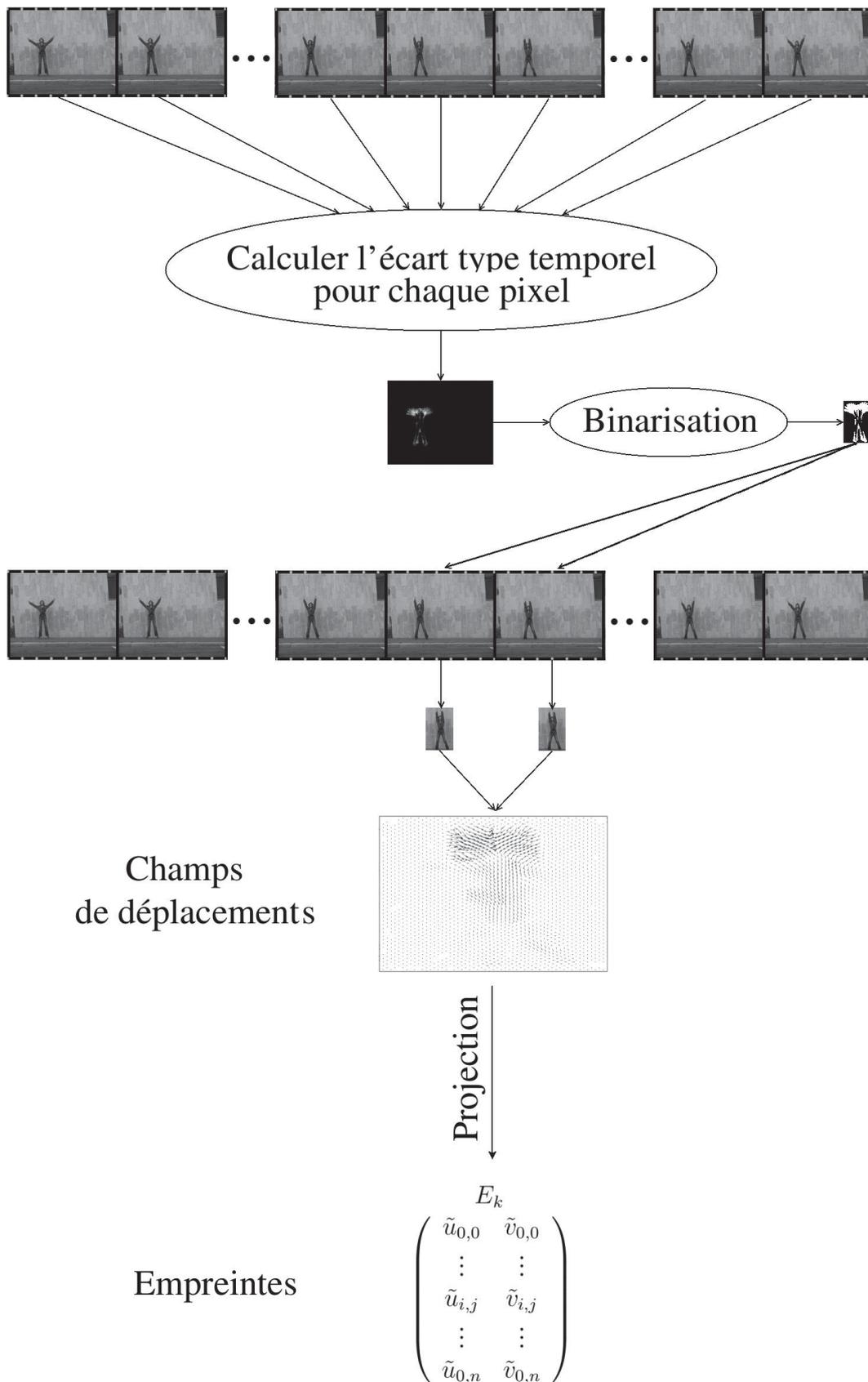


FIGURE 8.9 – Illustration des étapes d'extraction d'une empreinte dans la séquence d'images d'un mouvement de référence

À partir de maintenant une vidéo est considérée comme une suite d'empreintes extraites avec cet algorithme. Dans la suite, nous utilisons donc l'expression "Extraire l'empreinte" pour faire référence aux étapes 1,2 et 3 de l'algorithme précédent.

Une vidéo de la base est ensuite soumise à notre algorithme qui fonctionne comme suit. Nous cherchons à classifier k classes de mouvements. Notre algorithme nécessite alors k variables initialisées à 0.

Pour chaque couple d'images de la vidéo, l'empreinte est extraite puis pour chacun des vecteurs de l'ensemble d'apprentissage, nous devons :

- trouver l'empreinte qui maximise la similarité avec E_{test} ;
- ajouter à la variable qui correspond à ce vecteur la mesure de similarité maximale qui a été trouvée ; dans le cas où nous avons plusieurs références pour un même mouvement type, c'est le meilleur score de similarité entre ces références qui est retenu.

À la fin de la vidéo, la variable qui est la plus élevée permettra de trouver le mouvement type le plus ressemblant à la vidéo analysée. Cet algorithme est illustré sur la figure FIG 8.10.

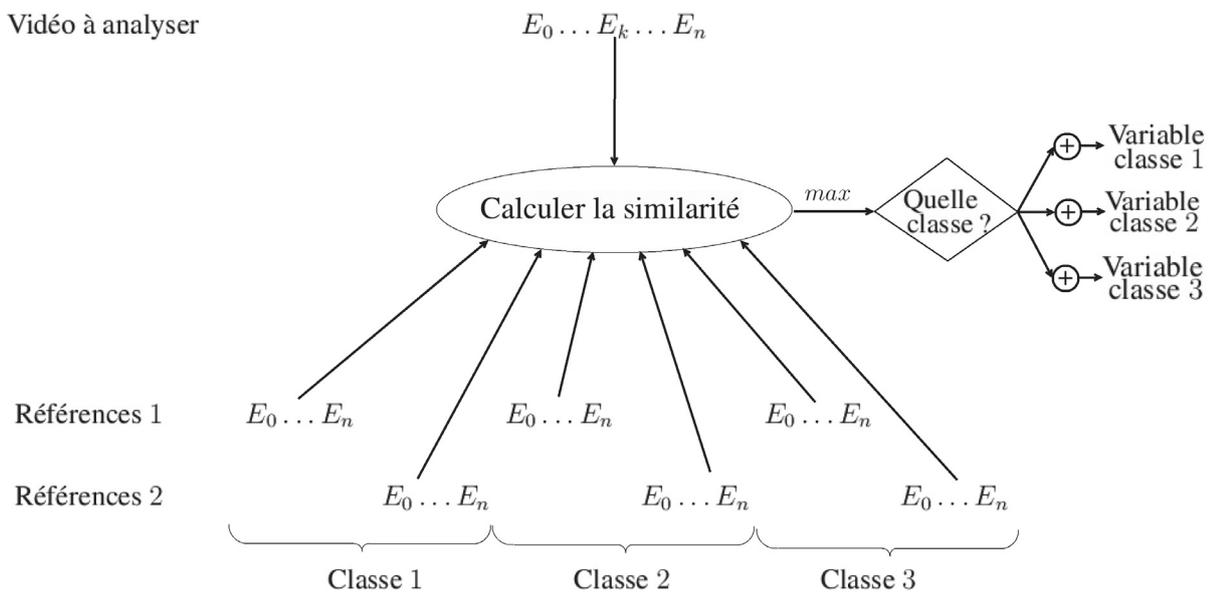


FIGURE 8.10 – Illustration de notre algorithme de classification de vidéo ; pour cet exemple, la classification est effectuée parmi 3 classes avec 2 mouvements de références par classe

8.5.3 Présentation de l'algorithme hiérarchique de classification

Nous avons vu au chapitre 4, différentes méthodes de subdivisions permettant d'obtenir une approximation de qualité tout en utilisant de bases de degré faible. Nous utilisons ici ce principe pour représenter une empreinte. Pour cela, nous utilisons le quadtree sur un seul niveau de décomposition. L'ensemble de l'algorithme reste identique, à l'exception de l'étape de projection. Le champ de déplacements est toujours projeté sur la base de polynômes afin d'obtenir une empreinte, mais celle-ci ne constitue pas à elle seule le modèle d'un champ donné. Nous concaténons cette empreinte avec les quatre empreintes obtenues au premier niveau de décomposition par quadtree. La figure FIG 8.11 illustre cette nouvelle étape. Dans la section suivante, nous présentons les résultats obtenus avec l'algorithme global puis ceux obtenus avec ce dernier.

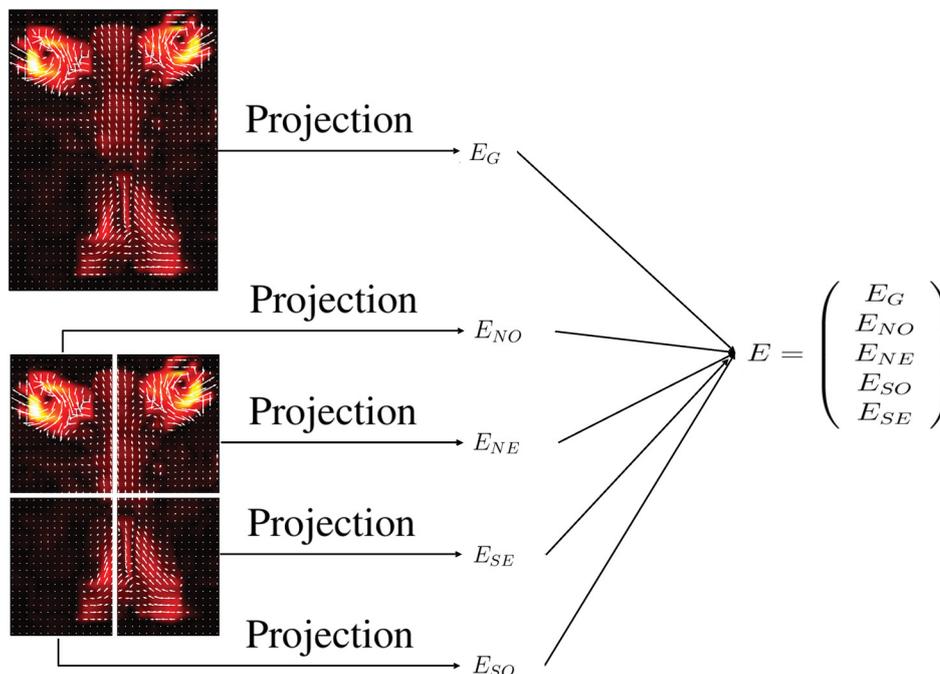


FIGURE 8.11 – Illustration de la création d'une empreinte avec l'algorithme hiérarchique

8.5.4 Résultats de classification sur la base de WEIZMANN

Nous présentons les résultats sous la forme de matrices de confusions. La première colonne de la matrice indique, sur chaque ligne, une classe donnée. Ensuite, chaque colonne permet d'identifier comment les représentants de cette classe sont classifiés. Ici, nous présentons les résultats en pourcentage. Dans le cas d'une classification parfaite, toutes les cases de la diagonale sont à 100% et les autres sont nulles.

Cette section "résultats" est organisée comme suit. Dans un premier temps, nous présentons les résultats de l'algorithme global et de l'algorithme hiérarchique dans le cas où nous utilisons une puis deux références par mouvement type. Dans chaque cas, nous réalisons plusieurs classifications de la base de WEIZMANN en faisant aussi varier le degré de la base de polynômes utilisée. L'objectif est de démontrer que notre algorithme obtient des résultats acceptables même avec un nombre réduit de références. Les matrices de confusions présenteront donc la moyenne des résultats obtenus pour toutes les combinaisons de références (d'acteurs) utilisées pour la création de l'ensemble d'apprentissage. Par exemple, le tableau TAB 8.3 représente la matrice de confusion moyenne obtenue pour toutes les combinaisons de l'ensemble d'apprentissage avec une seule référence par mouvement type, c'est-à-dire la moyenne de 9 matrices de confusion. Le tableau TAB 8.6 représente la matrice de confusion moyenne obtenue pour toutes les combinaisons de l'ensemble d'apprentissage avec deux références par mouvement type (acteur 1 et acteur 2, acteur 1 et acteur 3, etc) soit la moyenne de 45 matrices de confusion.

Ensuite, dans le but de nous comparer aux travaux de la littérature, nous présentons nos résultats en excluant de l'ensemble d'apprentissage uniquement les vidéos ayant le même acteur que celle que l'on cherche à classifier. Cette stratégie, aussi utilisée dans [Tran 08], est appelée en anglais "Leave One Actor Out" (LOAO). D'autres auteurs utilisent toutes les vidéos de la base comme ensemble d'apprentissage à l'exception de la vidéo analysée, cette stratégie est appelée en anglais "Leave One Out" (LOO).

8.5.5 Résultats avec l'algorithme global

8.5.5.1 Une seule référence par mouvement type

Le tableau TAB 8.3 présente la matrice de confusion dans le cas d'une base de degré 1. Le taux de reconnaissance global de la base est de 73.84% et 5 classes obtiennent un taux de reconnaissance supérieur à de 80%. Il est à noter que dans ce cas, chaque empreinte résume un champ de déplacements avec 6 coefficients. Il semble donc normal de ne pas réussir à obtenir de meilleurs taux de reconnaissance pour l'ensemble des classes avec un modèle polynomial aussi simple.

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	42.22	33.33	10	0	0	0	0	0	14.44	0
<u>Run</u>	5.55	85.55	0	0	0	0	0	0	8.89	0
<u>Jump</u>	0	4.94	81.48	0	0	0	0	0	13.58	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	65.43	34.57	0	0	0	0
<u>Wave1</u>	0	0	0	0	8.64	91.36	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	98.76	0	0	1.23
<u>Pjump</u>	0	0	0	0	0	0	12.34	28.39	0	59.26
<u>Skip</u>	1.11	27.78	22.22	0	0	0	0	0	48.89	0
<u>Jack</u>	0	0	0	0	0	0	1.23	2.47	0	96.29

TABLEAU 8.3 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 1 et une seule référence par mouvement type (6 coefficients, 73.84%).

Nous notons que les confusions ne semblent pas complètement aléatoires. Par exemple, les mouvements walk, run et skip sont confondus les uns avec les autres. Ces mouvements sont relativement proches pour l'homme et ces résultats sont donc compréhensibles étant donné la simplicité du modèle. C'est aussi le cas pour la confusion entre le mouvement jump et le mouvement skip, sauter à cloche pied ou sauter à pieds joints sont deux mouvements relativement proches. Pour le cas de la confusion entre les mouvements wave1 et bend, la confusion semble moins compréhensible du point de vue de la perception humaine de ces mouvements. Ici la raison est principalement liée à l'algorithme utilisé pour extraire la zone d'intérêt. En effet, dans le cas du mouvement wave1, seul le bras est en mouvement. C'est donc la seule partie du corps à l'intérieur de la boîte englobante avant l'estimation du mouvement. Le mouvement du bras de l'action wave1 est très proche du mouvement global du corps pour l'action bend. Les résultats de notre algorithme d'extraction de la zone d'intérêt, spécifiquement dans le cas de ces deux actions est présenté sur la figure FIG 8.12. Nous y associons les exemples de champs de déplacements ainsi que leur modélisation avec une base de degré 1. Cela permet de comprendre cette confusion. La confusion entre le mouvement wave2 et le mouvement jack est liée au fait que le mouvement jack est composé du mouvement wave2 est d'un mouvement proche du mouvement pjump. Cette explication est valable aussi pour la confusion entre le mouvement pjump et le mouvement jack. En augmentant le degré de la base ce problème devrait disparaître.

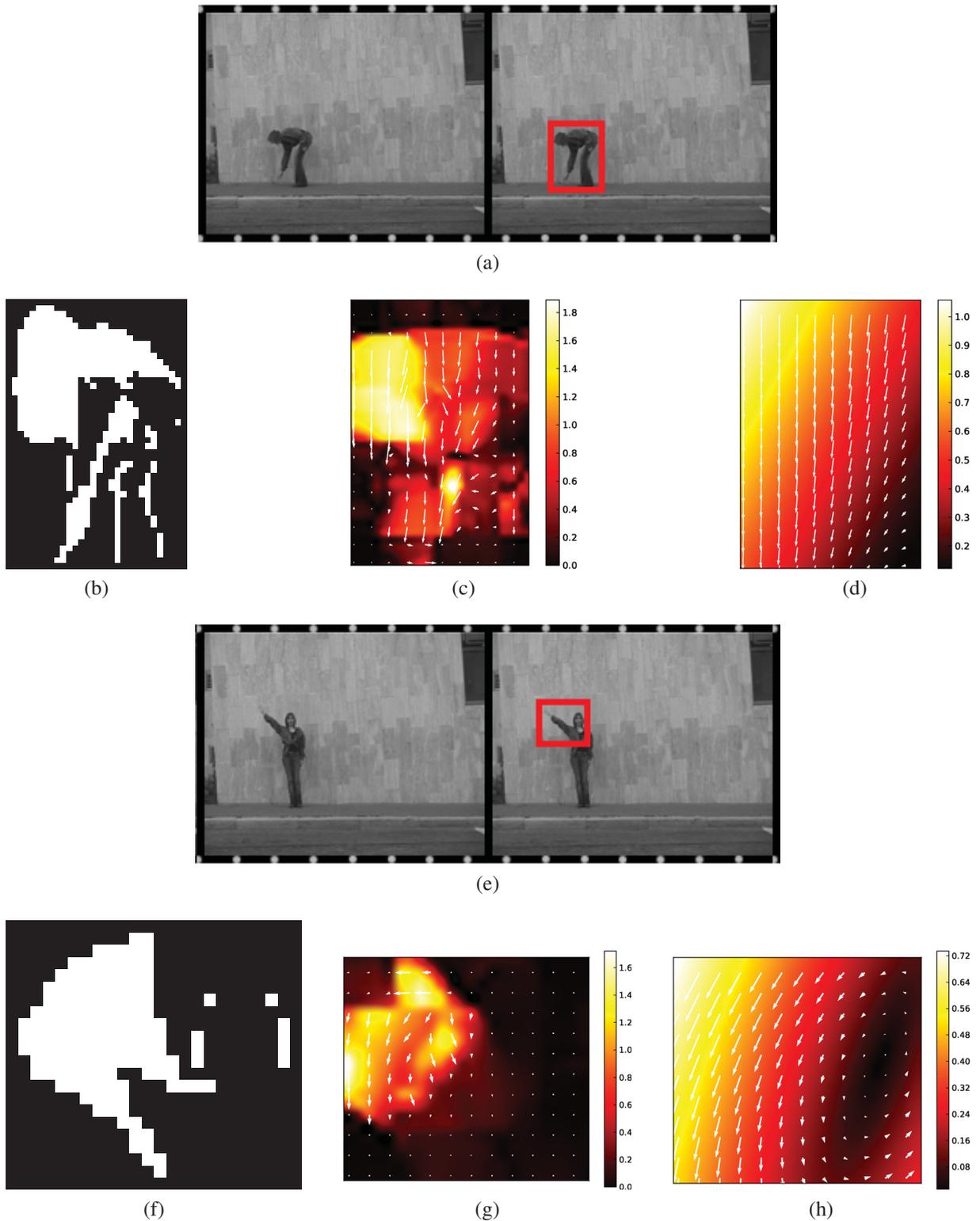


FIGURE 8.12 – Illustration de l'impact de l'extraction de la zone d'intérêt sur la classification des actions bend et wave1 ; (a) Images 28 et 29 de la séquence *daria bend* - le rectangle rouge représente la zone d'intérêt ; (b) Zone d'intérêt extraite pour la série de 11 images centrées sur l'image 28 ; (c) Champ de déplacements estimé entre les images 28 et 29 (a) ; (d) Modélisation de (c) par projection sur une base de polynômes de degré 1 ; (e) Images 33 et 34 de la séquence *daria wave* - le rectangle rouge représente la zone d'intérêt ; (f) Zone d'intérêt extraite pour la série de 11 images centrées sur l'image 33 ; (g) Champ de déplacements estimé entre les images 33 et 34 (e) ; (h) Modélisation de (g) par projection sur une base de polynômes de degré 1 ;

Nous reproduisons la même expérience avec des bases de degré 3, 5, 8 et 12. Les matrices de confusions associées au degré 5 et 12 sont présentées respectivement sur les tableaux TAB 8.4 et TAB 8.5. Nous indiquons, dans la légende de chaque tableau, le nombre de coefficients par empreinte ainsi que le score global de classification. Dans le cas d'une base de degré 12, seule la classe correspondant au mouvement skip obtient un score inférieur à 85%. On peut remarquer que dans la version initiale de la base de WEIZMANN [Blank 05], ce mouvement n'existe pas. Nous aurions donc un taux de reconnaissance supérieur. Nous précisons aussi que pour un ensemble d'apprentissage constitué de l'acteur 1 ou de l'acteur 2, seule la classe skip n'est pas reconnue à 100%.

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	100	0	0	0	0	0	0	0	0	0
<u>Run</u>	0	97.78	0	0	0	0	0	0	2.22	0
<u>Jump</u>	0	1.23	82.72	0	0	0	0	0	16.05	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	83.95	16.05	0	0	0	0
<u>Wave1</u>	0	0	0	0	6.17	93.83	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	97.53	0	0	2.47
<u>Pjump</u>	0	0	0	0	0	0	0	97.53	0	2.47
<u>Skip</u>	0	30	22.22	0	0	0	0	0	47.78	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.4 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 5 et une seule référence par mouvement type (42 coefficients, 90.12%).

La figure FIG 8.13 synthétise les résultats obtenus avec les degré 1 à 12 en représentant l'évolution du taux de reconnaissance en fonction du nombre de coefficients par empreinte. Nous constatons que l'augmentation du degré permet d'améliorer les résultats.

8.5.5.2 Deux références par mouvement type

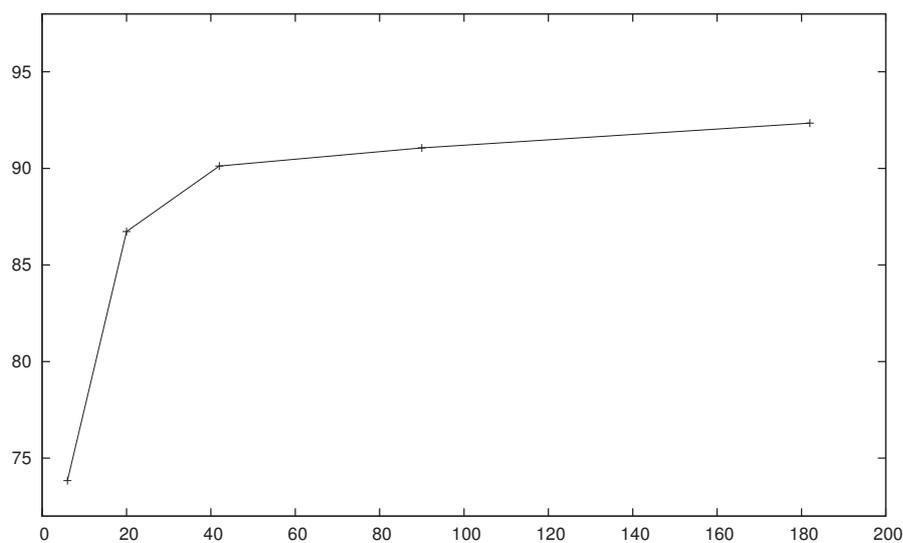
Afin d'améliorer les résultats, nous proposons d'utiliser deux références par mouvement type. Nous constatons sur le tableau TAB 8.6 qu'avec deux références et pour une base de degré 1, il y a 6 classes reconnues à plus 90% contre 4 dans le cas où il n'y a qu'une référence. Les résultats de reconnaissance globaux sont eux aussi supérieurs avec 79.8% avec 2 références contre 73.84% avec une seule référence. Les explications fournies dans la partie précédente pour l'analyse de la matrice de confusion d'une base de degré 1 avec une seule référence sont aussi valables ici.

Le tableau TAB 8.7 présente la matrice de confusion avec une base de degré 8 et 2 références par mouvement type. La base est reconnue à 95.76% et 9 classes sont reconnues avec un taux supérieur à 90%. Notons que lorsque l'ensemble d'apprentissage est composé du couple acteur 1 et acteur 2, la base est reconnue avec un score de 100%. De plus, dans 16 autres combinaisons

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>W</u> <u>a</u> <u>k</u>	100	0	0	0	0	0	0	0	0	0
<u>R</u> <u>u</u> <u>n</u>	0	94.44	0	0	0	0	0	0	5.55	0
<u>J</u> <u>u</u> <u>m</u> <u>p</u>	0	0	87.65	0	0	0	0	0	12.34	0
<u>S</u> <u>i</u> <u>d</u> <u>e</u>	0	0	0	100	0	0	0	0	0	0
<u>B</u> <u>e</u> <u>n</u> <u>d</u>	0	0	0	0	86.42	13.58	0	0	0	0
<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	0	0	0	0	4.94	95.06	0	0	0	0
<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	0	0	0	0	0	1.23	98.76	0	0	0
<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	0	0	0	0	0	0	0	100	0	0
<u>S</u> <u>k</u> <u>i</u> <u>p</u>	3.33	16.67	18.89	0	0	0	0	0	61.11	0
<u>J</u> <u>a</u> <u>c</u> <u>k</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.5 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 12 et une seule référence par mouvement type (182 coefficients, 92.34%).

taux de reconnaissance



Nombre de coefficients par empreinte

FIGURE 8.13 – Évolution du taux de reconnaissance de l'algorithme global en fonction du nombre de coefficients par empreinte avec une seule référence par mouvement type

d'acteurs, seule la classe skip n'est pas reconnue à 100%. Comme dans le cas où nous n'utilisons qu'une seule référence, les classes qui posent problème lors de la classification sont principalement la classe skip et dans une moindre mesure la classe bend.

Nous synthétisons les résultats obtenus avec les degrés 1 à 12 sur la figure FIG 8.14, où nous représentons l'évolution du taux de reconnaissance en fonction du nombre de coefficients

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	49.17	29.72	3.89	0	0	0	0	0	17.22	0
<u>Run</u>	2.22	94.44	0	0	0	0	0	0	3.33	0
<u>Jump</u>	0	1.23	92.90	0	0	0	0	0	5.86	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	76.54	23.46	0	0	0	0
<u>Wave1</u>	0	0	0	0	3.39	96.60	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	99.38	0	0	0.62
<u>Pjump</u>	0	0	0	0	0	0	12.65	31.48	0	55.86
<u>Skip</u>	0.28	26.39	15.83	0	0	0	0	0	57.5	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.6 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 1 et deux références par mouvement type (6 coefficients, 79.8%).

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	99.72	0	0	0	0	0	0	0	0.28	0
<u>Run</u>	0	99.17	0	0	0	0	0	0	0.83	0
<u>Jump</u>	0	0	93.83	0	0	0	0	0	6.17	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	93.21	6.79	0	0	0	0
<u>Wave1</u>	0	0	0	0	0	100	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	100	0	0	0
<u>Pjump</u>	0	0	0	0	0	0	0	100	0	0
<u>Skip</u>	1.67	10	16.67	0	0	0	0	0	71.67	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.7 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme avec une base de degré 8 et deux références par mouvement type (90 coefficients, 95.76%).

par empreinte. Nous constatons, comme dans le cas où nous n'utilisons qu'une référence, que l'augmentation du degré permet d'améliorer les résultats de reconnaissance.

8.5.6 Résultats avec l'algorithme hiérarchique

Nous avons constaté que l'augmentation du degré de la base de polynôme permet d'améliorer les résultats de classification. Cependant, nous avons étudié au chapitre 4 différentes méthodes permettant d'obtenir une approximation de qualité sans augmenter le degré de la base. Pour cela, nous utilisons une modélisation plus locale. L'algorithme hiérarchique que nous util-

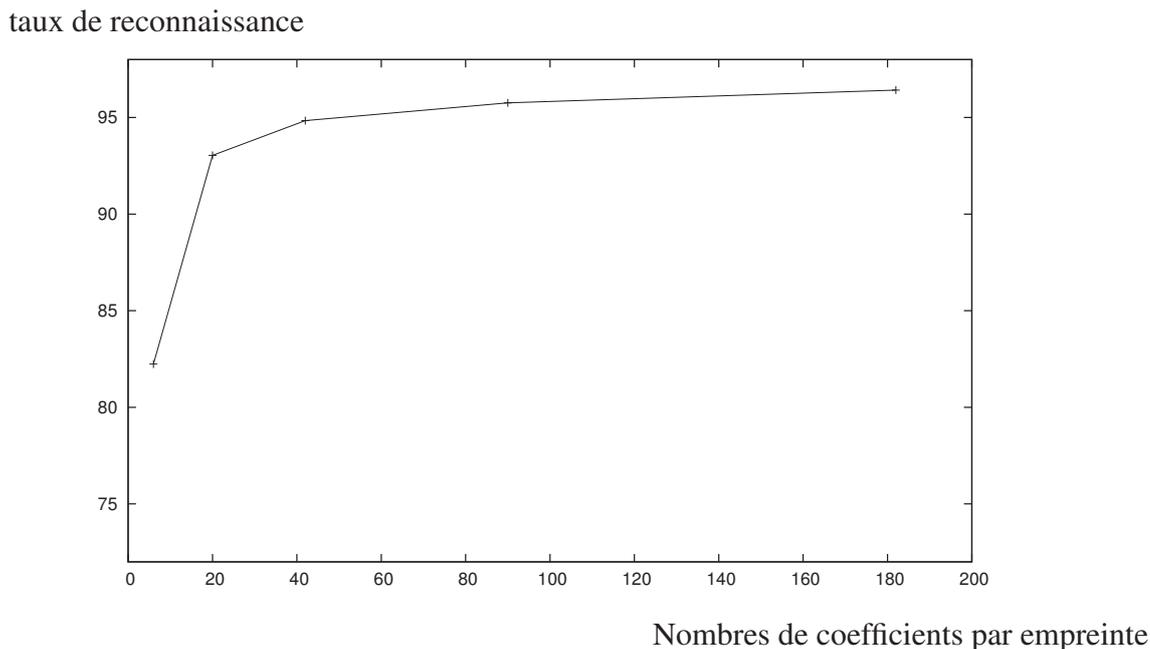


FIGURE 8.14 – Évolution du taux de reconnaissance de l'algorithme global en fonction du nombre de coefficients par empreinte avec deux références par mouvement type

isons ici exploite ce principe. Lors de la modélisation avec une base de degré 1, chaque empreinte est modélisée à partir de $6 \times 5 = 30$ coefficients. Comme dans le cas précédent, nous étudions la classification de la base avec une ou deux références.

8.5.6.1 Une référence par mouvement type

Dans un premier temps, nous présentons, dans le tableau TAB 8.8, la matrice de confusion obtenue avec une base de degré 1. Nous constatons que les résultats sont comparables à ceux obtenus avec l'algorithme global, pour un nombre de coefficients correspondants (cf. figure FIG 8.13). Le taux de classification global est de 89.25% et 7 classes sont reconnues avec un taux de reconnaissance supérieur à 90%. Comme cela était prévisible, les confusions concernent les mêmes classes qu'avec l'algorithme précédent.

Nous présentons sur le tableau TAB 8.9, la matrice de confusion dans le cas d'une base de degré 3, ce qui représente 100 coefficients par empreinte. Le taux de reconnaissance est de 91.96% et comme dans le cas d'une base de degré 1 avec l'algorithme hiérarchique, 7 classes sont reconnues avec un taux de reconnaissance supérieur à 90%. Nous constatons que l'augmentation du degré de la base a un effet moins important sur l'augmentation du taux de reconnaissance que dans le cas de l'algorithme global. Sur la figure FIG 8.15, nous présentons l'évolution du taux de reconnaissance en fonction du nombre de coefficients par empreinte avec l'algorithme hiérarchique et une seule référence par mouvement type. Nous constatons que le taux de reconnaissance cesse d'augmenter au delà du degré 3, ce qui correspond à 100 coefficients par empreintes. Ici le taux de reconnaissance maximum est de 91.96% ce qui légèrement supérieur au taux obtenu avec 90 coefficients pour l'algorithme global avec une seule référence (91.06%) et très légèrement inférieur à celui obtenu avec 182 coefficients pour l'algorithme

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	97.78	1.11	0	0	0	0	0	0	1.11	0
<u>Run</u>	0	97.78	1.11	0	0	0	0	0	1.11	0
<u>Jump</u>	0	2.47	81.48	0	0	1.23	0	0	14.81	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	81.48	18.52	0	0	0	0
<u>Wave1</u>	0	0	0	0	7.41	92.59	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	100	0	0	0
<u>Pjump</u>	0	0	0	0	0	0	0	91.36	0	8.64
<u>Skip</u>	0	30	20	0	0	0	0	0	50	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.8 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 1 et une seule référence par mouvement type (30 coefficients, 89.25%).

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	100	0	0	0	0	0	0	0	0	0
<u>Run</u>	0	98.89	1.11	0	0	0	0	0	0	0
<u>Jump</u>	0	0	87.65	0	0	0	0	0	12.34	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	83.95	16.05	0	0	0	0
<u>Wave1</u>	0	0	0	0	7.41	92.59	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	100	0	0	0
<u>Pjump</u>	0	0	0	0	0	0	0	98.76	0	1.23
<u>Skip</u>	0	21.11	21.11	0	0	0	0	0	57.78	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.9 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 3 et une seule référence par mouvement type (100 coefficients, 91,96%).

global avec une seule référence (92.34%). On constate un pallier autour de 91% de reconnaissance pour les degrés 4, 5 et 8, puis le taux de reconnaissance diminue très rapidement pour une base de degré 12. Il semblerait qu'en augmentant le degré, l'approximation soit relativement précise, et réussisse à capter les différences entre les mouvements d'une même classe. En effet, certains auteurs travaillent sur la reconnaissance de la marche en tant qu'attribut biométrique d'un individu [Cuzzolin 06], [Elgammal 04], [Wang 07a]. Cela met en évidence que si la modélisation du mouvement est trop précise, elle prend en compte les caractéristiques personnelles de chaque acteur dans l'exécution de son mouvement et rend notre algorithme de classification moins robuste.

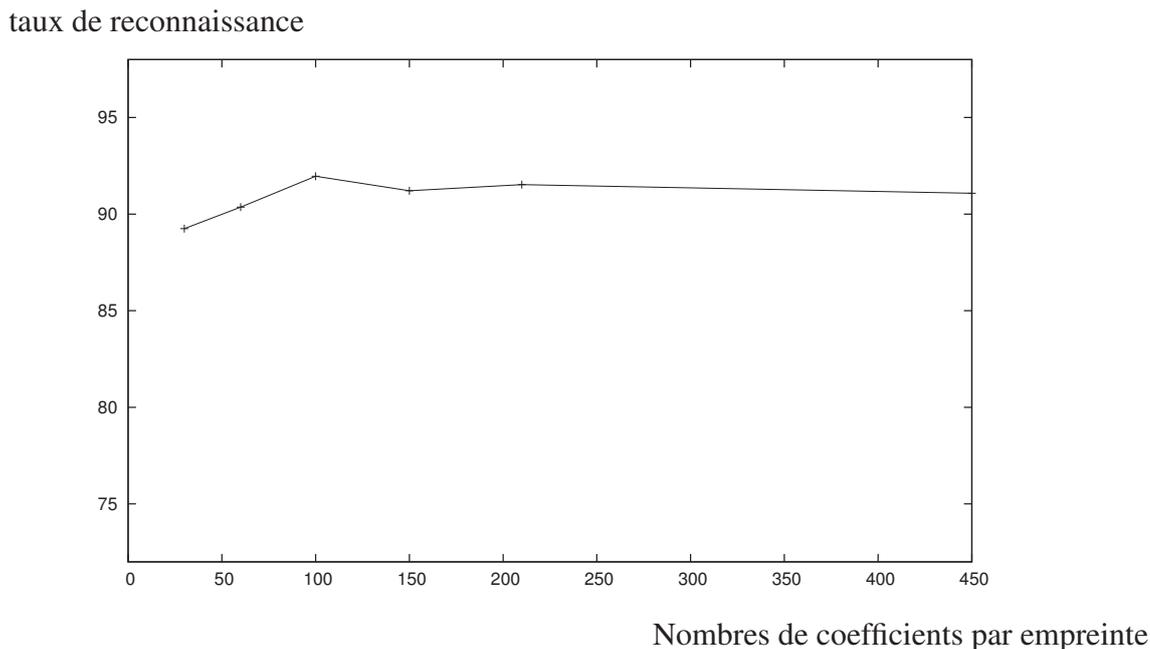


FIGURE 8.15 – Évolution du taux de reconnaissance en fonction du nombre de coefficients par empreinte pour l’algorithme hiérarchique avec une référence par mouvement type

8.5.6.2 Deux références par mouvement type

Le tableau TAB 8.10 présente les résultats avec une base de degré 1. Le taux de reconnaissance est de 94.08% et 9 classes sont reconnues à plus de 90%, dont 3 à 100%. Pour 6 combinaisons d’acteurs, seule la classe *skip* n’est pas reconnue à 100%. Le tableau TAB 8.11 présente les résultats avec une base de degré 3. Le taux de reconnaissance est de 96.22% et 9 classes sont reconnues à plus de 90%, dont 5 à 100%. Dans le cas où les références utilisées sont les acteurs 1 et 2, les acteurs 2 et 7 ou les acteurs 4 et 7, la base est reconnue à 100%. Dans 13 autres cas, seule la classe *skip* n’est pas reconnue à 100%. Comme avec l’algorithme global, l’augmentation du nombre de références améliore les résultats. Sur la figure FIG 8.16, nous présentons l’évolution du taux de reconnaissance en fonction du nombre de coefficients par empreinte. Nous constatons que le taux de classification maximum est ici de 96.22 avec une base de degré 3, ce qui correspond à 100 coefficients par empreinte. Comme avec une seule référence, nous constatons un pallier au niveau des degrés 4 et 5, cependant, après ce pallier, les résultats se rapprochent de ceux obtenus avec la base de degré 3. Même avec un nombre de références relativement faible, nous constatons que notre algorithme obtient des résultats supérieurs à certains algorithmes de la littérature [Ali 10], [Weinland 08], [Zhang 08], [Kellokumpu 08a] qui utilisent pourtant une stratégie “leave one out” ainsi que des classifieurs plus complexes que celui que nous utilisons.

8.5.7 Comparaison avec la littérature

Cependant, afin de nous comparer aux méthodes de la bibliographie, nous nous plaçons dans le contexte “Leave One Actor Out”. Comme précédemment, nous présentons d’abord les résultats de l’algorithme global, puis ceux de l’algorithme hiérarchique. Ici, les matrices de confusion

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	99.44	0.56	0	0	0	0	0	0	0	0
<u>Run</u>	0	99.72	0	0	0	0	0	0	0.28	0
<u>Jump</u>	0	0	90.43	0	0	0.62	0	0	8.95	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	90.74	9.26	0	0	0	0
<u>Wave1</u>	0	0	0	0	0.62	99.38	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	100	0	0	0
<u>Pjump</u>	0	0	0	0	0.31	0	0	91.67	0	8.02
<u>Skip</u>	0	17.22	13.33	0	0	0	0	0	69.44	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.10 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 1 et deux références par mouvement type (30 coefficients, 94.08%).

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	100	0	0	0	0	0	0	0	0	0
<u>Run</u>	0	99.44	0	0	0	0	0	0	0.56	0
<u>Jump</u>	0	0	95.06	0	0	0	0	0	4.94	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	91.97	8.02	0	0	0	0
<u>Wave1</u>	0	0	0	0	0	100	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	100	0	0	0
<u>Pjump</u>	0	0	0	0	0	0	0	98.76	0	1.23
<u>Skip</u>	0.278	9.44	13.33	0	0	0	0	0	76.94	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

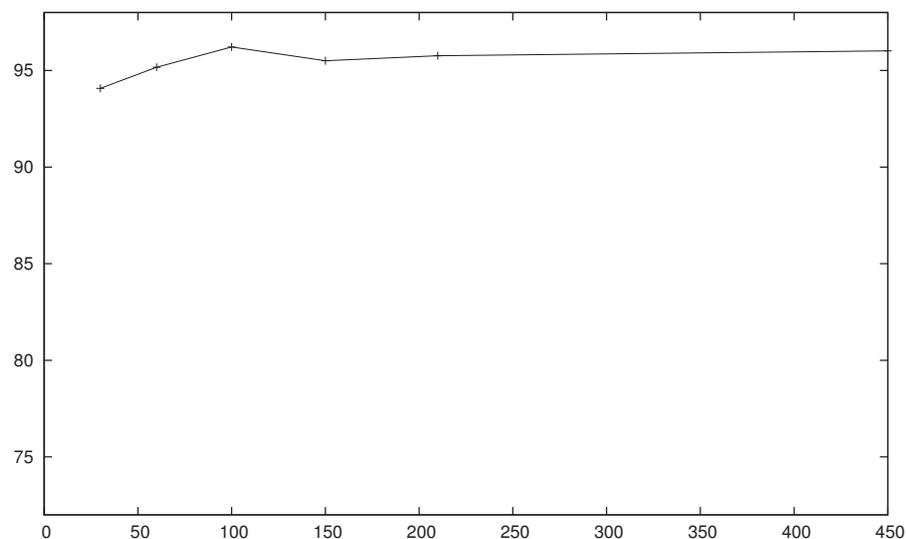
TABLEAU 8.11 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 3 et deux références par mouvement type (100 coefficients, 96.22%).

présentée ne sont plus des moyennes des différentes combinaisons d'ensemble d'apprentissage mais bien le résultat de la classification de la base de WEIZMANN.

8.5.7.1 Résultats de l'algorithme global avec une procédure LOAO

Dans un premier temps, nous présentons sur le tableau TAB 8.12 la matrice de confusion obtenue avec une base de degré 5 (42 coefficients). La base est classifiée avec un taux de reconnaissance de 98.89% et seule la vidéo de l'acteur 7 effectuant le mouvement skip est confondue avec une vidéo de la classe jump.

taux de reconnaissance



Nombres de coefficients par empreinte

FIGURE 8.16 – Évolution du taux de reconnaissance en fonction du nombre de coefficients par empreinte pour l'algorithme hiérarchique avec une référence par mouvement type

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	100	0	0	0	0	0	0	0	0	0
<u>Run</u>	0	100	0	0	0	0	0	0	0	0
<u>Jump</u>	0	0	100	0	0	0	0	0	0	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	100	0	0	0	0	0
<u>Wave1</u>	0	0	0	0	0	100	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	100	0	0	0
<u>Pjump</u>	0	0	0	0	0	0	0	100	0	0
<u>Skip</u>	0	0	11.11	0	0	0	0	0	88.89	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.12 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme global, une base de degré 5 et la stratégie “Leave One Actor Out” (42 coefficients, 98.89%).

À partir d'une base de degré 11, la base est classifiée sans erreur. Nous présentons sur le tableau TAB 8.13 la matrice de confusion obtenue avec une base de degré 11 (156 coefficients).

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	100	0	0	0	0	0	0	0	0	0
<u>Run</u>	0	100	0	0	0	0	0	0	0	0
<u>Jump</u>	0	0	100	0	0	0	0	0	0	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	100	0	0	0	0	0
<u>Wave1</u>	0	0	0	0	0	100	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	100	0	0	0
<u>Pjump</u>	0	0	0	0	0	0	0	100	0	0
<u>Skip</u>	0	0	0	0	0	0	0	0	100	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.13 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme global, une base de degré 11 et la stratégie “Leave One Actor Out” (156 coefficients, 100.00%).

8.5.7.2 Résultats de l’algorithme hiérarchique avec une procédure LOAO

Le tableau TAB 8.14 présente la matrice de confusion obtenue avec une base de degré 1 (30 coefficients). Le taux de reconnaissance est 97.78, les vidéos qui ne sont pas classifiées correctement sont l’acteur 6 et l’acteur 7 effectuant le mouvement skip, elles sont toutes les deux confondues avec la classe run. On peut remarquer que dans le cas de l’algorithme global avec une base de degré 5, la confusion avait lieu avec la classe jump.

	<u>W</u> <u>a</u> <u>t</u> <u>k</u>	<u>R</u> <u>u</u> <u>n</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u>
<u>Walk</u>	100	0	0	0	0	0	0	0	0	0
<u>Run</u>	0	100	0	0	0	0	0	0	0	0
<u>Jump</u>	0	0	100	0	0	0	0	0	0	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	100	0	0	0	0	0
<u>Wave1</u>	0	0	0	0	0	100	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	100	0	0	0
<u>Pjump</u>	0	0	0	0	0	0	0	100	0	0
<u>Skip</u>	0	0	22.22	0	0	0	0	0	77.78	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.14 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 1 et la stratégie “Leave One Actor Out” (30 coefficients, 97.78%).

Dans le cas d’une base de degré 2, il n’y a plus qu’une erreur. Il s’agit toujours de l’acteur 6 effectuant le mouvement skip qui est classifié comme un élément de la classe run. Avec une

base de degré 3 (100 coefficients), toute la base est classifiée sans erreur. Nous présentons sur le tableau TAB 8.15 la matrice de confusion obtenue avec une base de degré 3 pour l’algorithme de classification hiérarchique (100 coefficients). Nous constatons donc que l’algorithme hiérar-

	<u>W</u> <u>a</u> <u>k</u> <u>-</u>	<u>R</u> <u>u</u> <u>n</u> <u>-</u>	<u>J</u> <u>u</u> <u>m</u> <u>p</u> <u>-</u>	<u>S</u> <u>i</u> <u>d</u> <u>e</u> <u>-</u>	<u>B</u> <u>e</u> <u>n</u> <u>d</u> <u>-</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>1</u> <u>-</u>	<u>W</u> <u>a</u> <u>v</u> <u>e</u> <u>2</u> <u>-</u>	<u>P</u> <u>j</u> <u>u</u> <u>m</u> <u>p</u> <u>-</u>	<u>S</u> <u>k</u> <u>i</u> <u>p</u> <u>-</u>	<u>J</u> <u>a</u> <u>c</u> <u>k</u> <u>-</u>
<u>Walk</u>	100	0	0	0	0	0	0	0	0	0
<u>Run</u>	0	100	0	0	0	0	0	0	0	0
<u>Jump</u>	0	0	100	0	0	0	0	0	0	0
<u>Side</u>	0	0	0	100	0	0	0	0	0	0
<u>Bend</u>	0	0	0	0	100	0	0	0	0	0
<u>Wave1</u>	0	0	0	0	0	100	0	0	0	0
<u>Wave2</u>	0	0	0	0	0	0	100	0	0	0
<u>Pjump</u>	0	0	0	0	0	0	0	100	0	0
<u>Skip</u>	0	0	0	0	0	0	0	0	100	0
<u>Jack</u>	0	0	0	0	0	0	0	0	0	100

TABLEAU 8.15 – Tableau des résultats de classification de la base de vidéos de WEIZMANN avec notre algorithme hiérarchique, une base de degré 3 et la stratégie “Leave One Actor Out” (100 coefficients, 100.00%).

chique permet de classifier la base de WEIZMANN avec moins de coefficients que l’algorithme global.

8.5.7.3 Synthèse

Nous avons donc réussi à classifier sans erreur une base de vidéos de la littérature. Nous donnons dans le tableau TAB 8.16 quelques résultats de la littérature afin de mieux situer nos résultats. Nous rappelons que nous n’utilisons que notre descripteur de mouvement pour effectuer la classification. Nous pouvons donc en conclure que notre descripteur est adapté à la reconnaissance de mouvements humains.

Cependant, nous avons classifié chaque vidéo dans son intégralité. Il serait néanmoins intéressant de classifier la vidéo par partie, afin de pouvoir classifier des vidéos composées de plusieurs mouvements différents. Cette stratégie est utilisée dans [Blank 05] pour des groupes de 10 images par pas de 5 images et dans [Gorelick 07] pour des groupes de 8 images par pas de 4 images. De plus, si la base de WEIZMANN est très intéressante pour valider notre méthode, il serait intéressant de se confronter à une base plus complexe telle que la base KTH [Schuldt 04] par exemple.

8.6 Conclusion

Dans ce chapitre, nous avons présenté une utilisation des bases de polynômes bivariées dans le cadre de l’analyse de mouvements humains. Pour cela, nous nous sommes placés dans le contexte de la vision par ordinateur. Dans un premier temps, nous avons présenté une mesure

Méthode	Résultats (%)	descripteur(s)	classifieur	ensemble d'apprentissage
méthode hiérarchique	100	Base polynomiale degré 3 (100 coefficients)	somme des similarités	leave one actor out
méthode hiérarchique	97.78	Base polynomiale degré 1 (30 coefficients)	somme des similarités	leave one actor out
méthode globale	100	Base polynomiale degré 11 (156 coefficients)	somme des similarités	leave one actor out
méthode globale	98.89	Base polynomiale degré 5 (42 coefficients)	somme des similarités	leave one actor out
[Fathi 08]	100.0	descripteur [Efros 03] + 5 ^{ème} composante sur volume spatio-temporel centré sur des silhouette	AdaBoost	leave one out
[Tran 08]	100.0	flot optique (\mathcal{U} , \mathcal{V}) + silhouette + ACP + Histogrammes locaux	1PPV + apprentissage de métrique	leave one actor out
[Ali 10]	95.75	composantes cinématiques + ACP	MIL	leave one out
[Weinland 08]	93,33	silhouette ou contours	Wrapper [John 94]	leave one out
[Zhang 08]	92.89	“Motion context”	SVM	leave one out
[Kellokumpu 08a]	95.6	LBP-TOP (sur les images)	HMM	leave one out
[Kellokumpu 08b]	97.8	LBP (sur MHI et MEI)	HMM	leave one out
[Jhuang 07]	98.8	Modélisation du système visuel humain	SVM	$\frac{2}{3}$ de la base
[Wang 07b]	97.78	AME et MMS	k PPV	leave one out
[Blank 05] et [Gorelick 07]	100	volume spatio-temporel (équations de Poisson)	1PPV (distance euclidienne)	leave one out

TABLEAU 8.16 – Résultats de classification de la base de WEIZMANN de la littérature

de similarité s'appuyant directement sur notre modèle polynomial. Cette mesure permet d'évaluer la similarité entre deux champs de déplacements. En définissant une action humaine comme une succession de mouvements élémentaires, nous avons proposé deux applications. La première consiste à détecter et reconnaître un mouvement dans une vidéo dans un contexte de commande de jeu vidéo par exemple. Nous avons obtenu des résultats intéressants avec une complexité algorithmique satisfaisante pour envisager une mise en oeuvre en temps réel de cette application. Par ailleurs, afin de valider ces résultats, nous avons confronté notre descripteur de mouvement à une base de tests de la littérature. Ainsi nous avons proposé une deuxième application, qui concerne cette fois-ci la classification de vidéos de mouvements humains. Sur la base choisie, nous avons obtenu 100% de classification correcte, plaçant ainsi notre méthode à la hauteur des meilleures de la littérature.

Cependant, il serait intéressant de pousser plus loin l'étude de ce descripteur. D'abord en se confrontant à une base plus complexe que la base de WEIZMANN, et enfin en appliquant notre descripteur de mouvement à d'autres types de mouvements non nécessairement humains. Il est important de noter que les bases polynomiales peuvent être étendues aux dimensions supérieures, il serait alors possible de modéliser des blocs spatio-temporels. De plus, il est possible d'utiliser notre descripteur conjointement à d'autres descripteurs au sein d'un système de classification plus élaboré de type SVM.

Quatrième partie

Conclusion et perspectives

CHAPITRE 9

CONCLUSION ET PERSPECTIVES

Conclusion

Dans ce manuscrit, nous avons étudié la modélisation du mouvement. Le mouvement est estimé à partir d'une séquence d'images et présenté sous la forme de champs de déplacements. Nous avons introduit les principaux algorithmes d'estimation du mouvement ainsi qu'un grand nombre d'applications utilisant cette information dynamique. Forts du constat que toutes ces méthodes nécessitent une étape de modélisation du mouvement, nous avons présenté les principales méthodes de modélisation utilisées en traitement du signal et des images. Nous avons fait le choix d'utiliser un modèle polynomial puisque ceux-ci sont très bien adaptés à l'analyse du mouvement, notamment les mouvements polynomiaux de degré 1 qui permettent de décrire de nombreux phénomènes. De plus, ils sont très simples à analyser, les mouvements de caméra, par exemple, peuvent être très facilement analysés à partir d'une modélisation polynomiale de degré 1. L'originalité de notre approche consiste à utiliser un modèle polynomial estimé par projection sur une base de polynômes orthogonaux bivariés.

Cette méthode a été présentée par MARTIN DRUON [Druon 09]. Il utilise les bases de polynômes bivariés pour analyser des champs de déplacements issus de la mécanique expérimentale des fluides. Cet outil s'appuie sur un formalisme mathématique établi que nous avons présenté en expliquant, entre autres, l'importance de l'orthogonalité. En effet, cette propriété des polynômes nous permet de nous affranchir de l'estimation au sens des moindres carrés et d'obtenir les coefficients polynomiaux par projection d'un champ de déplacements sur la base de polynômes. Nous avons présenté quelques exemples d'analyse à partir de la modélisation globale d'un champ de déplacements avec des bases de différents degrés. Nous avons constaté, avec ce type de modélisation, qu'il peut être nécessaire d'utiliser des bases de degrés relativement élevés pour modéliser certains phénomènes. Cette augmentation implique une difficulté supplémentaire dans l'analyse des coefficients polynomiaux qui est pourtant très simple en degré 1. Pour surmonter cela et s'adapter à de nombreuses applications, nous avons montré qu'il est possible d'obtenir une subdivision irrégulière permettant une modélisation très efficace à partir de stratégies de subdivisions classiques telles que le quadtree ou le kd-tree. Ces méthodes permettent de garantir une qualité de reconstruction d'un champ de déplacements

complexe avec des bases de polynômes de degré faible tout en obtenant un taux de compression relativement élevé. Cependant, même avec des bases de degrés élevés ou avec ces techniques de subdivisions, nous avons constaté que les champs épars sont difficilement modélisables par base de polynômes bivariés. Nous avons proposé une nouvelle méthode de génération des bases polynomiales “éparses” prenant en compte le domaine spatial lors de la modélisation, afin de permettre la modélisation des champs épars. Il est alors possible de modéliser les champs épars avec une grande qualité d’approximation et cela nous offre la possibilité d’obtenir une interpolation des vecteurs nuls en transformant les coefficients de la base éparsée dans la base canonique. De plus, nous pouvons aussi utiliser toutes les méthodes de subdivisions étudiées au chapitre précédent sur ce type de champs. Enfin, nous nous sommes intéressés à la modélisation à erreur nulle à partir des bases de polynômes orthogonaux. Nous avons d’abord montré la correspondance entre le polynôme d’interpolation de Lagrange et les bases polynomiales dites “A-complètes”. À partir de ce résultat, nous avons comparé les bases polynomiales A-complètes avec la décomposition en valeurs singulières et avec la transformée en cosinus discrets. Enfin, nous avons utilisé les bases A-complètes pour réaliser une approximation multi-résolutions des champs de déplacements.

Finalement, nous avons présenté deux applications des bases de polynômes orthogonales relatives à deux domaines de recherche différents. La première application concerne la détection de points singuliers dans les champs de déplacements. Ces points qui représentent, par exemple, les centres des tourbillons sont très importants pour l’analyse des écoulements en mécanique expérimentale des fluides. Nous avons ensuite proposé trois schémas multi-résolutions s’appuyant sur notre modèle polynomial. L’un de ces algorithmes utilise la décomposition multi-résolutions obtenue à partir des bases A-complètes. Nous avons montré que notre estimateur de mouvement affine est plus efficace et plus robuste au bruit Gaussien que celui utilisé par RAO et JAIN [Rao 92a]. Nous avons aussi comparé nos résultats à ceux de la bibliographie avec succès.

La deuxième application concerne le domaine de l’analyse du mouvement humain à partir de notre modèle polynomial. Nous appelons ce descripteur empreinte polynomiale. Nous avons introduit une mesure de similarité des champs de déplacements s’appuyant sur la norme matricielle du tenseur défini à partir de deux empreintes. Cette mesure nous a permis de présenter deux algorithmes d’analyse du mouvement humain. Le premier algorithme permet de détecter et de reconnaître le mouvement humain dans une vidéo à partir d’une référence unique. Ce descripteur est appliqué à la reconnaissance de cinq mouvements de boxe effectués par un utilisateur positionné devant une caméra basse résolution de type webcam. Nous avons obtenu des résultats encourageants. Cependant, afin de comparer notre descripteur à ceux de la bibliographie, nous avons proposé un algorithme de classification de mouvement très simple et nous l’avons appliqué sur les données d’une base de mouvements humains très utilisée dans la littérature. Nous obtenons des résultats comparables aux méthodes les plus efficaces tout en étant plus simple à mettre en œuvre. En effet, notre méthode ne repose pas sur un classifieur tel que les modèles de Markov caché, les séparateurs vastes marges ou les méthodes de type AdaBoost... Dans cette partie, nous avons donc appliqué notre modèle polynomial à deux applications appartenant chacune à un domaine de recherche différent. Dans les deux cas, nos résultats sont comparables à ceux de la littérature. Les avantages de notre modèle sont sa simplicité de mise en œuvre tant au niveau de la programmation informatique de la méthode que de la rapidité d’exécution lors de l’utilisation.

Perspectives

De nombreuses pistes restent à explorer. Tout d'abord, nous proposons quelques perspectives théoriques à ces travaux.

Dans ce manuscrit, nous avons utilisé des bases de polynômes en dimension 2. Celles-ci sont évidemment généralisables en toutes dimensions. Il serait intéressant d'étudier l'apport des bases de polynômes de dimensions supérieures dans différents contextes. Nous avons, par exemple, utilisé une approche par portrait de phase sur des champs de déplacements en dimension 2 or, il est possible d'utiliser cette approche sur des champs en dimension 3 [Denis-Brossard 00]. Dans un autre contexte, les bases polynomiales spatio-temporelles pourraient apporter une information importante pour la modélisation du mouvement humain avec une empreinte de mouvement qui prendrait en compte l'ordre temporel de l'exécution d'un mouvement. Dans ce manuscrit, les bases A-complètes n'ont été utilisées dans un contexte applicatif que pour l'un de nos algorithmes multi-résolutions de détection de singularités. Ces bases semblent offrir beaucoup d'autres possibilités en raison de la nature polynomiale des fonctions d'approximations. Quelques résultats préliminaires nous laissent à penser que l'utilisation de bases de polynômes orthogonales A-complètes en dimension 3 (spatio-temporelle) permettrait d'obtenir une méthode d'estimation des champs de déplacements avec une approche par tenseur de structure relativement efficace. En effet, à partir d'un modèle polynomial, le calcul de dérivées spatio-temporelles est trivial. De même, l'utilisation de bases A-complètes en dimension 3 permettrait d'obtenir une analyse multi-résolutions spatio-temporelle.

Du point de vue applicatif, plusieurs méthodes de subdivision étudiées n'ont pas été exploitées dans le cadre d'une application précise pour en valider l'utilité. C'est le cas, par exemple, du diagramme de Voronoï. Ce pavage de l'espace, dual à la triangulation de Delaunay, semble adapté à un certain nombre de travaux qui nécessitent la résolution d'équations différentielles aux dérivés partielles. C'est le cas par exemple de la simulation d'écoulement, qui a fréquemment recours à la triangulation de Delaunay pour la résolution d'équations différentielles aux dérivées partielles. Le formalisme mathématique des bases de polynômes étant très adapté aux calculs de dérivées, pourrait alors offrir de nouvelles possibilités. De plus, ces méthodes reposent souvent sur des méthodes d'interpolations. Nous avons montré que notre modèle polynomial permet de réaliser des interpolations en calculant le passage des coefficients obtenus dans une base de polynômes éparses vers la base canonique.

Dans le cas de l'analyse de mouvements humains, nous avons volontairement proposé un algorithme simple afin de mettre en évidence la robustesse de notre descripteur de mouvement. La base de vidéos de WEIZMANN a été classifiée avec succès mais cette base est l'une des plus simples de la littérature, puisque les fonds sont toujours les mêmes, qu'il y a pas de mouvement de caméra parasite et que malgré la faible définition spatiale, la qualité des images est très acceptable. Il serait donc intéressant d'inclure le descripteur de mouvement dans un schéma de classification plus complet avec un classifieur évolué afin de le tester sur des vidéos plus complexes.

De plus, un grand nombre des domaines d'applications mentionnés au chapitre 2 n'ont pas été étudiés ici. Notre modèle polynomial pourrait être applicable à un grand nombre d'entre eux. Nous prenons comme exemple la segmentation au sens du mouvement. En effet, il est envisageable de façon assez rapide de proposer un algorithme de segmentation au sens du mouvement reposant sur une méthode de type division-fusion à partir du résultat obtenu avec un quadtree.

De manière générale, notre modèle s'appuie sur des fondements mathématiques solides, indépendants de toute application physique. Il est alors possible d'envisager d'appliquer ce modèle à beaucoup d'autres domaines relatifs à l'analyse des champs de déplacements.

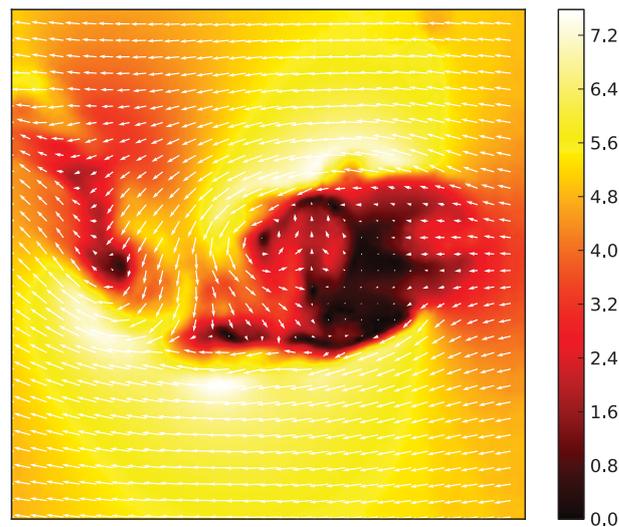
ANNEXE A

SQUENCE CYLINDRE-CARRÉ

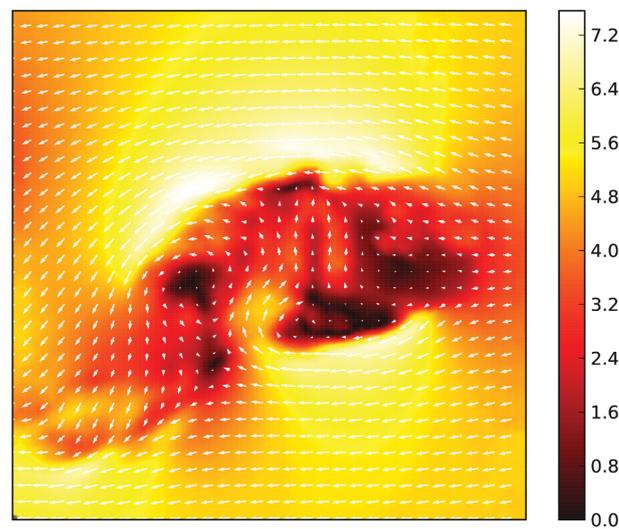
Sur la figure FIG A.1, nous présentons deux champs de la séquence Cylindre-Carré. Nous présentons sur la figure FIG A.1(a) le champ 100 de cette séquence et sur la figure FIG A.1(b) le champ 131 de cette séquence. Ce champ est issu d'une séquence présentée dans les travaux de DAVID et AL. [David 06] ainsi que dans les travaux de KUTURLUS et AL. [Kurtulus 07]. Cette séquence a aussi été utilisée dans les travaux de MARTIN DRUON [Druon 09]. Cette séquence a été fournie par le département Fluides, Thermique et Combustion de l'Institut P'. Elle représente l'évolution de lâchers tourbillonnaires alternés dans le sillage d'un cylindre carré. L'acquisition est effectuée au sein de la soufflerie Sessiat de l'Institut P' qui fonctionne en boucle fermée (cf. FIG A.3). Sa plage de vitesse est comprise entre 1.5 m/s et 30 m/s. Un cylindre carré de 30 mm de côté est placé dans la veine d'essai (cf. FIG A.3). Le fluide utilisé est de l'air dont la vitesse en amont du cylindre est de 2.37 m/s. Le nombre de Reynolds, calculé à partir du diamètre du cylindre est de $Re = 4900$. Pour un tel nombre de Reynolds, l'écoulement est dans un régime turbulent fortement instationnaire. Nous étudions cet écoulement pendant un intervalle de temps représentant 10 périodes d'échappements tourbillonnaires. Durant cet intervalle, la séquence peut alors être considérée comme de cycle limite.

L'acquisition est effectuée avec une caméra CMOS Photron-Fastcam dont la fréquence d'acquisition est de 2000 Hz pour une résolution de 1024×1024 pixels avec une sensibilité de 8 bits en niveau de gris. L'objectif utilisé est un AF Nikkor de 105 mm permettant d'obtenir un champ de 145×145 mm². Le laser utilisé pour illuminer la section de mesure est un ND-YLF Quantronix qui possède deux cavités ayant chacune une puissance de 18 mJ par pulse. L'ensemencement de l'écoulement est réalisé grâce à un générateur de fumée. Le diamètre moyen des particules est de 1 μ m. Un exemple d'images obtenues avec ce dispositif d'acquisition est présenté FIG A.4.

Les champs de déplacements sont calculés grâce à une méthode itérative d'inter-corrélation avec fenêtres déformables [Scarano 00, Scarano 02] de 31×31 pixels et un taux de recouvrement de 75 %. La résolution de ces champs est de 124×124 vecteurs. Afin de réduire les bruits de mesure non corrélés, une approximation en temps sur un polynôme du second ordre par minimisation aux moindres carrés est effectuée sur une fenêtre glissante composée de cinq champs de déplacements. Les données présentées ici sont adimensionnées par rapport à la taille du cylindre carré et à la vitesse en amont de la zone étudiée. La séquence est composée de 976 champs.



(a)



(b)

FIGURE A.1 – Deux champs de la séquence Cylindre-Carré; (a) Champ 100 de la séquence Cylindre-Carré; (b) Champ 131 de la séquence Cylindre-Carré.

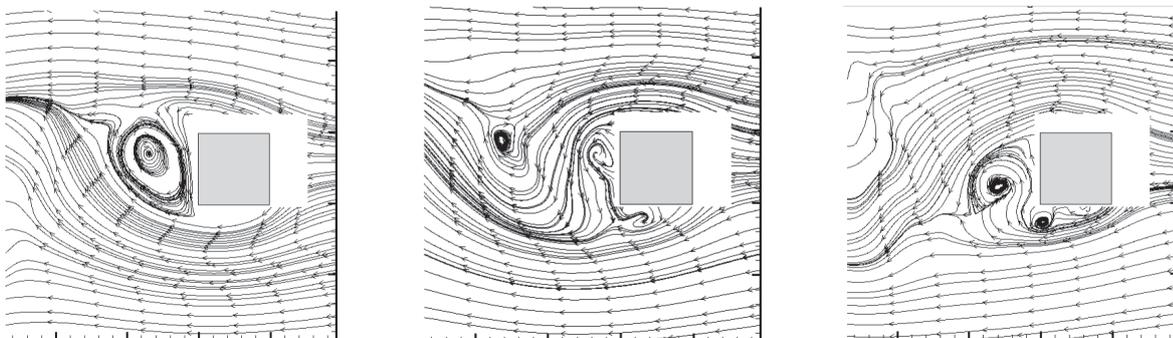
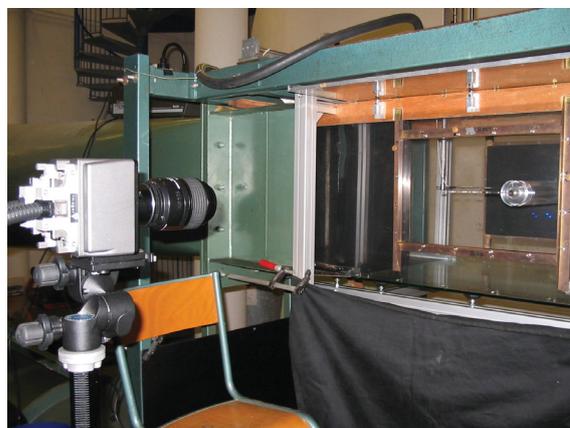


FIGURE A.2 – Séquence « cylindre-carré » : topologie de l'écoulement à trois instants différents. Le fluide s'écoule de la droite vers la gauche.



(a) - Soufflerie Sessiat du LEA



(b) - Veine d'essai et dispositif d'acquisition

FIGURE A.3 – Séquence « cylindre-carré » : représentation du dispositif expérimental comprenant (a) - la soufflerie Sessiat du Laboratoire d'Études Aérodynamiques (LEA) de Poitiers et (b) - la veine d'essai et le dispositif d'acquisition.

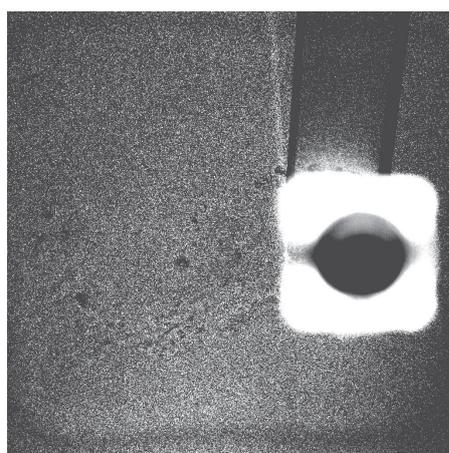


FIGURE A.4 – Séquence « cylindre-carré » : exemple d'image obtenue avec le dispositif d'acquisition.

ANNEXE B

COMPLEXITÉ ET OCCUPATION MÉMOIRE DES BASES POLYNOMIALES DENSES

Complexité et occupation mémoire

Nous proposons d'étudier la complexité et l'occupation mémoire de la génération d'une base orthogonale de polynômes bivariables. Nous donnons ces résultats en fonction du choix de l'implantation informatique que nous avons utilisé.

Soit C_+ , C_- , C_\times , $C_/$, le coût respectif d'une addition, d'une soustraction, d'une multiplication et d'une division. Soit M , le nombre de lignes du champ de déplacements et N , le nombre de colonnes.

Le coût de génération d'un polynôme de la base est :

$$C_P = (5 \times N \times M \times C_\times) + (C_/) + (2 \times N \times M \times C_-) \quad (\text{B.1})$$

Le nombre de polynômes d'une base de degré D étant $\frac{(D+1) \times (D+2)}{2}$, le coût de la génération d'une base de polynômes est :

$$C_B = \frac{(D+1) \times (D+2)}{2} \times C_P \quad (\text{B.2})$$

Nous rappelons que la génération d'une base de polynômes ne dépend pas des données. Cependant, par ce type d'implantation informatique, le calcul de la base de polynômes dépend des dimensions spatiales du champ de déplacements.

Les polynômes calculés sont ensuite stockés en mémoire. Nous donnons donc des informations sur l'occupation mémoire des polynômes. L'occupation mémoire d'un polynôme est donnée par l'équation :

$$S_P = N \times M \times S_r \quad (\text{B.3})$$

où S_r est l'occupation mémoire d'un réel.

L'occupation mémoire d'une base est alors de

$$S_B = \frac{(D + 1) \times (D + 2)}{2} \times S_P \quad (\text{B.4})$$

L'occupation mémoire d'une base polynomiale de degré 16 pour un champ de déplacements de 124×124 vecteurs avec des réels codés sur 8 octets est d'environ 18 Mo. Pour un champ de déplacements de 1024×1024 vecteurs et une base de même degré, l'occupation est alors de 1224 Mo.

Nous donnons maintenant le coût de la projection d'un champ sur un polynôme de la base.

$$C_{proj} = (M \times N \times C_{\times}) + 1 \times C_{/} \quad (\text{B.5})$$

Par extension, le coût d'une projection sur l'ensemble des polynômes de la base est de

$$C_{PROJ} = \frac{(D + 1) \times (D + 2)}{2} \times C_{proj} \quad (\text{B.6})$$

BIBLIOGRAPHIE

- [Ahad 08] M.A.R. Ahad, T. Ogata, J.K. Tan, HS Kim & S. Ishikawa. Motion recognition approach to solve overwriting in complex actions. In Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on, pages 1–6. IEEE, 2008.
- [Ali 10] Saad Ali & Mubarak Shah. Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, pages 288–303, 2010.
- [Altunbasak 98] Y. Altunbasak, P.E. Eren & A.M. Tekalp. Region-Based Parametric Motion Segmentation Using Color Information* 1. Graphical models and image processing, vol. 60, no. 1, pages 13–23, 1998.
- [Andres 91] E. Andres. Cercles Discrets et Rotations Discrettes. In 1er colloque DGCI Géométrie Discrète en Imagerie, Strasbourg (France), September 1991.
- [Anosov 88] DV Anosov & V.I. Arnold. Encyclopedia of mathematical sciences volume 1 : Dynamical systems i :ordinary differential equations and smooth dynamical systems, volume 1. Springer-Verlag, New York, 1988.
- [Augereau 05] B. Augereau, B. Tremblais & C. Fernandez-Maloigne. Vectorial Computation of the Optical Flow in Color Image Sequences. In Thirteenth Color Imaging Conference, pages 130–134, November 2005.
- [Ayer 94] S. Ayer, P. Schroeter & J. Big
"un. Segmentation of moving objects by robust motion parameter estimation over m
Computer Vision - ECCV'94, pages 316–327, 1994.
- [Barjatya 07] A. Barjatya. Block matching algorithms for motion estimation. In IEEE Conference on Computational Intelligence and Multimedia Applications, volume 3, pages 89–96, 2007.
- [Barron 94] J.L. Barron, D.J. Fleet & S.S. Beauchemin. Performance of Optical Flow Techniques. International Journal of Computer Vision, vol. 12, no. 1, pages 43–77, 1994.
- [Baszenski 97] G. Baszenski & M. Tasche. Fast polynomial multiplication and convolutions related
Linear Algebra and its Applications, vol. 252, no. 1-3, pages 1–25, 1997.

- [Bay 06] H. Bay, T. Tuytelaars & L. Van Gool. Surf : Speeded up robust features. Computer Vision–ECCV 2006, pages 404–417, 2006.
- [Belongie 00] S. Belongie, J. Malik & J. Puzicha. Shape context : A new descriptor for shape matching and object recognition. In In NIPS. Citeseer, 2000.
- [Ben-Ari 09] Rami Ben-Ari & Nir Sochen. A Geometric Framework and a New Criterion in Optical Flow Modeling. J. Math. Imaging Vis., vol. 33, pages 178–194, February 2009.
- [Bentley 75] Jon Louis Bentley. Multidimensional Binary Search Trees Used for Associative Searching. Commun. ACM, vol. 18, no. 9, pages 509–517, 1975.
- [Besag 86] J. Besag. On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society. Series B (Methodological), pages 259–302, 1986.
- [Bigun 91] J. Bigun, G.H. Granlund & J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 8, pages 775–790, 1991.
- [Black 96a] Michael J. Black & P. Anandan. The robust estimation of multiple motions : parametric and piecewise-smooth flow fields. Comput. Vis. Image Underst., vol. 63, pages 75–104, January 1996.
- [Black 96b] M.J. Black & A.D. Jepson. Estimating optical flow in segmented images using variable-or. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, pages 972–986, 1996.
- [Blank 05] M. Blank, L. Gorelick, E. Shechtman, M. Irani & R. Basri. Actions as space-time shapes. In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 2, pages 1395–1402, 2005.
- [Blum 73] H. Blum. Biological shape and visual science (Part I). Journal of theoretical Biology, vol. 38, no. 2, pages 205–287, 1973.
- [Bobick 96] A. Bobick & J. Davis. An appearance-based representation of action. In International Conference on Pattern Recognition, volume 13, pages 307–312. Citeseer, 1996.
- [Bouthemy 99] P. Bouthemy, M. Gelgon & F. Ganansia. A unified approach to shot change detection and camera motion characterization. IEEE Transaction on Circuits and Systems for Video Technology, vol. 9, no. 7, pages 1030–1044, 1999.
- [Boyd 01] J.E. Boyd. Video phase-locked loops in gait recognition. 2001.
- [Bruno 00] E. Bruno & D. Pellerin. Global motion fourier series expansion for video indexing and ret. Advances in Visual Information Systems, pages 185–200, 2000.
- [Bruno 01] E. Bruno & D. Pellerin. Global motion model based on B-spline wavelets : application to. In Proc. of the 2nd Int. Symposium. on Image and Signal Processing and Analysis, ISPA'01. Citeseer, 2001.

- [C.-F. Shu 94] R.C C.-F. Shu Jain. Vector field analysis for oriented patterns. Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 9, pages 946–950, Sep 1994.
- [Cai 96] W. Cai & J. Wang. Adaptive Multiresolution Collocation Methods for Initial-Boundary Value Problems of Nonlinear PDEs. SIAM Journal on Numerical Analysis, vol. 33, no. 3, pages 937–970, 1996.
- [Chang 97] M.M. Chang, A.M. Tekalp & M.I. Sezan. Simultaneous motion estimation and segmentation. IEEE Transactions on Image Processing, vol. 6, no. 9, pages 1326–1333, 1997.
- [Chaudhry 09] R. Chaudhry, A. Ravichandran, G. Hager & R. Vidal. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 0, pages 1932–1939, 2009.
- [Chen 06] H.S. Chen, H.T. Chen, Y.W. Chen & S.Y. Lee. Human action recognition using star skeleton. In Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks, pages 171–178. ACM, 2006.
- [Chong 90] MS Chong, AE Perry & BJ Cantwell. A general classification of three-dimensional flow fields. In Topological Fluid Mechanics : Proceedings of the IUTAM Symposium, Cambridge, UK, 13-18 August 1989, page 408. Cambridge Univ Pr, 1990.
- [Chu 00] E.C. Chu & A. George. Inside the FFT black box : serial and parallel fast Fourier transform algorithms. CRC, 2000.
- [Cohen 96] I. Cohen & I. Herlin. Optical flow and phase portrait methods for environmental satellite image motion estimation. Lecture Notes in Computer Science, vol. 1065, pages 141–150, 1996.
- [Cordier 02] L. Cordier & M. Bergmann. Proper orthogonal decomposition : an overview. Lecture series, vol. 4, 2002.
- [Corpetti 03] T. Corpetti, E. Memin & P. Perez. Extraction of singular points from dense motion fields : an analytic approach. Journal of mathematical imaging and vision, vol. 19, no. 3, pages 175–198, 2003.
- [Cremers 03] Daniel Cremers. A Variational Framework for Image Segmentation Combining Motion Estimation and Shape Regularization. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 1, page 53, 2003.
- [Cremers 05] D. Cremers & S. Soatto. Motion competition : A variational approach to piecewise planar motion estimation. International Journal of Computer Vision, vol. 62, no. 3, pages 249–265, 2005.
- [Cuzzolin 06] F. Cuzzolin. Using bilinear models for view-invariant action and identity recognition. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 2, pages 1701–1708. IEEE, 2006.

- [Danafar 07] Somayeh Danafar & Niloofar Gheissari. Action Recognition for Surveillance Applications Using Optic Flow and SVM. In Yasushi Yagi, Sing Kang, In Kweon & Hongbin Zha, éditeurs, *Computer Vision - ACCV 2007*, volume 4844 of *Lecture Notes in Computer Science*, pages 457–466. Springer Berlin Heidelberg, 2007.
- [Daubechies 88] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, vol. 41, no. 7, pages 909–996, 1988.
- [Daubechies 92] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial Mathematics, 1992.
- [David 05] L. David. Habilitation à diriger des recherches. PhD thesis, Université de Poitiers, 2005.
- [David 06] L. David, F. Scarano & P. Braud. Accompagnement de la mesure de vitesse par PIV-RT ou double plans : avantages et limitations. In *Proceedings of the Congrès Francophone de Techniques Laser*, pages 273–280, 2006.
- [Davis 97] J.W. Davis & A.F. Bobick. The representation and recognition of action using temporal templates. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 928–934, 1997.
- [Dementhon 06] D. Dementhon & D. Doermann. Video retrieval of near-duplicates using κ -nearest neighbor retrieval of spatio-temporal descriptors. *Multimedia Tools and Applications*, vol. 30, no. 3, pages 229–253, 2006.
- [Demonceaux 04] C. Demonceaux, A. Potelle & D. Kachi-Akkouche. Obstacle detection in a road scene based on motion analysis. *Vehicle Technology, IEEE Transactions on*, vol. 53, no. 6, pages 1649–1656, 2004.
- [Denis-Brossard 00] M.A. Denis-Brossard. Modélisation multiéchelle de structures bidimensionnelles fluides. Application à l'analyse du mouvement de masses nuageuses atmosphériques. PhD thesis, Université René Descartes - Paris V, 2000.
- [Do Carmo 76] M.P. Do Carmo & M.P. Do Carmo. *Differential geometry of curves and surfaces*, volume 2. Prentice-Hall Englewood Cliffs, NJ, 1976.
- [Druon 06] M. Druon, B. Tremblais, B. Augereau, F. Billy & L. David. Algorithmes pour la vélocimétrie par imagerie de particules résolue dans le temps. In *Congrès Francophone de Techniques Laser (CFTL)*, pages 243–252, Toulouse, France, Septembre 2006.
- [Druon 09] M. Druon. Modélisation du mouvement par polynômes orthogonaux : application à l'étude d'écoulements fluides. PhD thesis, Université de Poitiers, 2009.
- [Durik 01] M. Durik & J. Benois-Pineau. Robust Motion Characterisation for Video Indexing Based on MPEG2 optical flow. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, pages 57–64, 2001.
- [Eaton 02] John W. Eaton. *Gnu octave manual*. Network Theory Limited, 2002.
- [Eckart 36] C. Eckart & G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, vol. 1, no. 3, pages 211–218, 1936.

- [Efros 03] A.A. Efros, A.C. Berg, G. Mori & J. Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, volume 2, pages 726–733, 2003.
- [Elgammal 04] A. Elgammal & C.S. Lee. Separating style and content on a nonlinear manifold. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–478. IEEE, 2004.
- [Erol 07] A. Erol, G. Bebis, M. Nicolescu, R.D. Boyle & X. Twombly. Vision-based hand pose estimation : A review. *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pages 52–73, 2007.
- [Escobar 09] M.J. Escobar, G.S. Masson, T. Vieville & P. Kornprobst. Action recognition using a bio-inspired feedforward spiking network. *International journal of computer vision*, vol. 82, no. 3, pages 284–301, 2009.
- [Fathi 08] A. Fathi & G. Mori. Action recognition by learning mid-level motion features. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [Fleet 05] David Fleet & Yair Weiss. Optical flow estimation, volume 19, pages 1–10. Springer, 2005.
- [Ford 94] R.M. Ford, R.N. Strickland & B.A. Thomas. Image models for 2-D flow visualization and compression. *CVGIP : Graphical Models and Image Processing*, vol. 56, no. 1, pages 75–93, 1994.
- [Fujiyoshi] H. Fujiyoshi & AJ Lipton. Real-time human motion analysis by image skeletonization. In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pages 15–21. IEEE.
- [Fujiyoshi 04] H. Fujiyoshi, A.J. Lipton & T. Kanade. Real-time human motion analysis by image skeletonization. *IEICE Transactions on Information and Systems E Series D*, vol. 87, no. 1, pages 113–120, 2004.
- [Galvin 98] B. Galvin, B. McCane, K. Novins, D. Mason & S. Mills. Recovering Motion Fields : An Evaluation of Eight Optical Flow Algorithms. In *British Machine Vision Conference, 1998*.
- [Gelgon 98] M. Gelgon. Segmentation spatio-temporelle et suivi dans une séquence d'images : application à la structuration et à l'indexation de vidéo. PhD thesis, Université de Rennes 1, Télécommunications et Traitement du Signal, December 1998.
- [Gelgon 00] M. Gelgon & P. Bouthemy. A region-level motion-based graph representation and la. *Pattern Recognition*, vol. 33, no. 4, pages 725–740, 2000.
- [Giese 03] M.A. Giese & T. Poggio. Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, vol. 4, no. 3, pages 179–192, 2003.
- [Gorelick 06] L. Gorelick, M. Galun, E. Sharon, R. Basri & A. Brandt. Shape representation and classification using the poisson equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1991–2005, 2006.

- [Gorelick 07] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani & Ronen Basri. Actions as Space-Time Shapes. Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pages 2247–2253, December 2007.
- [Harris 88] C. Harris & M. Stephens. A combined corner and edge detector. In Alvey vision conference, volume 15, page 50. Manchester, UK, 1988.
- [Heitz 10] D. Heitz, E. Mémin & C. Schnörr. Variational fluid flow measurements from image sequences : synopsis and perspectives. Exp. Fluids, vol. 48, no. 3, pages 369–393, 2010.
- [Hoey 00] J. Hoey & J.J. Little. Representation and recognition of complex human motion. In cvpr, page 1752. Published by the IEEE Computer Society, 2000.
- [Hoey 02] J. Hoey. Clustering Contextual Facial Display Sequences. In Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, page 354. IEEE Computer Society, 2002.
- [Horn 81] B.K.P. Horn & B.G. Schunck. Determining Optical Flow. Artificial IntelligenceI, vol. 17, no. 1-3, pages 185–203, August 1981.
- [Hu 62] M.K. Hu. Visual pattern recognition by moment invariants. Information Theory, IRE Transactions on, vol. 8, no. 2, pages 179–187, 1962.
- [Ikizler 08] N. Ikizler, R.G. Cinbis & P. Duygulu. Human action recognition with line and flow histograms. In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, pages 1–4. IEEE, 2008.
- [Inria] Inria. Motion2D. Copyright ©1995-2005 by Inria.
- [Interrante 98] V. Interrante & C. Grosch. Visualizing 3D flow. Computer Graphics and Applications, IEEE, vol. 18, no. 4, pages 49–53, 1998.
- [Irani 92] M. Irani, B. Rousso & S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In Computer Vision - ECCV'92, pages 282–287. Springer, 1992.
- [Jacob 95] M-A. Jacob & E. Andres. On Discrete Rotations. In 5th Int. Workshop on Discrete Geometry for Computer Imagery, Clermont-Ferrand (France), pages 161–174. Université de Clermont-Ferrand I, September 1995.
- [Jahne 93] B. Jahne. Spatio-temporal image processing : Theory and scientific applications. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1993.
- [Jardin 08] T. Jardin, L. David, A. Farcy & L. Chatellier. Correlation between vortex structures and unsteady loads for flapping motion in hover. In 14th international symposium on application of laser techniques to fluid mechanics, 2008.
- [Jeong 95] J. Jeong & F. Hussain. On the identification of a vortex. Journal of Fluid Mechanics, vol. 285, no. 69, pages 69–94, 1995.
- [Jhuang 07] H. Jhuang, T. Serre, L. Wolf & T. Poggio. A biologically inspired system for action recognition. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8. IEEE, 2007.
- [Jiang 08] H. Jiang & D.R. Martin. Finding Actions Using Shape Flows. In Proceedings of the 10th European Conference on Computer Vision : Part II, pages 278–292. Springer-Verlag, 2008.

- [JOHANSSON 73] G. JOHANSSON. Visual perception of biological motion and a model for its analysis. Perception & psychophysics, vol. 14, no. 2, pages 201–211, 1973.
- [John 94] George H. John, Ron Kohavi & Karl Pfleger. Irrelevant Features and the Subset Selection Problem. In MACHINE LEARNING : PROCEEDINGS OF THE ELEVENTH INTERNATIONAL, pages 121–129. Morgan Kaufmann, 1994.
- [Kadir 03] T. Kadir & M. Brady. Scale saliency : A novel approach to salient feature and scale selection. In Visual Information Engineering, 2003. VIE 2003. International Conference on, pages 25–28. IET, 2003.
- [Ke 05] Y. Ke, R. Sukthankar & M. Hebert. Efficient Visual Event Detection Using Volumetric Features. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1-Volume 01, pages 166–173. IEEE Computer Society, 2005.
- [Ke 07a] Y. Ke, R. Sukthankar & M. Hebert. Event detection in crowded videos. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8. IEEE, 2007.
- [Ke 07b] Y. Ke, R. Sukthankar & M. Hebert. Spatio-temporal shape and flow correlation for action recognition. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007.
- [Kellokumpu 08a] V. Kellokumpu, G. Zhao & M. Pietikäinen. Human activity recognition using a dynamic texture based method. In Proc. The British Machine Vision Conference (BMVC 2008), Leeds, UK, pages 885–894. Citeseer, September 2008.
- [Kellokumpu 08b] V. Kellokumpu, G. Zhao & M. Pietikäinen. Texture Based Description of Movements for Activity Analysis. In Int. Conf. on Computer Vision Theory and Applications (VISAPP 2008), volume 1, pages 206–213, 2008.
- [Kihl 08a] O. Kihl, B. Tremblais & B. Augereau. Multivariate orthogonal polynomials to extract singular points. In IEEE International Conference on Image Processing (ICIP08), San Diego, USA, October 2008.
- [Kihl 08b] Olivier Kihl, Ludovic Chatellier, Benoit Tremblais & B. Augereau. Détection de structures singulières : étude comparative de deux approches. In Congrès Francophone de Techniques Laser, CFTL 2008, Futuroscope, 16-19 septembre 2008 Congrès Francophone de Techniques Laser, CFTL 2008, pages –, Poitiers France, 09 2008.
- [Kläser 08] A. Kläser, M. Marszałek & C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In British Machine Vision Conference, pages 995–1004. Citeseer, 2008.
- [Koga 85] T. Koga, K. Iinuma, A. Hirano, Y. Iijima & T. Ishiguro. Motion-compensated interframe coding for video conferencing. Teleconferencing, page 63, 1985.
- [Kurtulus 07] D. F. Kurtulus, F. Scarano & L. David. Unsteady aerodynamic forces estimation on a square cylinder by TR-PIV. Experiments in Fluids, vol. 42, no. 2, pages 185–196, 2007.

- [Laptev 03] I. Laptev & T. Lindeberg. Space-time interest points. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 432–439. IEEE, 2003.
- [LAPTEV 05] I. LAPTEV. On Space-Time Interest Points. International Journal of Computer Vision, vol. 64, no. 2/3, pages 107–123, 2005.
- [Laptev 08] I. Laptev, M. Marszalek, C. Schmid & B. Rozenfeld. Learning realistic human actions from movies. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008.
- [Le Besnerais 05] G. Le Besnerais & F. Champagnat. Dense optical flow by iterative local window registration. In Proceedings of Int. Conf. on Image Processing, Gènes (Italie), 2005.
- [Le Pennec 00] E. Le Pennec & S. Mallat. Image compression with geometrical wavelets. In 2000 International Conference on Image Processing, 2000. Proceedings, volume 1, 2000.
- [Li 94] R. Li, B. Zeng & ML Liou. A new three-step search algorithm for block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 4, pages 438–442, 1994.
- [Li 07] X. Li. HMM based action recognition using oriented histograms of optical flow field. Electronics Letters, vol. 43, no. 10, pages 560–561, 2007.
- [Lowe 04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. International journal of computer vision, vol. 60, no. 2, pages 91–110, 2004.
- [Lu 97] J. Lu & M.L. Liou. A simple and efficient search algorithm for block-matching motion estimation. IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 2, pages 429–433, 1997.
- [Lu 06a] W.L. Lu & JJ Little. Simultaneous Tracking and Action Recognition using the PCA-HOG. In Computer and Robot Vision, 2006. The 3rd Canadian Conference on, pages 6–6. IEEE, 2006.
- [Lu 06b] W.L. Lu & J.J. Little. Tracking and recognizing actions at a distance. In Proceedings of the ECCV Workshop on Computer Vision Based Analysis in Sport Environments (CVBASE06), Graz, Austria, 2006.
- [Lucas 81a] B.D. Lucas & T. Kanade. An iterative image registration technique with an application to. In International joint conference on artificial intelligence, volume 3, page 3. Citeseer, 1981.
- [Lucas 81b] B.D. Lucas & T. Kanade. An iterative image registration technique with an application to. In International joint conference on artificial intelligence, volume 3, page 3, 1981.
- [Mallat 99] S.G. Mallat. A wavelet tour of signal processing. Academic Pr, 1999.
- [Mallat 09] S. Mallat & G. Peyré. A Wavelet Tour of Signal Processing : The Sparse Way. Academic Pr, 2009.
- [Martin 04] D.R. Martin, C.C. Fowlkes & J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 5, pages 530–549, 2004.

- [Maurizot 95] M. Maurizot, P. Bouthemy, B. Delyon, A. Juditski, JM Odobez & R. IRISA. Determination of singular points in 2D deformable flow fields. In *Image Processing, 1995. Proceedings., International Conference on*, volume 3. Santa Barbara, 1995.
- [Maurizot 97] M. Maurizot. Analyse du mouvement fluide 2D dans des séquences d'images numériques : localisation, caractérisation et suivi des points singuliers. PhD thesis, Université de Rennes 1, 1997.
- [Mémin 98] É. Mémin & P. Pérez. Joint estimation-segmentation of optic flow. *Computer Vision - ECCV'98*, page 563, 1998.
- [Memini 99] E. Memini & P. Perez. Fluid motion recovery by coupling dense and parametric vectors. In *Proc. ICCV*, pages 620–625, 1999.
- [Mémin 02] E. Mémin & P. Pérez. Hierarchical Estimation and Segmentation of Dense Motion Fields. *International Journal of Computer Vision*, vol. 2, no. 46, pages 129–155, 2002.
- [Meyer 90] Y. Meyer. *Ondelettes.* Hermann, 1990.
- [Mitra 07] S. Mitra & T. Acharya. Gesture recognition : A survey. *Systems, Man, and Cybernetics, Part C : Applications and Reviews*, IEEE Transactions on, vol. 37, no. 3, pages 311–324, 2007.
- [Moeslund 06] T.B. Moeslund, A. Hilton & V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, vol. 104, no. 2-3, pages 90–126, 2006.
- [Moscheni 98] F. Moscheni, S. Bhattacharjee & M. Kunt. Spatiotemporal Segmentation Based on Region Merging. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, pages 897–915, 1998.
- [Neagoe 90] V. Neagoe. Chebyshev nonuniform sampling cascaded with the discrete cosine transform. *IEEE TRANS. ACOUST. SPEECH SIGNAL PROCESS.*, vol. 38, no. 10, pages 1812–1815, 1990.
- [Ngo 02] C.W. Ngo, T.C. Pong & H.J. Zhang. Motion-based video representation for scene change detection. *International Journal of Computer Vision*, vol. 50, no. 2, pages 127–142, 2002.
- [Nie 02] Y. Nie & K.K. Ma. Adaptive rood pattern search for fast block-matching motion estimation. *IEEE Transactions on Image Processing*, vol. 11, no. 12, pages 1442–1449, 2002.
- [Nir 08] Tal Nir, Alfred M. Bruckstein & Ron Kimmel. Over-Parameterized Variational Optical Flow. *Int. J. Comput. Vision*, vol. 76, pages 205–216, February 2008.
- [Nogawa 97] H. Nogawa, Y. Nakajima, Y. Sato & S. Tamura. Acquisition of symbolic description from flow fields : a new approach based on a fluid flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pages 58–63, 1997.

- [Odobez 95] J.M. Odobez & P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of visual communication and image representation*, vol. 6, no. 4, pages 348–365, 1995.
- [Odobez 98] J.M. Odobez & P. Bouthemy. Direct incremental model-based image motion segmentation. *Signal Processing*, vol. 66, no. 2, pages 143–155, 1998.
- [Ogata 06] T. Ogata, W. Christmas, J. Kittler & S. Ishikawa. Improving human activity detection by combining multi-dimensional motion descriptors with boosting. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 295–298. IEEE, 2006.
- [Oikonomopoulos 05] A. Oikonomopoulos, I. Patras & M. Pantic. Spatiotemporal salient points for visual recognition of human actions. *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, vol. 36, no. 3, pages 710–719, 2005.
- [Ojala 02] Timo Ojala, Matti Pietikäinen & Topi Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pages 971–987, 2002.
- [Okuma 04] K. Okuma, A. Taleghani, N. Freitas, J.J. Little & D.G. Lowe. A boosted particle filter : Multitarget detection and tracking. *Computer Vision-ECCV 2004*, pages 28–39, 2004.
- [Palis 82a] J. Palis & W. De Melo. Geometric theory of dynamical systems. An introduction. Transl. from the Portuguese by AK Manning. New York-Heidelberg-Berlin : Springer-Verlag, 1982.
- [Palis 82b] J. Palis, W. De Melo & A.K. Manning. Geometric theory of dynamical systems. Springer-Verlag, New York, 1982.
- [Panagiotakis 04] C. Panagiotakis & G. Tziritas. Recognition and tracking of the members of a moving human articulated motion and deformable objects, pages 86–98, 2004.
- [Park 04] S.C. Park, H.S. Lee & S.W. Lee. Qualitative estimation of camera motion parameters from the linear composition of optical flow. *Pattern Recognition*, vol. 37, no. 4, pages 767–779, 2004.
- [Pennec 02] E. Le Pennec. Bandelettes et représentation géométrique des images. PhD thesis, École polytechnique, Décembre 2002.
- [Po 96] L.M. Po & W.C. Ma. A novel four-step search algorithm for fast block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pages 313–317, 1996.
- [Poelman 94] C. Poelman & T. Kanade. A paraperspective factorization method for shape and motion recovery. *Computer Vision ECCV 94*, pages 97–108, 1994.
- [Polana 94] R. Polana & R. Nelson. Low level recognition of human motion. In *Proc. IEEE Workshop on Nonrigid and Articulate Motion*, pages 77–82, 1994.
- [Popinet 03] S. Popinet. Gerris : a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *J. Comput. Phys.*, vol. 190, no. 2, pages 572–600, 2003.
- [Pope 10] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, vol. 28, no. 6, pages 976–990, 2010.

- [R. A. Finkel 74] J. L. Bentley R. A. Finkel. Quad Trees : A Data Structure for Retrieval on Composite Keys. Acta Informatica, vol. 4, no. 1, pages 1–9, March 1974.
- [Ramis 83] E. Ramis, C. Deschamps & J. Odoux. Cours de mathématiques spéciales, algèbre, vol. 1. Masson, FRANCE, 1983.
- [Rao 92a] AR Rao & RC Jain. Computerized flow field analysis : Oriented texture fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 7, pages 693–709, 1992.
- [Rao 92b] AR Rao & RC Jain. Computerized flow field analysis : Oriented texture fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 7, pages 693–709, 1992.
- [Rapantzikos 07] K. Rapantzikos, Y. Avrithis & S. Kollias. Spatiotemporal saliency for event detection and representation in the 3D Wavelet Domain : Potential in human action recognition. In Proceedings of the 6th ACM international conference on Image and video retrieval, pages 294–301. ACM, 2007.
- [Rice 69] J. R. Rice. Approximation des fonctions - théorie linéaire. DUNOD, 1969.
- [Rougon 00] N.F. Rougon, MA Brossard-Pailleux & F.J. Preteux. Robust parametric estimation over optimal support of fluid flow structure in multiscale. In Proceedings of SPIE, volume 4121, page 39, 2000.
- [Samet 07] Hanan Samet. Spatial data structures. In SIGGRAPH '07 : ACM SIGGRAPH 2007 courses, page 1, New York, NY, USA, 2007. ACM.
- [Sánchez 05] C.C. Sánchez. Modèle connexionniste neuromimétique pour la perception visuelle embarquée du mouvement. These de doctorat : Docteur en Informatique, Université Henri Poincaré (Nancy I), Nancy, Meurthe-et-Moselle, Lorraine, France, 2005.
- [Scarano 00] F. Scarano & M. L. Riethmuller. Advances in iterative multigrid PIV image processing. Experiments in Fluids, vol. 29, no. 7, pages 51–60, 2000.
- [Scarano 02] F. Scarano. Iterative image deformation methods in P.I.V. Measurement, Science and Technology, vol. 13, no. 1, pages R1–R19, 2002.
- [Schindler 08] K. Schindler & L. Van Gool. Action snippets : How many frames does human action recognition require ? In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008.
- [Schuldt 04] C. Schuldt, I. Laptev & B. Caputo. Recognizing human actions : A local SVM approach. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, volume 3, pages 32–36. IEEE, 2004.
- [Scovanner 07] P. Scovanner, S. Ali & M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In Proceedings of the 15th international conference on Multimedia, pages 357–360. ACM, 2007.

- [Shechtman 05] E. Shechtman & M. Irani. Space-Time Behavior Based Correlation. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Volume 1-Volume 01, pages 405–412. IEEE Computer Society, 2005.
- [Sheikh 03] HR Sheikh, Z. Wang, L. Cormack & AC Bovik. LIVE image quality assessment database release 2. Download at : <http://live.ece.utexas.edu/research/quality>, 2003.
- [Sheikh 06] H.R. Sheikh, M.F. Sabir & A.C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. Image Processing, IEEE Transactions on, vol. 15, no. 11, pages 3440–3451, 2006.
- [Shu 93] C.F. Shu & R.C. Jain. Direct estimation and error analysis for oriented patterns. CVGIP : Image Understanding, vol. 58, no. 3, pages 383–398, 1993.
- [Simoncelli 98] E.P. Simoncelli & D.J. Heeger. A model of neuronal responses in visual area MT. Vision Research, vol. 38, no. 5, pages 743–761, 1998.
- [SIROVICH 87] L. SIROVICH. Turbulence and the dynamics of coherent structures. I : Coherent structures. Quarterly of applied mathematics, vol. 45, no. 3, pages 561–590, 1987.
- [Smeaton 08] A. F. Smeaton, P. Over & W. Kraaij. TREC Video Retrieval Evaluation. www-nlpir.nist.gov/projects/trecvid/, 2008.
- [Snoek 09] J. Snoek, J. Hoey, L. Stewart, R.S. Zemel & A. Mihailidis. Automated detection of unusual events on stairs. Image and Vision Computing, vol. 27, no. 1-2, pages 153–166, 2009.
- [Souvenir 08] R. Souvenir & J. Babbs. Learning the viewpoint manifold for action recognition. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–7. IEEE, 2008.
- [Spivak 79] M. Spivak & M. Spivak. A comprehensive introduction to differential geometry, volume 1. Publish or perish Berkeley, 1979.
- [Sun 09] J. Sun, X. Wu, S. Yan, L.F. Cheong, T.S. Chua & J. Li. Hierarchical spatio-temporal context modeling for action recognition. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 2004–2011. IEEE, 2009.
- [Susset 02] A. Susset. Développement de traitements d'images pour l'étude de la stabilisation de flammes turbulentes non prémélangées générées par des brûleurs industriels modèles. PhD thesis, Université de Poitiers, 2002.
- [Tabbone 06] S. Tabbone, L. Wendling & J.P. Salmon. A new shape descriptor defined on the Radon transform. Computer Vision and Image Understanding, vol. 102, no. 1, pages 42–51, 2006.
- [Taubman 00] D. Taubman. High performance scalable image compression with EBCOT. Image Processing, IEEE transactions on, vol. 9, no. 7, pages 1158–1170, 2000.
- [Tran 08] Du Tran & Alexander Sorokin. Human Activity Recognition with Metric Learning. In David Forsyth, Philip Torr & Andrew Zisserman, editeurs, Computer Vision - ECCV 2008, volume 5302 of Lecture Notes in Computer Science, pages 548–561. Springer Berlin Heidelberg, 2008.

- [Truchetet 98] F. Truchetet. Ondelettes pour le signal numérique. Hermès, 1998.
- [Vasilescu 02] M.A.O. Vasilescu. Human Motion Signatures : Analysis, Synthesis, Recognition. In IN PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION (ICPR 02), pages 456–460. Citeseer, 2002.
- [Viola 01] P. Viola & M. Jones. Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, volume 1, pages I–511. IEEE Comput. Soc, 2001.
- [Vishwanathan 07] S.V.N. Vishwanathan, A.J. Smola & R. Vidal. Binet-Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. International Journal of Computer Vision, vol. 73, no. 1, pages 95–119, 2007.
- [Vollmers 01] H. Vollmers. Detection of vortices and quantitative evaluation of their main parameters. Measurement Science and Technology, vol. 12, no. 8, pages 1199–1207, 2001.
- [Wald 06] V. Wald I. Havran. On building fast kd-Trees for Ray Tracing, and on doing that in $O(N \log N)$. In Interactive Ray Tracing 2006, IEEE Symposium on, pages 61–69, 2006.
- [Wang 94] J.Y.A. Wang & E.H. Adelson. Representing moving images with layers. IEEE Transactions on Image Processing, vol. 3, no. 5, pages 625–638, 1994.
- [WANG 96] Z. WANG, GA JULLIEN & WC MILLER. On computing Chebyshev optimal nonuniform interpolation. Signal processing, vol. 51, no. 3, pages 223–228, 1996.
- [Wang 04] Z. Wang, A.C. Bovik, H.R. Sheikh & E.P. Simoncelli. Image quality assessment : From error visibility to structural similarity. Image Processing, IEEE Transactions on, vol. 13, no. 4, pages 600–612, 2004.
- [Wang 07a] J.M. Wang, D.J. Fleet & A. Hertzmann. Multifactor Gaussian process models for style-content separation. In Proceedings of the 24th international conference on Machine learning, pages 975–982. ACM, 2007.
- [Wang 07b] L. Wang & D. Suter. Learning and matching of dynamic shape manifolds for human motion. IEEE Transactions on Image Processing, vol. 16, no. 6, page 1646, 2007.
- [Wang 07c] Y. Wang, K. Huang & T. Tan. Human activity recognition based on r transform. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007.
- [Weickert 06] Joachim Weickert, Andr s Bruhn, Thomas Brox & Nils Papenberg. A Survey on Variational Optic Flow Methods for Small Displacements. In Otmar Scherzer, Hans-Georg Bock, Frank Hoog, Avner Friedman, William Langford, Helmut Neunzert, William R. Pulleyblank, Torgeir Rusten & Anna-Karin Tornberg, editeurs, Mathematical Models for Registration and Applications to Medical Imaging, volume 10 of

- Mathematics in Industry, pages 103–136. Springer Berlin Heidelberg, 2006.
- [Weinland 05] D. Weinland, R. Ronfard & E. Boyer. Motion history volumes for free viewpoint action recognition. In IEEE International Workshop on Modeling People and Human Interaction, volume 104, pages 249–257. Cite-seer, 2005.
- [Weinland 06] D. Weinland, R. Ronfard & E. Boyer. Free viewpoint action recognition using motion history volumes. Computer Vision and Image Understanding, vol. 104, no. 2-3, pages 249–257, 2006.
- [Weinland 07] D. Weinland, E. Boyer & R. Ronfard. Action Recognition from Arbitrary Views using 3D Exemplars. In International Conference on Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–7. IEEE, 2007.
- [Weinland 08] D. Weinland & E. Boyer. Action recognition using exemplar-based embedding. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–7. IEEE, 2008.
- [Willems 08] G. Willems, T. Tuytelaars & L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. Computer Vision–ECCV 2008, pages 650–663, 2008.
- [Wu 00] Y.T. Wu, T. Kanade, C.C. Li & J. Cohn. Image registration using wavelet-based motion model. International Journal of Computer Vision, vol. 38, no. 2, pages 129–152, 2000.
- [Wu 04] J. Wu, J.M. Rehg & M.D. Mullin. Learning a rare event detection cascade by direct feature selection. Advances in Neural Information Processing Systems, vol. 16, pages 1523–1530, 2004.
- [Yang 00] Q. Yang & B. Parvin. Feature based visualization of geophysical data. In IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings, volume 2, 2000.
- [Yang 07] C. Yang, Y. Guo, H. Sawhney & R. Kumar. Learning actions using robust string kernels. Human Motion–Understanding, Modeling, Capture and Animation (HUMO-07), pages 313–327, 2007.
- [Zelnik-Manor 01] L. Zelnik-Manor & M. Irani. Event-based analysis of video. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 2, pages II–123. IEEE, 2001.
- [Zhang 08] Ziming Zhang, Yiqun Hu, Syin Chan & Liang-Tien Chia. Motion Context : A New Representation for Human Action Recognition. In David Forsyth, Philip Torr & Andrew Zisserman, editeurs, Computer Vision - ECCV 2008, volume 5305 of Lecture Notes in Computer Science, pages 817–829. Springer Berlin Heidelberg, 2008.
- [Zhao 07] G. Zhao et al. Dynamic texture recognition using local binary patterns with an application. IEEE transactions on pattern analysis and machine intelligence, pages 915–928, 2007.
- [Zhu 97] S. Zhu & K.K. Ma. A new diamond search algorithm for fast block matching motion estimation. In Proceedings of 1997 International Conference on Information, Communications and Signal Processing, 1997. ICICS., volume 1, 1997.

PUBLICATIONS PERSONELLES

Conférences internationales :

[1] O. Kihl, B. Tremblais, B. Augereau. Multivariate orthogonal polynomials to extract singular points, in Proceedings of the IEEE International Conference on Image Processing 2008 ICIP 2008, United States, Oct. 2008.

[2] O. Kihl, B. Tremblais, B. Augereau, M Khoudeir. Human activities discrimination with motion approximation in polynomial bases, IEEE International Conference on Image Processing 2010 ICIP 2010.

Conférences nationales :

[3] O. Kihl, L. Chatellier, B. Tremblais, B. Augereau. Détection de structures singulières : étude comparative de deux approches, Congrès Francophone de Techniques Laser, CFTL 2008, France, Sept 2008.

Modélisations polynomiales hiérarchisées applications à l'analyse de mouvements complexes

Résumé : Dans ce mémoire, nous proposons une méthode de modélisation du mouvement à l'aide de bases de polynômes bi-variables orthogonaux. L'objectif est de proposer une modélisation hiérarchique des champs de vecteurs adaptable à tous types de champs. Nous étudions différentes méthodes de partitionnement de l'espace, parmi lesquelles les grilles régulières, les quadtree ou les diagrammes de Voronoï, afin de prendre en compte la complexité locale du champ pour affiner la modélisation. Nous obtenons des résultats qualitatifs montrant l'intérêt de ce type de modélisation. De plus, nous proposons une procédure de génération des bases afin de modéliser les champs épars. Nous appliquons cette méthode de modélisation dans deux contextes. La détection de points singuliers dans les champs de déplacement et la reconnaissance de mouvements humains. Ces deux applications nous permettent de mettre en évidence l'efficacité de cette méthode, puisque nous obtenons des résultats comparables aux méthodes les plus efficaces de la littérature.

Hierarchical polynomial modeling applications to complex motion analysis

Abstract : In this PhD thesis, we suggest a method which models every kind of movement with orthogonal polynomials basis. The main goal is to propose a hierarchical modeling of vector fields suitable for all types of fields. We study different space partitioning methods, including regular grids, quadtree and Voronoi diagrams, in order to take into account the local complexity of the field in order to refine modeling. We obtain qualitative results showing the benefit of this method. In addition, we propose a procedure for generating polynomial bases to model sparse fields. We apply this modeling approach in two contexts. The detection of singular points in the motion fields and the recognition of human movements. Both applications allow us to demonstrate the effectiveness of this method, since we obtain similar results to the most effective methods of literature.

Discipline : traitement du signal et des images

Mots clés : modélisation du mouvement, analyse du mouvement, polynômes orthogonaux, détection de singularités, mouvements humains

Laboratoire XLIM-SIC, UMR CNRS 6172
Bâtiment SP2MI - Téléport 2, Bd Marie et Pierre Curie
BP 30179 - 86962 Futuroscope Chasseneuil CEDEX